

Thomas Seidl Marwan Hassani Christian Beecks (Eds.)

**Proceedings of the LWA 2014
Workshops: KDML, IR and FGWM**

Aachen, Germany, September 8-10, 2014

Preface

The 16th edition of the conference LWA - “Lernen, Wissen, Adaption” (in German) which translates to “Learning, Knowledge, Adaptation” brings together researchers which deal with the discovery, the management and the retrieval of knowledge. Following the tradition of previous years, LWA comprises three workshops organized by representatives of the respective special interest groups of the Gesellschaft für Informatik (GI), which is the German computer science society. These workshops address the following topics:

- KDML - Knowledge Discovery, Data Mining, and Machine Learning
- IR - Information Retrieval
- FGWM - Knowledge Management

The papers have been selected by independent program committees from the respective domains. The workshops run in parallel sessions in close vicinity to enable the exchange of ideas. They particularly meet in a joint session which includes flagship contributions of particular interest for all conference participants.

Recent trends in the corresponding research areas are highlighted by distinguished keynote speakers. In his talk on “Interdisciplinary Machine Learning”, Ulf Brefeld from TU Darmstadt points out the inherent interdisciplinarity of machine learning research as an important building block. Michael Kohlhase from Jacobs University Bremen speaks about “Mathematical Knowledge Management and Information Retrieval: Transcending the One-Brain-Barrier” and advocates for exploiting more mathematics in developing and evaluating of knowledge management concepts. Carsten Dolch from Deloitte illustrates how data analytics projects in industrial consulting contexts are conducted. In addition to these keynotes, Mirjam Minor from the Goethe University Frankfurt am Main will give a special keynote about “Case-based Reasoning in the Cloud”.

As a hand-shake of social and technical program, all authors got invited to present their work at the poster reception. A guided city tour in the historical city center of Aachen followed by the conference dinner in the university quarter complements the social program. The data management and data exploration group in the Department of Computer Science at RWTH Aachen University is proud to host the LWA 2014 conference. We hope the participants will keep the venue as an inspiring event with fruitful discussions in mind and the readers will enjoy studying the scientific contributions in this proceedings volume.

September 2014

Thomas Seidl
Marwan Hassani
Christian Beecks
Editors, LWA'14

Organization

The LWA conference series traditionally comprises the workshops IR, KDML and FGWM which are organized by the respective special interest groups within the Gesellschaft für Informatik (German Computer Science Society). LWA 2014 is organized by the Chair of Computer Science 9 (Data Management and Data Exploration) at the Department of Computer Science, RWTH Aachen University, Germany.

KDML'14 Workshop Organization

| | |
|--------------------|------------------------------------|
| Florian Lemmerich | University of Würzburg |
| Eneldo Loza Mencía | Darmstadt University of Technology |

IR'14 Workshop Organization

| | |
|------------------|---|
| Sascha Kriewel | University of Duisburg-Essen |
| Claus-Peter Klas | GESIS Leibniz Institute for the Social Sciences |

FGWM'14 Workshop Organization

| | |
|--------------------|--|
| Michael Leyer | Frankfurt School of Finance & Management |
| Joachim Baumeister | denkbares GmbH |

General Coordination

| | |
|-----------------------|---|
| Institution: | Chair of Computer Science 9, RWTH Aachen University |
| General Chair: | Thomas Seidl |
| Local Coordinator: | Christina Rensinghof |
| Technical Program: | Christian Beecks |
| Social Program: | Merih Seran Uysal |
| Technical Assistance: | Sergej Fries, Brigitte Boden and Detlef Wetzeler |
| Web Setup: | Ines Färber and Anca Zimmer |
| Proceedings Manager: | Marwan Hassani |

KDML'14 Program Committee

| | | |
|---------------------|--------------|--------------------|
| Martin Atzmueller | Daniel Bengs | Wouter Duivesteijn |
| Christian Bauckhage | | |

Johannes Fürnkranz
Stephan Günnemann
Andreas Hotho
Kristian Kersting
Peer Kröger

Florian Lemmerich
Eneldo Loza Mencía
Emmanuel Müller
Nico Piatkowski
Ute Schmid

Lars Schmidt-Thieme
Robin Senge
Albrecht Zimmermann

IR'14 Program Committee

David Elsweiler
Reginald Ferber
Norbert Fuhr
Joachim Griesbaum
Daniel Hienert
Andreas Henrich
Katja Hofmann

Frank Hopfgartner
Udo Kruschwitz
Johannes Leveling
Thomas Mandl
Philipp Mayr
Henning Müller
Peter Mutschke

Ralf Schenkel
Ingo Schmitt
Hans-Christian Sperker
Christian Wolff
Christa Womser-Hacker
David Zellhoefer

FGWM'14 Program Committee

Klaus-Dieter Althoff
Kerstin Bach Verdande
Axel Benjamins
Mareike Dornhöfer
Susanne Durst
Michael Fellmann

Martina Freiberg
Dimitris Karagiannis
Andrea Kohlhase
Christoph Lange
Ronald Maier
Mirjam Minor

Ulrich Reimer
Jochen Reutelshöfer
Thomas Roth-Berghofer
Bodo Rieger
Peter Rossbach

Keynote Talks

Interdisciplinary Machine Learning

Ulf Brefeld

Knowledge Mining & Assessment Group
TU Darmstadt, Germany
`brefeld@kma.informatik.tu-darmstadt.de`

Abstract. Interdisciplinary cooperations are sometimes viewed sceptically as they often involve non-standard problem settings and interactions with researchers and practitioners from other domains. Thus, interdisciplinary projects may require more dedication and engagement than working on off-the-shelf problems and downloadable data sets. In this talk, I will argue that machine learning is intrinsically an interdisciplinary discipline. Reaching out to other domains constitutes an important building block to advance the field of machine learning as it is the key to finding interesting and novel challenges and problem settings. Establishing an abstract view on such a novel problem setting often allows to identify surprisingly unrelated tasks that fall into the same equivalence class of problems and can thus be addressed with the same methods. I will present examples from ongoing research projects.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

Analytics Applied: Current Market Trends and Case Studies

Carsten Dolch

Deloitte GmbH
cdolch@deloitte.de

Abstract. Analytics is becoming more and more part of the decision making process for management and operational work. Within this session the Deloitte Analytics Institute wants to provide you with an insight into an user experience based approach, how to engage customers with analytics applications and how analytics becomes the key driver for IT landscape transformation.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

Mathematical Knowledge Management and Information Retrieval: Transcending the One-Brain-Barrier

Michael Kohlhase

Jacobs University Bremen, Germany
m.kohlhase@jacobs-university.de

Abstract. The talk presents the discipline of Mathematical Knowledge Management (MKM), which studies the possibility of computer-supporting and even automating the representation, cataloguing, retrieval, refactoring, plausibilization, change propagation and in some cases even application of knowledge. Mathematics is a suitable test domain, as mathematical language is intrinsically rich in structure, rigorous but diverse in presentation, and non-trivial but sufficiently well-understood in content.

We focus on theory graph technology here, which supports modular and thus space/computation/cognitively-efficient representations of mathematical knowledge and allows MKM systems to achieve a limited mathematical literacy that is necessary to complement the abilities of human mathematicians and thus to enhance their productivity.

For more details see <http://www.ems-ph.org/journals/newsletter/pdf/2014-06-92.pdf>.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

The Joint Session

Landmark Recognition: State-of-the-Art Methods in a Large-Scale Scenario*

Magdalena Rischka and Stefan Conrad

Institute of Computer Science
Heinrich-Heine-University Duesseldorf
D-40225 Duesseldorf, Germany
rischka@cs.uni-duesseldorf.de
conrad@cs.uni-duesseldorf.de

Abstract. The recognition of landmarks in images can help to manage large image collections and thus is desirable for many image retrieval applications. A practical system has to be scalable with an increasing number of landmarks. For the domain of landmark recognition we investigate state-of-the-art CBIR methods on an image dataset of 900 landmarks. Our experiments show that the kNN classifier outperforms the SVM in a large-scale scenario. The examined visual phrase concept has shown not to be as effective as the classical Bag-of-Words approach although the most landmarks are objects with a relatively fixed composition of their (nearby) parts.

Keywords: Image Retrieval, Large-Scale, Landmark Recognition, Bag-of-Words, Bag-of-Phrases

1 Introduction

The ongoing development of personal electronic devices like digital cameras, mobile phones or tablets with integrated camera and high-capacity memory cards, as well as their decreasing prices enable taking photos everywhere and at any time. Collecting and storing photos as well as sharing photos with others on online social network platforms leads to huge photo collections in personal households and to a much greater extent on the world wide web. To manage and reuse these images in an useful way (e.g. for search purposes) it is necessary to capture the images' content, i.e. to annotate the images with meaningful textual keys. A large amount of the collections' images are photos shot in the photographer's vacations and trips showing (prominent) places and landmarks the photographer visited. The detection and recognition of landmarks in images offers several advantages regarding applications: the above-mentioned annotation constitutes

* *Copyright* © 2014 by the paper's authors. *Copying permitted only for private and academic purposes.* In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

a foundation for a search or can be used as a suggestion for a photo description to the user. Another usage is the identification of locations the photographers visited, for example to summarize personal image collections by offering an overview of places. The application of mobile landmark recognition enables tourists to look up sights in real-time to obtain informations on them. Capturing images' content by manual annotation of images with landmarks however is very time-consuming, in the scale of these collections even inconvertible, therefore an automatic solution is needed. Several systems for automatic landmark recognition have been proposed [2–6] differing in the focus of application scenario, the initial situation referring metadata, problem definition and implemented techniques. For example the authors of [2] create a database from geo-tagged Flickr photos and Wikipedia. Object-level recognition is performed with the aid of an index and candidate images are ranked using a TF-IDF scheme. [3] also creates a dataset from Flickr images and then derives scene maps of landmarks which are retrieved with an inverted index. [4] creates the database by crawling travel guide websites and then builds a matching graph out of the feature matches of the images. For retrieval a kd-tree is used. We concentrate on images without any metadata, thus on content-based methods only. Several state-of-the-art methods in CBIR have been examined and applied successfully on small or average size datasets. Our focus is on the large-scale aspect of a landmark recognition system and the usability in real world scenarios, thus our contribution is the comparison of these methods with reference to scalability.

The remainder of this paper is organized as follows: in the next section we outline and formalize the problem of landmark recognition by defining the landmark term, describing the characteristics of landmark images, specifying the landmark recognition task and presenting the components of the landmark recognition system evaluated in section 3. In section 4 we summarize our results and discuss future work.

2 Landmark Recognition Problem and System

A *landmark* is a physical object, created by man or by nature, with a high recognition value. Usually a landmark is of remarkable size and is located on a fixed position of the earth. Examples of landmarks are buildings, monuments, statues, parks, mountains and other structures and places. Due to their recognition value, landmarks often serve as geographical points for navigation and localisation. The largest amount of photos of landmarks contain only one landmark, which in the most cases takes in 80% of the photo area, in very few cases it takes only a small part of the photo (when it is taken from apart). A marginal part of photos show two or more landmarks. A landmark recognition system has to conduct the following task automatically:

Definition 1 (Landmark Recognition Task). *Given a set of L landmarks $\mathcal{L} = \{l_1, \dots, l_L\}$ and an image i whose semantic content is unknown. The task is*

to assign a set of landmarks to the image:

$$i \rightarrow \begin{cases} \emptyset & \text{if image } i \text{ does not contain any landmark} \\ \{l_{j_1}, \dots, l_{j_n}\} & \text{if image } i \text{ contains landmarks } \{l_{j_1}, \dots, l_{j_n}\} \end{cases} \quad (1)$$

This definition implies a multi-label classification problem with a decision refusal. We simplify the multi-label classification problem defined in (1) by building our system on a single-label classification approach, thus we accept a possible misclassification of images containing more than one landmark. We focus on the classification step, the decision refusal which is usually performed with a post-processing verification algorithm (like RANSAC) is beyond this work. The main components of our landmark recognition system, which are the image representation and the classifier are discussed in the following paragraphs.

Image Representation To describe images we extract the popular SIFT [7] features. The SIFT algorithm extracts local features by detecting stable points and then describing the (small) surrounding area around each point by an histogram of gradients. An image i is represented by a set of local SIFT points: $SIFT(i) = \{p_1, \dots, p_P \mid p = (x, y, s, d)\}$ with x, y are the coordinates of the point p in the image, s is the scale and d the 128-dimensional descriptor. We analyse two types of image representation based on the local SIFT features: Bag-of-Words (BoW) and the Bag-of-Phrases (BoP) model based on the visual phrase concept. Although visual phrases have been used in general object recognition applications, they raised less attention in the domain of landmark recognition. We like to analyse if visual phrases improve the BoW classification results.

The *Bag-of-Words* model is a classical approach to create a compact image representation based on local features. The idea is to aggregate local features to one global descriptor and thus to avoid the expensive comparison of images by matching local descriptors against each other. The BoW descriptor bases on a dictionary of visual words which is obtained by partitioning the descriptor-space. Then each partition is represented by an instance of this partition, usually the center of the partition, which is called the *visual word*. Several methods for partitioning the descriptor-space have been proposed, a simple and most used one is the k-Means clustering algorithm which requires the input parameter k (which onwards is denoted as D to differentiate between the kNN parameter k) for the number of clusters (visual words) to obtain.

Definition 2 (Bag-of-Words Model).

Given a dictionary $\mathcal{D} = \{(w_1, c_1), \dots, (w_D, c_D)\}$ of D visual words (w_j, c_j) (w_j is label, c_j the center of the partition) and an image in SIFT representation. Each SIFT point p of the image is assigned to its visual word w_p by:

$$w_p := w_j = \underset{(w_j, c_j) \in \mathcal{D}}{\operatorname{argmin}} (\operatorname{EuclideanDistance}(d, c_j)) \quad (2)$$

The Bag-of-Words image representation is given by:

$$BoW(i) = \{f_1, \dots, f_D\} \text{ with } f_j = \frac{1}{P} \sum_{p=1}^P \begin{cases} 1, & w_p = j \\ 0, & \text{else} \end{cases} \quad (3)$$

Visual phrases catch spatial relations in local neighborhood by considering pairs of nearby local features or visual words to support more semantic, analogously to phrases in text retrieval. We follow [8] and define the visual phrase and the Bag-of-Phrases model as follows:

Definition 3 (Bag-of-Phrases Model).

Given the visual dictionary $\mathcal{D} = \{(w_1, c_1), \dots, (w_D, c_D)\}$. A visual phrase $ph_{j,k}$ is a pair of visual words: $ph_{j,k} = (w_j, w_k)$ with $j \leq k$. An image in SIFT representation with its visual words $SIFT'(i) = \{p_1, \dots, p_P \mid p = (x, y, s, d, w)\}$ contains the phrase $ph_{j,k}$ if there exist two SIFT points p_m and p_n with their visual words w_j and w_k and it holds

$$EuclideanDistance((x_m, y_m), (x_n, y_n)) \leq \max(\lambda \cdot s_m, \lambda \cdot s_n) \quad (4)$$

for a fixed scale factor λ . The Bag-of-Phrases image representation is given by:

$$BoP(i) = \{f_1, \dots, f_{D_2}\} \text{ with } D_2 = \frac{D \cdot (D + 1)}{2} \quad (5)$$

with f_j is the relative frequency of the visual phrase ph_j in image i .

Classifier For the choice on the classifier, we evaluate three well-known classifiers, the Support-Vector-Machine (SVM), the k-Nearest Neighbor (kNN) and the Nearest Center classifier (NC). The SVM is a popular classifier as it provides better classification results than other standard classifiers in the most (computer vision) classification tasks. However the drawback of the SVM classifier is the long classifier learning time, especially with an increasing training data size. In addition to that [1] has shown that the superiority of the SVM over the kNN classifier (with regard to classification quality) can swap with an increasing number of classes. As our focus is on the large-scale landmark recognition with reference to an increasing number of landmarks, we investigate the landmark recognition on both classifiers. The kNN classifier has no training part, instead the classification time is linear in the number of training examples, which in a scenario of over 100.000 images and a system implementation without the use of appropriate and efficient access structures can put a strain on the user. However the classifier NC can be seen as a lightweight classifier, as both the learning and the classification time is linear in the number of classes. For the SVM we use the RBF kernel and the one-vs-one mode, for the kNN we set $k = 5$ (as a result of preliminary experiments on different k), for the kNN and the NC classifier we choose the histogram intersection as the similarity function.

3 Evaluation

Evaluation Dataset For the evaluation we use a self-provided dataset of landmarks. We gathered landmark terms from several websites which lists landmarks from all over the world, including the website of [4]¹. Our dataset consists of 900 landmarks from 449 cities and 228 countries. To get images for the training and test sets, we queried the google image search engine with each landmark term (specified by its region - city or country) and then downloaded the results from the original source. For scalability analysis we derived training sets of four different sizes: 45, 300, 600 and 900 landmarks. For each size three training sets (A, B, C) have been created. To create a challenging test set we have chosen images manually from the results of the google image search: The images show the landmarks in their canonical views, under different perspective changes, distortions and lighting conditions (also at night) as well as indoor shootings and parts of the landmarks. The test set consists of 900 landmark images (45 well- and lesser-known landmarks from Europe with 20 test images per landmark). The test images have been proofed to be disjoint from the training set.

Evaluation Measures The outcome of a single-label classifier on a test image is the predicted landmark. To retain a fine-grained evaluation of a test image, we are also interested in a ranking of landmarks, as the ranking reveals how far away is the groundtruth landmark from the top ranking position. The SVM delivers us a ranking based on the probability values of the one-vs-one voting, the NC classifier based on the histogram intersection similarity. The kNN returns only the predicted landmark. To evaluate the results of the classifiers we use two (instance-based) evaluation measures: the (instance-based) recall on the predicted landmark and the MAP measure (which is equal to MRR measure in this case) for the landmarks ranking. We finally report the average value over all test images.

Experiments The first experiment evaluates the Bag-of-Words model in combination with the three classifiers and the four training sets of size 45, 300, 600 and 900. The Bag-of-Words model has one parameter which is the visual dictionary size. We examine the following six different visual dictionary sizes: 500, 1000, 2000, 4000, 6000 and 8000. Figure 1 shows the results of this experiment. The recall and MAP values reported are averages over the training sets A,B,C of the corresponding training set size. Table 1 shows the average recognition time for a test image on the Bag-of-Words model with a visual dictionary size of 8000 depending on the classifier and the training set size. The average recognition time does not include the processing time for image representation computation. Experiments are performed on an usual Intel i7 960 3.2 GHz (64-bit) architecture with 16 GB memory size. The system (kNN) is implemented without the use of any efficient access structure. Considering the recall values of all classifiers for all

¹ http://mingzhao.name/landmark/landmark_html/demo_files/1000_landmarks.html

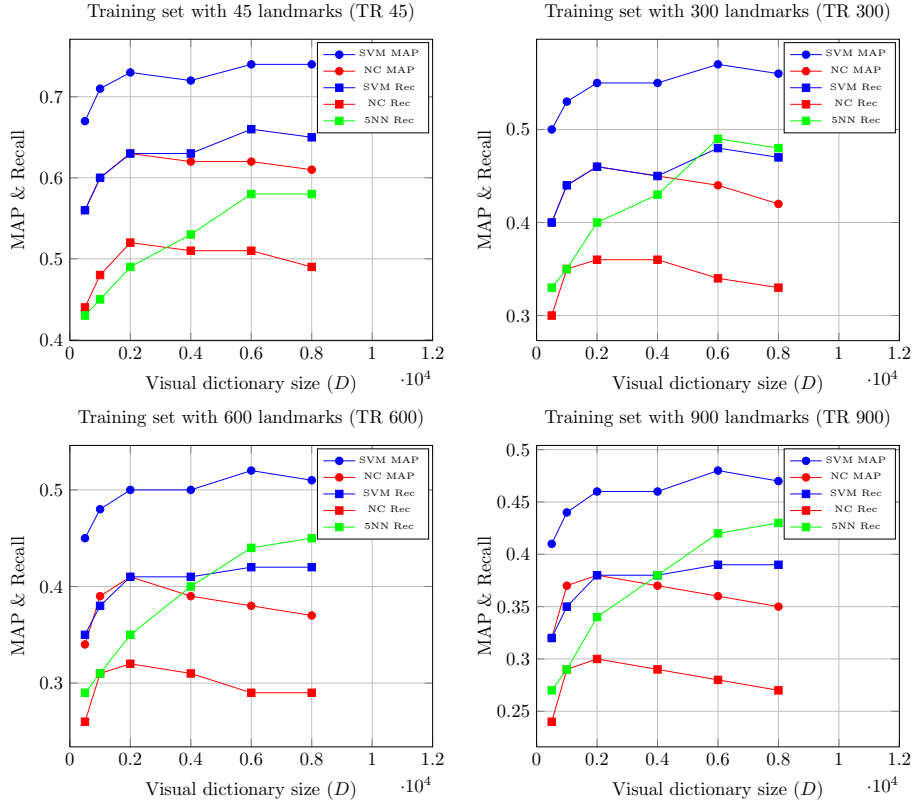


Fig. 1. Classification results of the BoW model depending on the parameters visual dictionary size D (x-axis), classifier with evaluation measure (plots) and training set size (subimages)

training set sizes, we can see that the best values are achieved by the SVM and the 5NN with a visual dictionary size of 6000 and 8000. On the training set TR 45 the SVM gets the best recall value with 0.65 ($D = 6000$). From the training set TR 300 on the 5NN outperforms slightly the SVM resulting in a difference of 4% on TR 900 and $D = 8000$. Furthermore the 5NN shows the tendency to achieve higher results with a growing visual dictionary. These results confirm the observation of the superiority of kNN over the SVM in large-scale problems stated in [1]. The NC classifier achieves best results on $D = 2000$ for all training set sizes, however its best values are on average 13% lower than the best system (SVM or kNN). The MAP values of the SVM and the NC reveal that there is potential to improve these classifiers when involving the next to top ranking positions in the classification decision. In general the recognition accuracy decreases with an increasing number of landmarks which is not surprising. A recall value of 0.66 on the TR 45 (SVM, $D = 6000$) can be somewhat satisfying, however the best result of 0.43 on TR 900 (5NN, $D = 8000$) is less delightful. The

second experiment concentrates on the Bag-of-Phrases model. Again we report experiments in combination with the three classifiers and the four training set sizes. The BoP model requires two parameters to be set: the visual dictionary size D and the scale factor λ . As the dimension of the image’s descriptor in this representation becomes very large on already small visual dictionary sizes, we examined the two sizes 500 and 1000 resulting in the descriptor dimension of 5050 and 125250, respectively. For the scale factor we choose the values 1, 2, 4 and 6. The BoP results (Figure 2) for all classifiers and all training set sizes are on average 10% lower than the BoW results. The larger visual dictionary (500, λ) achieves better results than the smaller one (100, λ), especially on the SVM, whereas the scale factor influences the results slightly. In the most cases the scale factor of 2 gets best results. Due to the high-dimensional descriptor ($D \geq 500$) the recognition time is many times higher than of the BoW model.

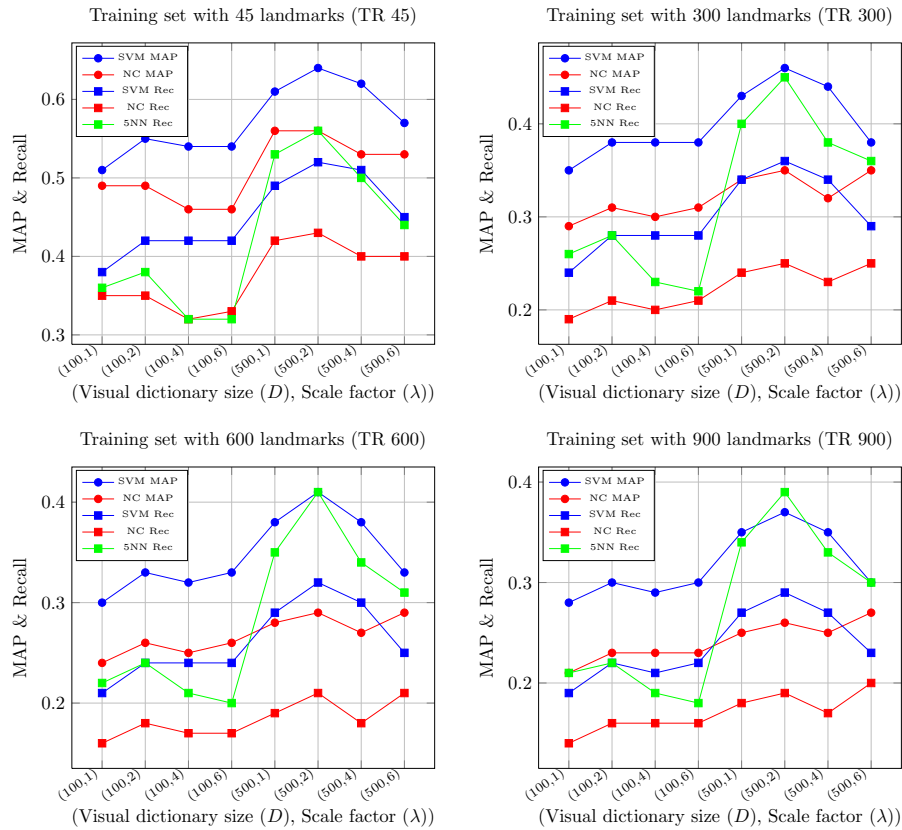


Fig. 2. Classification results of the BoP model depending on the parameters visual dictionary size D and scale factor λ (x-axis), classifier with evaluation measure (plots) and training set size (subimages)

| | TR 45 | TR 300 | TR 600 | TR 900 |
|-----|--------|--------|--------|--------|
| SVM | 0.0577 | 0.1016 | 0.2710 | 0.5081 |
| NC | 0.0039 | 0.0239 | 0.0424 | 0.0612 |
| 5NN | 0.1129 | 0.4637 | 0.8444 | 1.2232 |

Table 1. Average recognition time (in seconds) for the BoW model with a dictionary of size 8000 dependent on the three classifiers and the four training set sizes.

4 Summary and Future Work

To build a landmark recognition system with large number of landmarks (TR 900) supported, the Bag-of-Words model together with the kNN classifier offers a higher recognition accuracy than the SVM but on the cost of a relatively high recognition time of about 1.2 seconds per image. A solution to use kNN and to reduce the recognition time is to integrate an appropriate and efficient access structure into the system and to try to reduce the number of training images per landmark (by a compressed representation) without losing too much relevant informations. The BoP model alone does not convince, therefore the question arises, if this model returns additional knowledge to the BoW model. In fact, some few landmarks (33% of the tested landmarks) benefit from the BoP model, others not. A detailed analysis of this and a suitable combination of both models are matters for further research beyond this work. Furthermore it would be interesting to compare our state-of-the-art approach with a system which bases on an inverted file index working directly on local features.

References

1. Deng, J., Berg, A. C., Li, K., Fei-Fei, L.: What does classifying more than 10,000 image categories tell us? In: Proceedings of the 11th European conference on Computer vision (ECCV'10), 2010
2. Gammeter, S., Bossard, L., Quack, T., Gool, L.V.: I know what you did last summer: object-level auto-annotation of holiday snaps. In: IEEE international conference on computer vision (ICCV), 2009
3. Avrithis, Y., Kalantidis, Y., Tolias, G., Spyrou, E.: Retrieving landmark and non-landmark images from community photo collections. In: Proceedings of the international conference on Multimedia (MM '10). ACM, New York, 2010
4. Zheng, Y., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Tat-Seng Chua, Neven, H.: Tour the world: Building a web-scale landmark recognition engine. In: Computer Vision and Pattern Recognition (CVPR), 2009
5. Philbin, J., Zisserman, A.: Object Mining Using a Matching Graph on Very Large Image Collections. In: ICVGIP, 2008
6. Crandall, D. J., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world's photos. In Proceedings of the 18th international conference on World wide web (WWW '09). ACM, New York, 2009
7. Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. In: International Journal of Computer Vision, 2004
8. Zheng, Q. F., Gao, W.: Constructing visual phrases for effective and efficient object-based image retrieval. In ACM Trans. Multimedia Comput. Commun. Appl. 5, October 2008

Evaluating Assumptions about Social Tagging ^{*}

A Study of User Behavior in BibSonomy

Stephan Doerfel¹, Daniel Zoller², Philipp Singer³, Thomas Niebler²,
Andreas Hotho², and Markus Strohmaier^{3,4}

¹ ITeG & Knowledge and Data Engineering Group, University of Kassel (Germany)
`doerfel@cs.uni-kassel.de`

² Data Mining and Information Retrieval Group, University of Würzburg (Germany)
`{zoller, niebler, hotho}@informatik.uni-wuerzburg.de`

³ GESIS (Germany) `{philipp.singer, markus.strohmaier}@gesis.org`

⁴ University of Koblenz (Germany)

Abstract. Social tagging systems have established themselves as an important part in today’s web and have attracted the interest of our research community in a variety of investigations. Henceforth, several assumptions about social tagging systems have emerged on which our community also builds their work. Yet, testing such assumptions has been difficult due to the absence of suitable usage data in the past. In this work, we investigate and evaluate four assumptions about tagging systems by examining live server log data gathered from the public social tagging system BibSonomy. Our empirical results indicate that while some of these assumptions hold to a certain extent, other assumptions need to be reflected in a very critical light.

1 Introduction

Social tagging systems such as BibSonomy, Delicious or Flickr have attracted the interest of our research community for almost a decade. While previous research has significantly expanded our expertise to describe [4] and model [2], social tagging systems, the community has also built their work on certain assumptions about usage patterns in these systems, which have emerged over time. For such assumptions, arguments and evidence have been discussed, though it is not clear to which degree they remain valid in actual tagging systems. Only a few studies have analyzed user behavior in social tagging systems to better understand such assumptions, either by (i) conducting user surveys (e.g., [5]) or by (ii) tapping into the rich corpus of tagging data (i.e., the posts) that is available on the web (e.g., [2]). However, such studies lack of detailed data how users actually

^{*} Extended Abstract for Work-in-Progress.

Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

request information. In this paper we overcome these drawbacks by presenting and thoroughly investigating a detailed usage log of the real-world, open social tagging system BibSonomy.⁵

2 Assumptions and Results

The Social Assumption. Assuming that social tagging systems are social, we measure to which degree users collaboratively share resources and we discuss evidence for the interest of users in the content of others. Details of this analysis can be found in [3].

The Retrieval Assumption. For the retrieval assumption we investigate whether users store resources in BibSonomy for later retrieval. We discover that while users post a large number of resources and tags to BibSonomy, they only retrieve a rather small fraction of them later.

The Equality Assumption. The equality assumption claims that the three sets of entities in a tagging system – users, tags, and resources – are equally important for navigation and retrieval. However, we find a strong *inequality* in the use of these entity sets: in BibSonomy, requests to user pages dominate the number of requests to tags and to resources.

The Popularity Assumption. Finally, we test whether the popularity of users, tags, and resources in posts is matched by their popularity in retrieval. We observe common usage patterns in posting and requesting behavior on an aggregate level. The patterns are less pronounced on an individual level.

Acknowledgments. This work is in part funded by the DFG through the PoSTs II project.

References

1. Benz, D., Hotho, A., Jäschke, R., Krause, B., Mitzlaff, F., Schmitz, C., Stumme, G.: The social bookmark and publication management system BibSonomy. *The VLDB Journal* 19(6), 849–875 (Dec 2010)
2. Cattuto, C., Schmitz, C., Baldassarri, A., Servedio, V.D.P., Loreto, V., Hotho, A., Grahl, M., Stumme, G.: Network properties of folksonomies. *AI Communications Journal*, Special Issue on “Network Analysis in Natural Sciences and Engineering” 20(4), 245–262 (2007)
3. Doerfel, S., Zoller, D., Singer, P., Niebler, T., Strohmaier, M., Hotho, A.: How social is social tagging? In: *Proceedings of the 23rd International World Wide Web Conference. WWW 2014*, ACM, New York, NY, USA (2014)
4. Golder, S.A., Huberman, B.A.: Usage patterns of collaborative tagging systems. *Journal of information science* 32(2), 198–208 (April 2006)
5. Heckner, M., Heilemann, M., Wolff, C.: Personal information management vs. resource sharing: Towards a model of information behaviour in social tagging systems. In: *Proceedings of the 3rd International Conference on Weblogs and Social Media. ICWSM '09*, San Jose, CA, USA (May 2009)

⁵ <http://www.bibsonomy.org/>, see [1] for a detailed description and various analyses.

Maintenance of distributed case-based reasoning systems in a multi-agent system*

Pascal Reuss and Klaus-Dieter Althoff
pascal.reuss@dfki.de
klaus-dieter.althoff@dfki.de

Intelligent Information Systems Lab, University of Hildesheim
Competence Center for Case Based Reasoning, German Center for Artificial Intelligence,
Kaiserslautern

Abstract. In many knowledge-based systems the used knowledge is distributed among several knowledge sources. Knowledge maintenance of such systems has several challenges to be met. This paper gives a short overview of a maintenance approach using so-called Case Factories to maintain knowledge sources and considering the dependencies between these sources. Furthermore we present a concept how our maintenance approach can be applied to a multi-agent system with several case-based reasoning systems.

1 Introduction

When maintaining the knowledge among distributed case-based reasoning (CBR) systems the dependencies between the knowledge sources are of crucial importance. For maintaining a single CBR system there are also several approaches that deal with maintaining the case base, the similarity, or the adaptation knowledge. In general all the knowledge sources belonging to a knowledge-based system have to be considered, too. This paper describes a multi-agent system, based on the SEASALT architecture, that is extended with several agents to apply the Case Factory approach. We describe the tasks of every required agent and the communication between them. In addition we present the required agents for the explanation capabilities. Section 2 describes related work to knowledge maintenance. In Section 3 the agents required for applying the Case Factory approach to a multi-agent system are described. In Section 4 a short conclusion is given.

1.1 SEASALT architecture

The SEASALT (Shared Experience using an Agent-based System Architecture Layout) architecture is a domain-independent architecture for extracting, analyzing, sharing, and providing experiences [[5]]. The architecture is based on the Collaborative Multi-Expert-System approach [1][2] and combines several software engineering and

* *Copyright* © 2014 by the paper's authors. *Copying permitted only for private and academic purposes.* In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

artificial intelligence technologies to identify relevant information, process the experience and provide them via an interface. The knowledge modularization allows the compilation of comprehensive solutions and offers the ability of reusing partial case information in form of snippets. Figure 1 gives an overview over the SEASALT architecture.

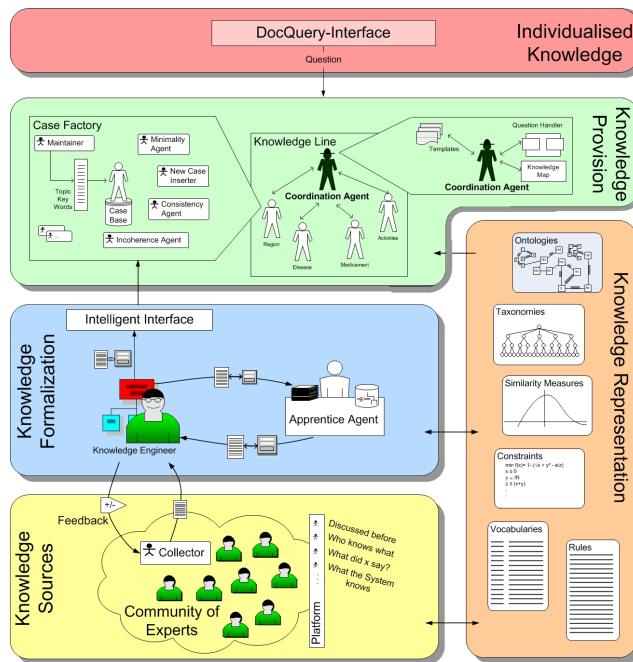


Fig. 1. Overview of the SEASALT architecture

The SEASALT architecture consists of five components: the knowledge sources, the knowledge formalization, the knowledge provision, the knowledge representation, and the individualized knowledge. The knowledge sources component is responsible for extracting knowledge from external knowledge sources like databases or web pages and especially Web 2.0 platforms. These knowledge sources are analyzed by so-called Collector Agents, which are assigned to specific Topic Agents. The Collector Agents collect all contributions that are relevant for the respective Topic Agent's topic [5]. The knowledge formalization component is responsible for formalizing the extracted knowledge from the Collector Agents into a modular, structural representation. This formalization is done by a knowledge engineer with the help of a so-called Apprentice Agent. This agent is trained by the knowledge engineer and can reduce the workload for the knowledge engineer [5]. The knowledge provision component contains the so called Knowledge Line. The basic idea is a modularization of knowledge analogous to the modularization of software in product lines. The modularization is done among the

individual topics that are represented within the knowledge domain. In this component a Coordination Agent is responsible for dividing a given query into several sub queries and pass them to the according Topic Agent. The agent combines the individual solutions to an overall solution, which is presented to the user. The Topic Agents can be any kind of information system or service. If a Topic Agent has a CBR system as knowledge source, the SEASALT architecture provides a Case Factory for the individual case maintenance. [5][4] The knowledge representation component contains the underlying knowledge models of the different agents and knowledge sources. The synchronization and matching of the individualized knowledge models improves the knowledge maintenance and the interoperability between the components. The individualized knowledge component contains the web-based user interfaces to enter a query and present the solution to the user.[5]

2 Related work

This section contains related work from other authors with focus on the maintenance of the knowledge containers of CBR systems and the maintenance of distributed knowledge in CBR systems. There exist several approaches to maintain the knowledge containers of a CBR system. For the maintenance of a case base various strategies were developed for example by [9], [10], [12], [13], [18], [17], [19] and [22]. [20] and [11] describe approaches to maintain the similarity measures within a CBR system. All this approaches are set up to maintain knowledge containers of a single CBR system. They neither consider the use of multiple CBR systems nor the dependencies between the knowledge containers of different CBR systems. All mentioned maintenance strategies could be applied within a Case Factory, but have to be embedded in an overall maintenance strategy managed by the Case Factory Organization.

Geissbuhler and Miller describe in their paper an approach for maintaining distributed knowledge bases in a clinical decision support system called WizOrder. Contrary to our approach, the maintenance in the WizOrder system is not done by one knowledge engineer, but by many different users of the system, like house staff, physicians, and nurses. The knowledge sources in the decision support system are heterogeneous and not homogenous as intended in our approach. Therefore many different tools for maintenance are used, each one with a specific interface for the respective user. The local knowledge bases are maintained by the users and an expert integrates the maintenance actions into the central knowledge base called knowledge library. From this knowledge library the cumulative changes are provided to the local knowledge bases. While this is done by human users and experts within the WizOrder system, in our approach we use software agents to suggest maintenance actions and central planning and supervising agents to generate a maintenance plan. This plan has still to be checked by a human knowledge engineer. [8]

Ferrario and Smyth described an approach for collaborative maintenance of a case base. The feedback of several users is evaluated and an appropriate maintenance action derived. When we compare our approach to theirs, our agents could be seen as users, that gives feedback and suggest maintenance actions. A Case Factory, maintaining one

CBR system could be compared to the collaborative maintenance. One difference between the approaches is that our approach is extended with maintenance capabilities for several CBR systems.[6][7]

3 Maintenance of distributed case-based reasoning systems

This section gives a short overview over the idea of the Case Factory (CF) and the Case Factory Organization (CFO). Then the software agents for the realization of the CF and CFO are described. At last the software agents for the explanation capabilities are described.

3.1 Agents of the Case Factory

Three types of software maintenance can be distinguished: corrective, adaptive and perfective maintenance. Corrective maintenance deals with processing failures, performance failures and implementation failures. Processing failures are situations like abnormal termination of an application. Performance failures deals with situations where the application violates defined performance constraints like to long response time. Implementation failure can lead to processing and performance failures, but may also be have no effect on the system. Adaptive maintenance deals with changes in the environment of an application and aims at avoiding failures caused by the change of an application environment. Perfective maintenance cover all actions that are performed to eliminate processing inefficiencies, enhance performance or improve the maintainability. This type of maintenance aims at keeping an application running at less expense or running to better serve the users needs [21]. [16] defines the knowledge maintenance of CBR systems as the combination of technical and associated administrative actions that are required to preserve the knowledge of a CBR system, or to restore the knowledge of the system to provide the intended functionality. This maintenance actions include also actions to adapt an CBR system to environment changes and enhance the performance.

The SEASALT architecture supports the maintenance of a CBR system with the help of a Case Factory. The original idea is from Althoff, Hanft and Schaaf [3] and the concept was extended by Reuss and Althoff [14]. The CF approach and the SEASALT architecture support the maintenance of distributed knowledge sources in multi-agent systems and the CF is intended to perform corrective maintenance as well as adaptive and perfective maintenance. Feedback from users about false solutions may lead to corrective maintenance actions, while the evaluation of the knowledge in a CBR system may lead to adaptive or perfective maintenance. The extended CF supports the maintenance of a single CBR system. It contains several software agents responsible for the evaluation and the maintenance of the case-based reasoning knowledge containers. This knowledge containers were introduced by [15].

The idea behind the Case Factory approach is maintenance of knowledge sources should consider the dependencies between the knowledge containers in a CBR system and the dependencies between the knowledge containers of different CBR systems.

There are dependencies between the vocabulary and the case base in a single CBR system or between case bases in different CBR systems. Changing the knowledge in one knowledge container may cause inconsistencies. Therefore additional maintenance actions may be necessary to restore the consistency of the knowledge.

To apply the CF approach to the a multi-agent system with CBR system nine agents are required: four monitoring and evaluation agents, each one responsible for one knowledge container (case base, vocabulary, similarity, and adaptation), and four maintenance agents, each one responsible for processing individual maintenance actions for the respective knowledge container. We propose an individual monitoring and evaluation agent for each knowledge container to process the monitoring and evaluation tasks in parallel. In addition it will be possible to activate and deactivate the monitoring and evaluation of a knowledge container during runtime by starting or shutting down the associated agent without affecting the monitoring and evaluation of the other knowledge containers. The last new agent is a supervising agent that coordinates the monitoring and evaluation of the knowledge containers and the processing of maintenance actions. In addition the agent communicates with the high-level Case Factory Organization. Figure 2 shows these agents in a multi-agent system.

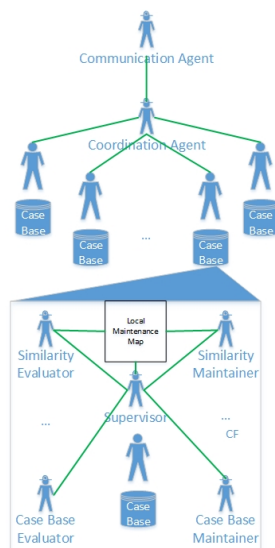


Fig. 2. Multi-agent system with Case Factory agents

In the following the tasks, permissions and responsibilities of the agent roles are described in GAIA notation [23]. Protocols and activities define the communication with other roles and the tasks a role can perform. The permissions are used to describe the knowledge a role has access to and the knowledge a role can change or generate. At last the responsibilities are used to describe the life cycle of a role. It is defined in which

order the protocols and activities are performed and if there are repetitions of protocols or activities. A (*) means, that a protocol or activity is performed 0 to n times, a (+) that a protocol or activity is performed 1 to n times. The exponent at the end of the liveness responsibilities describes the times the the whole process is performed. ω means it is repeated endlessly.

| | |
|----------------------------------|--|
| <i>Role Schema: Evaluator</i> | |
| <i>Description:</i> | <i>This role is responsible for monitoring and evaluating of a knowledge container of a cbr system. The evaluation results and derived maintenance actions are sent to the Supervisor.</i> |
| <i>Protocols and Activities:</i> | <i>MonitorKnowledgeContainer, EvaluateKnowledgeContainer, DeriveMaintenanceActions, SendEvaluationResults, SendDerivedMaintenanceActions</i> |
| <i>Permissions:</i> | <i>reads knowledge container, local maintenance map generates evaluation results, maintenance actions</i> |
| <i>Responsibilities</i> | <i>Liveness: COMMUNICATOR = (MONITORKNOWLEDGECONTAINER*.EvaluateKnowledgeContainer.DeriveMaintenanceActions.SendEvaluationResults, SendDerivedMaintenanceActions)⁰</i> |
| <i>Safety:</i> | <i>NONE</i> |

Fig. 3. Role schema Evaluator in Gaia notation

| | |
|----------------------------------|--|
| <i>Role Schema: Maintainer</i> | |
| <i>Description:</i> | <i>This role gets the confirmed maintenance actions from the Supervisor and processes these maintenance actions. It notifies the Supervisor about the result of the actions.</i> |
| <i>Protocols and Activities:</i> | <i>GetConfirmedMaintenanceActions, PerformMaintenanceActions, NotifySupervisor</i> |
| <i>Permissions:</i> | <i>reads confirmed maintenance actions, local maintenance map generates notification</i> |
| <i>Responsibilities</i> | <i>Liveness: MAINTAINER = (GETCONFIRMEDMAINTENANCEACTIONS.PERFORMMAINTENANCEACTIONS. NOTIFYSUPERVISOR)⁰</i> |
| <i>Safety:</i> | <i>NONE</i> |

Fig. 4. Role schema Maintainer in GAIA notation

Both generic roles are specialized for the specific agents and generic terms are substituted with the concrete knowledge container. Both roles have access to a local maintenance map, which contains information about available and preferred evaluation strategies and maintenance actions as well as evaluation metrics to compare the results to the maintenance goals. This way several evaluation strategies can be defined and applied to an agent.

| | |
|----------------------------------|---|
| <i>Role Schema: Supervisor</i> | |
| <i>Description:</i> | <i>This role supervises the monitoring and evaluation of a CBR system and the processing of maintenance actions. It also communicates with the Collector in the Case Factory Organization to send the suggested maintenance action and receive the confirmed actions.</i> |
| <i>Protocols and Activities:</i> | <i>GetEvaluationResults, GetMaintenanceSuggestions, SendMaintenanceSuggestions, GetConfirmedActions, SendConfirmedActions</i> |
| <i>Permissions:</i> | <i>reads evaluation results, maintenance suggestions, confirmed maintenance actions</i> |
| <i>Responsibilities</i> | <i>Liveness: SUPERVISOR = (GETEVALUATIONRESULTS*.GETMAINTENANCESUGGESTIONS*.SENDMAINTENANCESUGGESTIONS.GETCONFIRMEDACTIONS.SENDCONFIRMEDACTIONS)⁹</i> |
| <i>Safety:</i> | <i>NONE</i> |

Fig. 5. Role schema Supervisor in GAIA notation

3.2 Agents of the Case Factory Organization

While a Case Factory is able to maintain a single CBR system a high-level Case Factory Organization is required to coordinate the actions of all Case Factories and take the dependencies between the single CBR systems into account. This CFO consists of several additional software agents to supervise the communication between the Case Factories and the adherence of high level maintenance goals. Additionally, agents collect the maintenance suggestions from the Case Factories and derive a maintenance plan from all single maintenance suggestions. The agents are also responsible for checking constraints or solving conflicts between individual maintenance suggestions. In addition, a maintenance suggestion may trigger follow-up maintenance actions based on the dependencies between the CBR systems. The concept of the CFO allows to realize as many CFs and layers of CFOs as required. A multi-agent system can be divided into layers and each layer can have its own Case Factory Organization. This way a hierarchy of CFOs can be established that is scalable and supports multi-agent systems with many agents and layers. [14]

Each required Case Factory Organization consists of four software agents. A Collector Agent, a Maintenance Planning Agent, a Goal Monitoring Agent and a Team Supervisor Agent. For the assumed MAS only one Case Factory Organization level is required. Figure 6 shows the multi-agent system with the the additional agents for the Case Factory Organization.

Inside the CF agents evaluating the knowledge containers and derive maintenance suggestions from the result with the help of the local maintenance map (1). The results and the derived maintenance actions are send to the supervisor (2). The supervisor passed the maintenance actions to the collector (3). This collector gets the derived maintenance actions from all Case Factories and sends them to the goal monitoring agent. The goal monitoring agent is responsible for checking the maintenance actions against constraints from the team maintenance map. If no constraints are violated the maintenance actions are sent to the maintenance planner (5). This agent generates a plan from the maintenance actions. During the planning process it is possible to generate new maintenance actions based on the dependencies between different CBR systems. The

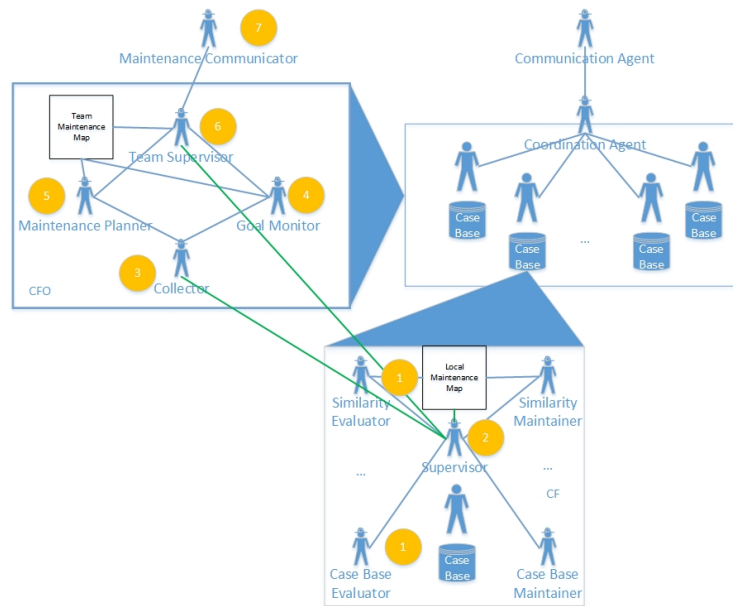


Fig. 6. docQuery multi-agent system with Case Factory Organization agents

maintenance plan is sent to the team supervisor (6). This agent checks the plan against constraint violation like the goal monitor does for individual actions. The checked plan is sent to the maintenance communicator and shown to a knowledge engineer (7). The knowledge engineer checks the plan and confirms the maintenance actions to be performed. He can also eliminate actions from the plan. The confirmed plan is sent back to the team supervisor in the CFO, the supervisor in the CF and the single maintenance actions to the maintaining agents.

Our concept for the Case Factory Organization includes explanation capabilities of the maintenance actions and the maintenance plan. The idea is to provide a set of explanations to support the knowledge engineer's understanding of the suggested maintenance plan and single actions. The idea is to use explanation templates that are filled with logging information. These templates consists of several text modules in human natural language. This way we try to use the systems logging information to generate human readable explanations.

To achieve this goal, the multi-agent system has to log all communication and actions of all agents, as well as evaluation results, feedback, constraint checks, and denied maintenance actions. From this logged information explanations should be extracted and combined for each maintenance action and the maintenance plan itself. Three additional roles are required to provide simple explanations: Logger, Logging Supervisor, Explainer. For each role at least one agent in the docQuery multi-agent system will be implemented. For several roles like the Logger or the Evaluator more than one instance is required. Some of the described roles and the respective agents can be combined in agent teams. For example, for a Case Factory a team of four Evaluators, four Main-

ainers and one Supervisor is required. Adding a new Case Factory will require the creation of nine software agents. Other roles like the Logger or the Explainer and its respective agents can be added as single agents. This way the multi-agent system has a high scalability and agents can be created and removed based on the tasks the single agents or the agent team are designed for.

Figure 7 shows the multi-agent system with all agents for the CF, CFO, and explanations. In the figure only the communication with the new agents is illustrated.

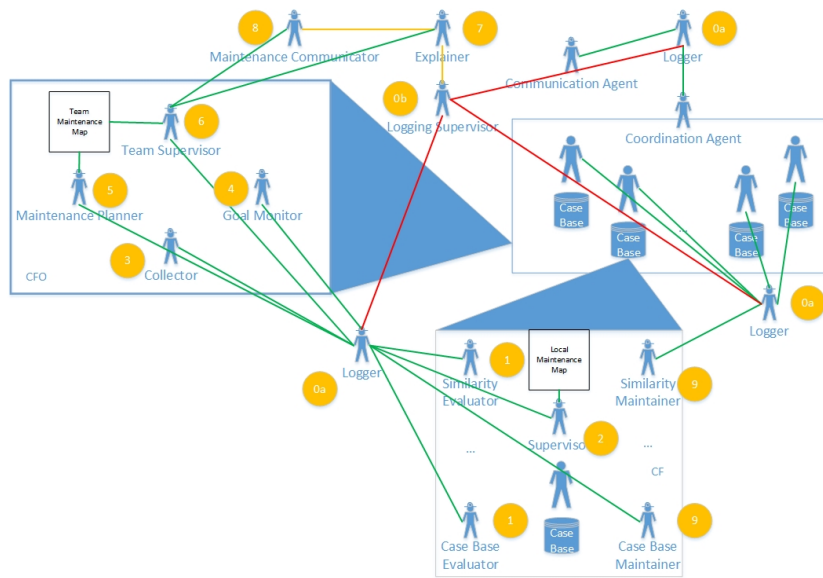


Fig. 7. MAS with CF, CFO and explanation agents

Several agents are responsible for logging the communication and performed tasks of the agent in the multi-agent system (0a) and the logged information are sent to the logging supervisor (0b). These information are used to generate explanations for suggested maintenance actions. Steps 1 till 6 are the same as described above. In addition, the checked plan is sent to the explanation agent (7). This agent uses the logged information to enrich the maintenance plan with explanations. The enriched plan is sent to the maintenance communicator and shown to a knowledge engineer (8). The knowledge engineer checks the plan and confirms the maintenance actions to be performed. He can also eliminate actions from the plan. The confirmed plan is sent back to the team supervisor in the CFO, the supervisor in the CF and the single maintenance actions to the maintaining agents (9).

4 Summary and Outlook

In this paper we presented the concept for a multi-agent system in the travel medicine domain with software agents for distributed maintenance with explanation capabilities. We gave an short overview of the Case Factory and Case Factory Organization and described the required tasks of the individual agent roles. The roles do not describe any concrete implementation of tasks or communications. The realization of the described concept and the implementation of the agents within a multi-agent system is the next step in our research. We will implement the single agents and agent teams as well as evaluation and maintenance strategies and evaluate the extended multi-agent system.

References

1. Althoff, K.D.: Collaborative multi-expert-systems. In: Proceedings of the 16th UK Workshop on Case-Based Reasoning (UKCBR-2012), located at SGAI International Conference on Artificial Intelligence, December 13, Cambridge, United Kingdom. pp. 1–1 (2012)
2. Althoff, K.D., Bach, K., Deutsch, J.O., Hanft, A., Mänz, J., Müller, T., Newo, R., Reichle, M., Schaaf, M., Weis, K.H.: Collaborative multi-expert-systems – realizing knowledge-product-lines with case factories and distributed learning systems. In: Baumeister, J., Seipel, D. (eds.) KESE @ KI 2007. Osnabrück (Sep 2007)
3. Althoff, K.D., Hanft, A., Schaaf, M.: Case factory: Maintaining experience to learn. In: Proceedings of the 8th European conference on Advances in Case-Based Reasoning. pp. 429–442 (2006)
4. Althoff, K.D., Reichle, M., Bach, K., Hanft, A., Newo, R.: Agent based maintenance for modularised case bases in collaborative multi-expert systems. In: Proceedings of the AI2007, 12th UK Workshop on Case-Based Reasoning (2007)
5. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013), dr. Hut Verlag München
6. Ferrario, M.A., Smyth, B.: A user-driven distributed maintenance strategy for large-scale case-based reasoning systems. In: ECAI Workshop Notes. pp. 55–63 (2000)
7. Ferrario, M.A., Smyth, B.: Distributing case-based maintenance: The collaborative maintenance approach. *Computational Intelligence* 17(2), 315–330 (2001)
8. Geissenbuhler, A., Miller, R.A.: Distributing knowledge maintenance for clinical decision-support systems: The "knowledge library" approach. In: Proceedings of the AMIA Symposium. pp. 770–774 (1999)
9. Iglezakis, I.: The conflict graph for maintaining case-based reasoning systems. In: Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning (2001)
10. Iglezakis, I., Roth-Berghofer, T.: A survey regarding the central role of the case base for maintenance in case-based reasoning. In: ECAI Workshop Notes. pp. 22–28 (2000)
11. Patterson, D., Anand, S., Hughes, J.: A knowledge light approach to similarity maintenance for improving case-base competence. In: ECAI Workshop Notes. pp. 65–78 (2000)
12. Racine, K., Yang, Q.: Maintaining unstructured case bases. In: Case-Based Reasoning and Development. pp. 553–564 (1997)
13. Racine, K., Yang, Q.: Redundancy and inconsistency detection in large and semi-structured case bases. *IEEE Transactions on Knowledge and Data Engineering* (1998)
14. Reuss, P., Althoff, K.D.: Explanation-aware maintenance of distributed case-based reasoning systems. In: LWA 2013. Learning, Knowledge, Adaptation. Workshop Proceedings. pp. 231–325 (2013)

15. Richter, M.M.: Introduction. chapter 1 in case-based reasoning technology - from foundations to applications. Inai 1400, springer (1998)
16. Roth-Berghofer, T.: Knowledge maintenance of case-based reasoning systems. The SIAM methodology. Akademische Verlagsgesellschaft Aka GmbH (2003)
17. Smyth, B.: Case-based maintenance. In: Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (1998)
18. Smyth, B., Keane, M.: Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. pp. 377–382 (1995)
19. Smyth, B., McKenna, E.: Competence models and the maintenance problem. Computational Intelligence (2001)
20. Stahl, A.: Learning feature weights from case order feedback. In: Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning (2001)
21. Swanson, E.B.: The dimensions of maintenance. In: Proceedings of the 2nd International Conference on Software Engineering. pp. 492–497 (1976)
22. Wilson, D.: Case-Based Maintenance: The Husbandry of Experience. Ph.D. thesis, Faculty of the University Graduate School, Department of Computer Science, University of Indiana (2001)
23. Wooldridge, M., Jennings, N., Kinney, D.: The gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems 3, 285–312 (2000)

**KDML: Workshop on Knowledge
Discovery, Data Mining and Machine
Learning**

Dyad Ranking Using a Bilinear Plackett-Luce Model (Abstract)*

Dirk Schäfer¹ and Eyke Hüllermeier²

¹ University of Marburg, Germany
dirk.schaefer@uni-marburg.de

² Department of Computer Science
University of Paderborn, Germany
eyke@upb.de

Preference learning is an emerging subfield of machine learning, which deals with the induction of preference models from observed or revealed preference information [2]. Such models are typically used for prediction purposes, for example, to predict context-dependent preferences of individuals on various choice alternatives. Depending on the representation of preferences, individuals, alternatives, and contexts, a large variety of preference models are conceivable, and many such models have already been studied in the literature.

A specific type of preference learning problem is the problem of *label ranking*, namely the problem of learning a model that maps instances to rankings (total orders) over a finite set of predefined alternatives [3]. An instance, which defines the context of the preference relation, is typically characterized in terms of a set of attributes or features; for example, an instance could be a person described by properties such as sex, age, income, etc. As opposed to this, the alternatives to be ranked, e.g., the political parties of a country, are only identified by their name (label), while not being characterized in terms of any properties or features.

In practice, however, information about properties of the alternatives is often available, too, and such information could obviously be useful from a learning point of view. Motivated by this observation, we introduce *dyad ranking* as a generalization of the label ranking problem. In dyad ranking, not only the instances but also the alternatives are represented in terms of attributes. For learning in the setting of dyad ranking, we propose an extension of an existing label ranking method based on the Plackett-Luce model, a statistical model for rank data [1]. First experimental studies with real and synthetic data confirm the usefulness of the additional feature information of alternatives.

References

1. W. Cheng, K. Dembczyński, and E. Hüllermeier. Label ranking methods based on the Plackett-Luce model. In *Proceedings ICML, 27th International Conference on Machine Learning*, pages 215–222, Haifa, Israel, 2010.

* Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

2. J. Fürnkranz and E. Hüllermeier, editors. *Preference Learning*. Springer, 2011.
3. S. Vembu and T. Gärtner. Label ranking: A survey. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*. Springer, 2011.

Improved Questionnaire Trees for Active Learning in Recommender Systems

Rasoul Karimi¹, Alexandros Nanopoulos², Lars Schmidt-Thieme¹

¹ Information Systems and Machine Learning Lab Marienburger Platz 22 University of Hildesheim 31141 Hildesheim Germany

`karimi, schmidt-thieme@ismll.uni-hildesheim.de`

² Department of Business Informatics., Schanz 49, University of Eichstatt-Ingolstadt, 85049 Ingolstadt, Germany
`nanopoulos@ku.de`

Abstract. A key challenge in recommender systems is how to profile new-users. This problem is called cold-start problem or new-user problem. A well-known solution for this problem is to use active learning techniques and ask new users to rate a few items in order to reveal their preferences. Recently, questionnaire trees (tree structures) have been proposed to build such adaptive questionnaires. In this paper, we improve the questionnaire trees by splitting the nodes of the trees in a finer-grained fashion. Specifically, the nodes are split in a 6-way manner instead of 3-way split. Furthermore, we compare our approach to on-line updating and show that our method outperforms online updating in order to fold-in the new user into recommendation model. Finally, we develop three simple baselines based on the questionnaire trees and compare them against the state-of-the-art baseline to show that the new-user problem in recommender systems is tough and demands a mature solution.

1 Introduction

Recommender systems help web users to address information overload in a large space of possible options [1]. Collaborative filtering is the traditional technique for recommender systems. Evidently, the performance of collaborative filtering depends on the amount of information that users provide regarding items, most often in the form of ratings. This problem is amplified for new users because they have not provided any rating which impacts negatively on the quality of generated recommendations. A simple and effective way to overcome this problem, is by posing queries to new users in order that they express their preferences about selected items, e.g., by rating them. Nevertheless, the selection of items must take into consideration that users are not willing to answer a lot of such

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

queries. To address this problem, *active learning* methods have been proposed to acquire the most informative ratings, i.e ratings from users that will help most in determining their interests [3, 4].

Recently, active learning based on tree structures have been proposed by (Golbandi et al. [2]). In [2], the tree structures are ternary because there are three possible answers for queries: "Like", "Dislike", or "Unknown". In datasets like Netflix and MovieLens that the range of the ratings is from 1 to 5, ratings from 1 to 3 are considered as "Dislike" and ratings 4 and 5 are treated as "Like". Moreover, the missing ratings are considered as "Unknown", meaning users do not know the queried item, so they can not rate it.

Nevertheless, [2] was a breakthrough in the literature of active learning for recommender systems. In our previous paper [5], we improved [2] by incorporating matrix factorization into the tree structures and proposing a sampling method to speed up the tree construction algorithm. In this paper, we improve it one step further by upgrading the ternary trees to 6-way trees, meaning the nodes are split in a 6-way fashion. In the 6-way split, there is one child node per each rating from 1 to 5 and one child node for the "Unknown" response. As the 6-way split distinguishes users tastes more precisely, it is expected that the accuracy of the rating prediction also improves. On the other hand, the 6-way split might lead to overfitting, which affects adversely on the accuracy. Therefore, we need a rating prediction model that handles the overfitting issue very well. We apply the 6-way split to two prediction models and show the effect of the overfitting on the accuracy of the 6-way split.

2 Related Work

The idea of using decision trees for the cold-start recommendation was proposed by (Rashid et al. [7]). They tried to formalize the cold-start problem in a supervised learning context and solve it through decision trees. However, they face challenges that force them not to use standard decision tree learning algorithms such as ID3 and C4.5. (Golbandi et al. [2]) improved [7] by advocating a specialized version of decision trees to adapt the preference elicitation process to the new user's responses. As our method relies on [2], we briefly explain it in this section.

Here, each interior node is labeled with an item $i \in I$ and each edge with the user's response to item i . The new user preference elicitation corresponds to following a path starting at the root by asking the user to rate items associated with the tree nodes along the path and traversing the edges labeled by the users response until a leaf node is reached. Here, decision trees are ternary. Each internal tree node represents a single item on which the user is queried. After answering the query, the user proceeds to one of the three subtrees, according to her answer. The answer is either Like, Dislike, or Unknown. The Unknown means users are not able to rate the queried item because they do not know it. Letting users not to rate the queried items in case they do not know it, is crucial because it happens frequently in recommender systems.

Each tree node represents a group of users and predicts item ratings by taking the average of ratings among corresponding users. Formally, let t be a tree node and $U_t \subseteq U$ be its associated set of users. \mathcal{D}^t denotes a subset of $\mathcal{D}^{\text{train}}$ which belong to the node t :

$$\mathcal{D}^t := \{(u, i, r) \in U \times I \times R \mid u \in U_t\},$$

the profile of the item i in the node t is denoted as \mathcal{D}_i^t :

$$\mathcal{D}_i^t := \{(u, r) \in U \times R \mid u \in U_t\},$$

and the predicted rating of item i at the node t is computed using the item average method :

$$\hat{r}_{ti} = \frac{\sum_{(u,r) \in \mathcal{D}_i^t} r_{ui} + \lambda_1 \hat{r}_{si}}{|\mathcal{D}_i^t| + \lambda_1} \quad (1)$$

To avoid over-fitting, the prediction of the item i is regularized towards its prediction in the parent node r_{si} . λ_1 is the regularization factor. The effect of the regularization for the item i becomes more significant when the number of the ratings in the item profile \mathcal{D}_i^t is less. The squared error associated with node t and item i is: $(e_i^t)^2 = \sum_{(u,r) \in \mathcal{D}_i^t} (r - \hat{r}_{ti})^2$. Also, the overall squared error at node

t is: $(e^t)^2 = \sum_{i \in I} (e_i^t)^2$.

Building decision trees is done in a top-down manner. For each internal node, the best splitting item is the one which divides the users into three groups such that the total squared prediction error is minimized. This process continues recursively with each of the subtrees and at the end all users are partitioned among subtrees.

Suppose we are at node t . Per each candidate item i , three candidate child nodes are defined: $tL(i)$, $tD(i)$, $tU(i)$ representing users who like the item i , dislike it, and have not rated it respectively. The squared error associated with this item is $Err_t(i) = (e^{tL})^2 + (e^{tD})^2 + (e^{tU})^2$. Among all candidate items, the item which minimizes the following equation is the best :

$$splitter(t) = \operatorname{argmin}_{i \in I} Err_t(i) \quad (2)$$

A naive construction of the tree would be intractable if the number of items and ratings is large. Therefore, (Golbandi et al. [2]) proposes a solution for that. The idea is to expand "Unknown" child nodes in a different way using some statistics collected from "Like" and "Dislike" child nodes.

3 Problem Definition

Let U be a set (of users), I be another set (of items), and $R \subseteq \mathbb{R}$ be a (finite) set of ratings, e.g., $R := \{1, 2, 3, 4, 5\}$. Let $R^+ := R \cup \{.\}$ with an additional symbol

for a missing value. The triple $(u, i, r) \in U \times I \times R$ denotes the rating r of user u for item i .

For a data set $\mathcal{D} \subseteq U \times I \times R$ denote the set of all users occurring in \mathcal{D} by

$$U(\mathcal{D}) := \{u \in U \mid (u, i, r) \in \mathcal{D}\}$$

Subsets $E \subseteq I \times R$ are called user profiles. The profile of user u in \mathcal{D} is denoted by

$$\mathcal{D}_u := \{(i, r) \in I \times R \mid (u, i, r) \in \mathcal{D}\}$$

The rating of item $i \in I$ in user profile $E \subseteq I \times R$ is denoted by

$$r(i; E) := \begin{cases} r & , \text{ if } (i, r) \in E \\ \cdot & , \text{ else} \end{cases}$$

We define a questionnaire as a tree where each interior node is labeled with an item $i \in I$, each branch with a rating value $r \in R^+$ and each leaf node corresponds to a rating predictive model $\hat{r} : I \rightarrow R$, where the rating of each item can be predicted. For a user profile $E \subseteq I \times R$ let $\hat{R}(E)$ denote the rating predictive model at the leaf one arrives when starting at the root of the tree and iteratively from a node with label $i \in I$ proceeds to its child node with label $r(i; E)$ until a leaf node is reached.

Given

- a data set $\mathcal{D}^{\text{train}} \subseteq U \times I \times R$,
- a loss $\ell : R \times \mathbb{R} \rightarrow \mathbb{R}$, and
- a maximal number of queries N ,

the active learning for the new-user problem in recommender systems is to find a questionnaire \hat{R} of maximal depth N s.t. for another data set $\mathcal{D}^{\text{test}} \subseteq U \times I \times R$ (sampled from the same distribution, not being used during training, and with non-overlapping users, i.e., $U(\mathcal{D}^{\text{train}}) \cap U(\mathcal{D}^{\text{test}}) = \emptyset$) the average loss is minimal.

Users in $\mathcal{D}^{\text{test}}$ are supposed to be new users. For each $u \in \mathcal{D}^{\text{test}}$, \mathcal{D}_u is split into $\mathcal{D}_u^{\text{pool}}$ (pool data) and $\mathcal{D}_u^{\text{test}}$ (test data). $\mathcal{D}_u^{\text{pool}}$ is used to find the predictive model $\hat{R}(\mathcal{D}_u^{\text{pool}})$ at the leaf node and $\mathcal{D}_u^{\text{test}}$ is used to evaluate it. $\mathcal{D}_u^{\text{pool}}$ should also contain items with missing value, so

$$\mathcal{D}_u^{\text{pool}} = \mathcal{D}_u^{\text{pool}} \cup \{(u, i, \cdot) \mid i \in I, i \notin \mathcal{D}_u^{\text{pool}}\}$$

The total loss is the loss over all test users:

$$\ell(\mathcal{D}^{\text{test}}; \hat{R}) := \frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{u \in U(\mathcal{D}^{\text{test}})} \sum_{(i, r) \in \mathcal{D}_u^{\text{test}}} \ell(r, \hat{R}(\mathcal{D}_u^{\text{pool}})(i)) \quad (3)$$

What we call decision tree or questionnaire here really is a multivariate regression tree for instances in $(R^+)^I$ (here called user profiles) with values in \mathbb{R}^I . The values in \mathbb{R}^I can be represented by models $\hat{r} : I \rightarrow \mathbb{R}$. Other names could be rating prediction tree or recommendation tree.

4 Factorized Decision Trees

In [2], the ratings are predicted based on the item average method, which may seem naive as there are more advanced algorithms, such as Matrix Factorization (MF) [6], which have already shown their superiority over the item average. The reason for using the item average is that building the tree structures is expensive in terms of time. There are many nodes that need to be expanded and per each node there are many candidate items that must be checked. On the other hand, we have to predict ratings in child nodes and compute the error in order to find the best split item. As a result, we need a method for rating prediction that is fast, even though it may not be the best method. Otherwise, building the tree structures would be intractable.

Now the question is "how can we improve rating prediction of the tree structures while keeping its complexity low?" To find a solution for this question, (Karimi et al. [5]) proposed a method, which is called Factorized Decision Trees (FDT). The FDT divides the learning algorithm of the tree structures into two steps. In the first step, the *structure* of the tree structures is learned according to [2]. After constructing the tree, an MF model is trained to learn the *labels* of the tree, in which the labels are the rating predictions in the leaf nodes. In this way, we achieve a learning algorithm that is more accurate and scalable. (Karimi et al. [5]) do not exploit MF during the tree construction algorithm because it causes too much complexity. (Zhou et al. [9]) proposed another approach to incorporate matrix factorization into the tree structures. However, as it has been detailed in [5], it is too complex and is not scalable.

The scalability becomes even more important when we notice how information overload is growing up every day. Until a few years ago, Netflix was the largest data set for recommender systems. But now we have Yahoo Music, containing 717 M ratings, so it is more than 7 times bigger than Netflix. Therefore, we need to think about the scalability of our approaches. Otherwise, even active learning methods are accurate, it is not possible to apply them in big recommender systems.

5 Fine-Grained Questionnaire Trees

First of all, we would like to clarify that trees that have been used for cold-start recommendation are different from decision trees in machine learning. In decision trees, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. But trees in the cold-start problem are not acting as predictive model. They are simply tools that are used to visually and explicitly represent decisions, in which decisions are new users' responses to the queried items. In fact, decision trees in machine learning describe data while in cold-start problem they represent decisions. Therefore, we use *questionnaire trees* term to refer to such trees.

(Golbandi et al.[2]) opts for 3-way splits corresponding to three possible user responses ("Like", "Dislike", and "Unknown"). In datasets like Netflix and

MovieLens that the range of ratings is from 1 to 5, ratings from 1 to 3 are considered as "Dislike" and ratings 4 and 5 are treated as "Like". Moreover, the missing ratings are considered as "Unknown", meaning users do not know the queried item, so they can not rate it. However, it is expected that a more refined split, such as a 6-way split that matches five star levels plus an "unknown" would improve accuracy. The main bottleneck to do so is the overhead caused by increasing the number of nodes. The higher the number of splits, the higher the number of nodes, which requires more time to build questionnaire trees. To make this overhead more clear, we provide an example. Suppose that questionnaire trees are built up to level 2. Given a 3-way split, the total number of nodes is 13. But if the nodes are split based on the 6-way split, questionnaire trees would have 43 nodes. This overhead increases exponentially by increasing the number of queries.

Fortunately, (Karimi et al. [5]) have already proposed a sampling method that drastically speeds up the tree construction algorithm. This method is called Most Popular Sampling (MPS). Instead of checking all candidate items at each node, the MPS checks only those items that are most popular among users associated with the node. Given that MPS is used, the 6-way split can be used to improve the accuracy of rating predictions while the tree learning algorithm is still tractable.

When the new user preference elicitation ends and a couple of ratings are received from the new user, she is treated as a normal user like existing users of the recommender system. On the other hand, there is already a recommendation model for the existing users, which is usually MF. We call it warm MF since it is for users who already have enough ratings in the data set, in contrast to cold (new) users who have a few ratings. Now we need to fill the gap between the new users and the existing users by folding the new user into the warm MF. Specifically, we need to learn the latent features of the new user in the warm MF. A naive approach for doing this is to add the ratings of the new user to the original data set and then retrain the MF with the whole training data set. However, as we have to repeat this process for all new users, it would be very slow. Therefore, we have to switch to online updating. In online updating, using the ratings that the new user has given, only the new user's latent features are updated and the rest of the features including item features and other user features are not touched [8].

Fortunately, the FDT can already provide us with the new user's latent features and there is no need to use online updating. The FDT generates user features for each type of new users, which corresponds to the leaf nodes of decision trees. In this way we learn the new user features with a higher accuracy. Moreover, there is no need for an online updating step to bridge the query prediction model (decision trees) and the recommendation model (MF). In fact, this is a new fold-in approach, in which, given a new user with a few ratings, a subset of training users who have the same ratings like the new user are selected and then the new users features are trained using all ratings of these users. In our experiments, we found out that FDT can become even faster if only user

features are updated and the rest of the features are fixed to the warm MF. However, the accuracy is slightly affected.

6 Experimental set up

The main challenge in applying active learning for recommender systems is that users are not willing to answer many queries in order to rate the queried items. For this reason, we report the performance of all examined methods in terms of prediction error (RMSE) versus the number of queried items, which is simply denoted as $\#queries$. The RMSE of user u is computed as follows:

$$RMSE_u = \sqrt{\frac{1}{|D_u^{test}|} \sum_{(i,r) \in D_u^{test}} (r - \hat{r}_{ui})^2} \quad (4)$$

where D_u^{test} is the set of the test items of user u , \hat{r}_{ui} is the predicted rating of user u for item i , and r_{ui} is the true (actual) rating. Thus, we examine the problem of selecting at each step, the item for which each new user u will be queried to provide a rating. The item has to be selected in order to minimize the $RMSE$. The RMSE of each test user is measured separately and then the average RMSE over all test users is reported.

We report the performance of 6-way questionnaire trees based on two predictive models: item average (6-way-AVG) and matrix factorization (6-way-FDT). Correspondingly, we choose two baselines: 3-way-AVG [2] and 3-way-FDT [5].

We implemented [2] by ourself in java. First, we followed the same hyper-parameters reported in [2] to calibrate our results against it and make sure that our implementation was correct. Then, in our experiments, we changed one of the hyper-parameters: (Golbandi et al. [2]) do not expand nodes in which the the number of ratings is fewer than $\alpha = 200000$ and stops the learning. The goal is to save runtime. In our experiments, we set α to zero because Most Popular Sampling (MPS) [5] is already able to save runtime and there is no need to stop the learning. The results show that this setting is significantly beneficial. For $\alpha = 200000$, the RMSE is 0.971 after 5 queries but for $\alpha = 0$ the RMSE would be 0.958.

As (Golbandi et al. [2]) conduct their experiment on the Netflix data set, we also run our experiments on this dataset. Since the data set is large, the experiments are done in one fold, the same evaluation protocol as [2]. The dataset is already split into train and test datasets. However, this split is not suitable for cold-start evaluation protocol since users in the training and test sets are the same. As test users are considered as new users, they should not already appear in the training set. Therefore, we split all users into two disjoint subsets, the training set and the test set, containing 75% and 25% users, respectively. The tree is learned based on the ratings of training users in the training data. The ratings of training users in the original Netflix test split is considered as validation data in our experiments to find the hyper-parameters of MF. The users in the test set are assumed to be new users. The ratings of test users in the

Netflix training dataset are used to generate the user responses in the interview process. To evaluate the performance after each query, the ratings of test users in the Netflix test data are used.

We will also compare our work to three simple baselines. The goal of this comparison is assess the difficulties of the new-user problem. These three baselines are as follows:

- **Random:** At each node, the split item is selected randomly.
- **Local Most Popular (LMP):** At each node, the most popular item according to the users associated with the node is selected.
- **Global Most Popular (GMP):** First, s most popular items are found based on all ratings available in the dataset. Then we start to build questionnaire trees. All the nodes that are at level l are expanded using the l -th most popular item. In this way, the dynamic aspect of questionnaire trees is omitted and all new users, regardless of their responses to the queries, receive the same questions.

6.1 Results

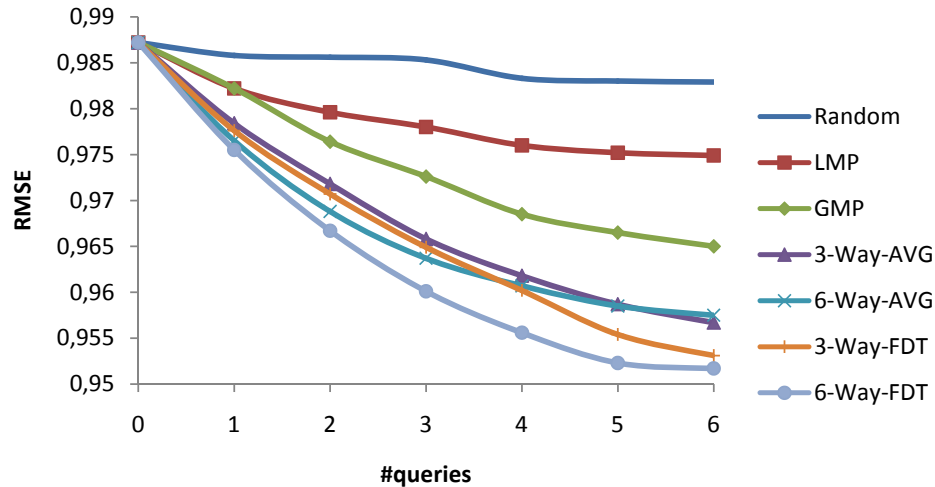


Fig. 1. RMSE results of 6-way split based on FDT (6-way-FDT), 6-way split based on item average (6-way-AVG), 3-way split based on FDT (3-way-FDT), and 3-way split based on item average (3-way-AVG).

Figure 1 shows the results of three simple baselines, 3-way-avg [2], 3-way-FDT [5], 6-way-AVG, and 6-way-FDT. All results are based on MPS where the sampling size is 200. First we discuss about the simple baselines. As the results

show, random item selection performs very badly and gains almost nothing after 8 queries. LMP doesn't work well either. Among the three simple baselines, GMP is the best, although it is still much worse than Bootstrapping. Table 1 shows some statistics which can justify these results. This table shows the probabilities of receiving different responses from new users by each method. The main reason that the random selection does not perform well is that it chooses items that will not be rated by new users. The probability that the random selection receives a rating is less than 0.01. When the new user does not rate the queried item, that new user is moved to the unknown child node. As the predictions in the unknown child node do not significantly differ from the predictions at the current node, this strategy is not able to improve the accuracy of predictions. Remember that test users and training users have the same distributions. If test (new) users do not know the split item, training users do not know it either. Therefore, decision trees which are built using training users with the random selection strategy are very imbalanced. This means that almost all users of the current node are moved to the unknown child node and consequently the predictions at the current node and the child nodes would be almost the same. LMP and GMP receive more ratings compared to the random selection, that is why their performance also improves in Figure 1.

Table 1. The probability that the new user likes the queried item (p_{like}), dislikes it ($p_{dislike}$), or does not rate it ($p_{unknown}$) for different active learning methods.

| Method | p_{like} | $p_{dislike}$ | $p_{unknown}$ |
|-----------------------|------------|---------------|---------------|
| Random | 0.004 | 0.003 | 0.993 |
| LMP | 0.18 | 0.16 | 0.66 |
| GMP | 0.26 | 0.17 | 0.57 |
| Bootstrapping and FDT | 0.18 | 0.15 | 0.67 |

Coming back to Figure 1, as we expected the 6-way-AVG beats 3-way-AVG because it provides more refined splits. 6-way-FDT further improves the 6-way-AVG since it leverages MF for rating prediction. The benefit of using MF for rating prediction instead of the item average is more clear in the 6-way split compared to the 3-way split. The reason is that the accuracy we gain at each level is the summation of the improvements of all users at the corresponding level. The higher the number of users, the larger the improvement. A grid search methodology was followed to find hyper-parameters, which are reported in table 2.

After 5 queries, 6-way-AVG converges to 3-way-AVG and even starts to become worse with the sixth query. This happens because, as we go down to the deeper layers of questionnaire trees, the number of associated users of nodes decreases. Therefore, the ratings in such nodes are predicted with less training data, which obviously adversely affects accuracy. Although the predictions are still regularized towards the predictions in the parent node, this regularization

might not be enough to compensate for the effect of less training data in such nodes. However, 6-way-FDT does not suffer from this problem because it does not use hierarchical regularization, instead it exploits typical ℓ_2 regularization. Due to the same reason, 3-way-FDT outperforms 6-way-AVG after 4 queries.

Regarding the running time, 6-way-FDT and 3-way-FDT are slower than 6-way-AVG and 3-way-AVG since these methods need to train a MF model. In our experiments, training a MF model takes around 3 hours, which considering the gained improvement, it pays off.

Table 2. Hyper-parameters of MF in 6-way-FDT in all levels. α is the learning rate and λ is the regularization factor.

| level | α | λ |
|-------|----------|-----------|
| 1 | 0.0011 | 0.016 |
| 2 | 0.0015 | 0.007 |
| 3 | 0.0013 | 0.005 |
| 4 | 0.0013 | 0.005 |
| 5 | 0.0013 | 0.009 |
| 6 | 0.0004 | 0.03 |

We finish this section by comparing the FDT to online updating [8]. To use online updating, first decision trees are built as in [2]. Then in the leaf nodes, the user features are retrained only based on the received ratings from the root node to the leaf node. Table 3 reflects the RMSE after each query. Clearly FDT outperforms online updating by a large margin. This happens because FDT uses all ratings of the training users who are similar to the new user to train the user features. However, online updating uses only a few ratings that are received from the new user. The more the number of ratings, the better the accuracy of the learned user features. Interestingly, online updating is even worse than Boot [2], which is based on item average prediction. However, as the Boot method does not provide the latent features, it cannot be used to fold the new user into the warm MF model.

Table 3. The RMSE of online updating in MF and FDT after each query

| | 0 | 1 | 2 | 3 | 4 | 5 |
|--------|--------|--------|--------|--------|--------|--------|
| Online | 1.056 | 1.0290 | 1.0056 | 1.0008 | 0.9948 | 0.9914 |
| FDT | 0.9872 | 0.9776 | 0.9707 | 0.9649 | 0.9602 | 0.9554 |

7 Conclusion

Using active learning to build adaptive questionnaire trees is the promising approach to address the cold-start problem in recommender systems. The performance of questionnaire trees can be improved by splitting the nodes in a finer-grained fashion, i.e. one child node per each possible rating (including the "Unknown" answer).

As the future work, we plan to use other data sets, in which the maximum rating is higher than 5. For example, in EachMovie, the range of ratings is from one to six, or in IMDb, it is from one to ten. The hypothesis is that opting for the higher number of splits, i.e. 7-way and 11-way splits respectively, may lead to a better accuracy. On the other hand, there might be limitation in the accuracy gained by increasing the number of splits. One needs to verify this hypothesis.

References

1. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.
2. N. Golbandi, Y. Koren, and R. Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *WSDM*, pages 595–604. ACM, 2011.
3. A. S. Harpale and Y. Yang. Personalized active learning for collaborative filtering. In *SIGIR*, pages 91–98. ACM, 2008.
4. R. Jin and L. Si. A bayesian approach toward active learning for collaborative filtering. In *UAI*, 2004.
5. R. Karimi, M. Wistuba, A. Nanopoulos, and L. Schmidt-Thieme. Factorized decision trees for active learning in recommender systems. In *25th IEEE International Conference on Tools With Artificial Intelligence (ICTAI)*, 2013.
6. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, 2009.
7. A. M. Rashid, G. Karypis, and J. Riedl. Learning preferences of new users in recommender systems: an information theoretic approach. *SIGKDD Explor. Newsl.*, 10(2):90–100, Dec. 2008.
8. S. Rendle and L. Schmidt-Thieme. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *ACM Conference on Recommender Systems (RecSys)*, pages 251–258. ACM, 2008.
9. K. Zhou, S.-H. Yang, and H. Zha. Functional matrix factorizations for cold-start recommendation. *SIGIR '11*, pages 315–324. ACM, 2011.

Archetypal Game Recommender Systems

Rafet Sifa¹, Christian Bauckhage^{1,2}, and Anders Drachen³

¹ Fraunhofer IAIS, Sankt Augustin, Germany

² University of Bonn, Bonn, Germany

³ Aalborg University, Aalborg, Denmark

Abstract. Contemporary users (players, consumers) of digital games have thousands of products to choose from, which makes finding games that fit their interests challenging. Towards addressing this challenge, in this paper two different formulations of Archetypal Analysis for Top-L recommender tasks using implicit feedback are presented: factor- and neighborhood-oriented models. These form the first application of recommender systems to digital games. Both models are tested on a dataset of 500,000 users of the game distribution platform Steam, covering game ownership and playtime data across more than 3000 games. Compared to four other recommender models (nearest neighbor, two popularity models, random baseline), the archetype based models provide the highest recall rates showing that Archetypal Analysis can be successfully applied for Top-L recommendation purposes.

Keywords: Game Data Mining, Recommender Systems, Behavior Analysis, Predictive Analytics, Business Intelligence, Game Analytics, Player Profiling

1 Introduction

Consumers of digital games are inundated with choices. Thousands of games are produced every year and available across a variety of platforms, from PC, console and mobile units, and across a broad design space. While getting solid numbers on the diversity of games and the number of units shipped is difficult due to the lack of disclosure of sales numbers in the game industry, as of June 2014, the game database site *MobyGames.com* has a total of 84,739 games in its archive. However, only commercial titles are included, and according to [13]: *The sheer amount of product ensures that no source can truly be definitive.* This is exemplified by [17] who in 2012 reported over 222,000 active games on the iOS AppStore alone. According to the same source, thousands of games are submitted to the AppStore every month (see Fig. 1). In terms of the games market, the Gartner Group [8] reported a *93 Billion global market for games (including mobile), pushing upwards of a predicted 100 Billion plus in 2014.* Irrespective of the

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

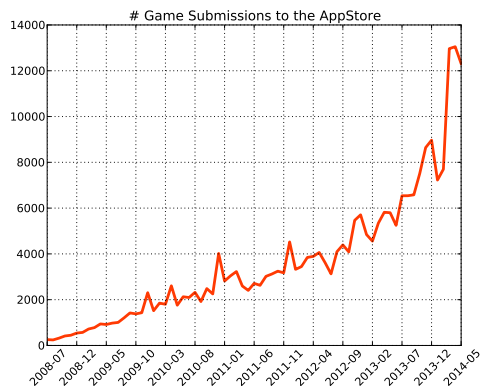


Fig. 1. The number of game applications submitted per month to the iTunes App Store. This is determined by the application release date. Source: www.pocketgamer.biz

specific numbers, there are unprecedented opportunities to meet the needs of a variety of players (consumers) in terms of matching needs with products. According to e.g. [12] this form of recommendation is key to enhance user satisfaction with products. In the context of machine learning, recommender systems are collections of supervised, unsupervised, reinforcement and hybrid learning techniques that aim to predict user’s preference or confidence level to a particular set of features based on the previously observed features [11, 10, 14, 18]. In essence, the goal is to provide personalized recommendations that suit the interests of the user in question. Recommender systems are virtually unheard of in digital games beyond basic notions such as popularity rankings. However, recommender systems are a topic of increasing interest, perhaps most importantly due to the recent growth in the number and type of digital games, which has made the issue of information search and selection increasingly serious. Additionally, *user acquisition* costs have increased in the game industry, and with the emergence of Free-to-play (F2P) business models, and the generally low conversion rates of non-paying to paying players, it has become important to devise new strategies to not only acquire but also qualify users, for example via churn analysis [9], and focus acquisition on those channels providing valuable users, e.g. users who are interested in playing the games in question [21]. Recommender systems are perhaps particularly useful for the kind of media products games constitute, because many players will have played the same games, and people often own multiple games [7]. Furthermore, due to the ability to track behavioral telemetry from games, it is possible to obtain implicit information about the level of engagement with a game, for instance, via measuring how much time a given player spends on a game [21]. This means that large-scale data are available on the appeal of specific games to specific customers. Implicit feedback can also be combined with explicit feedback from e.g. user rating systems.

In the background of this interest is a series of changes the game industry has gone through in recent years, which has added new sources of information about users and -behavior which can be used in recommender systems. These changes include the rapid rise of mobile games to become a substantial part of the marketplace [7], and a break from traditional retail-based business models which are today supplemented with notably F2P models. However, despite a remarkable growth in the number of games and the rapid emergence and advancement of business intelligence methods in game development collectively referred to as *game analytics* [21], the majority of the research on user (player) behavior in digital games has been confined to single games, limiting the broader application of the results, and the knowledge about practices in the industry is limited due to confidentiality issues [21]. Thus, in this paper, to the best knowledge of the authors, the first recommender system developed specifically for digital games is presented.

1.1 Related Work

Over the past few years the domain of game analytics has emerged to support the increasing needs for business intelligence solutions in the game industry [21]. While previous and current work studies touch various aspects of data mining for games, to the best knowledge of the authors there is no prior work done towards player/user based recommender systems for games. Recommender systems form a major topic of investigation outside the games domain, for example the e-commerce movement led by Amazon.com, or for the recommendation of movies, music and physical products [11, 12, 18, 14, 4]. In the below sections, the parts from existing relevant work on recommender systems and their evaluations is explained and related to on an ongoing basis, however, it is also valuable to briefly consider some of the work on large-scale data mining in digital games.

Cross-games analytics is a rare occurrence, in part due to the recent rapid emergence of the practice in the industry, the confidentiality associated with behavioral data and the lack of public datasets. Exceptions exist, such as Bauckhage et al. [1] and Chambers et al. [3], both focusing on analyzing playtime distributions. Within network analysis, an example is Pittman and GauthierDickey [16] who investigated player distribution in the two online games World of Warcraft and Warhammer Online. While analyzing in-game spatial data, Bauckhage et al. [2] propose methods to categorize spatial player behavior which could be later used to learn the transitions between game sectors. Drachen et al. [6] used clustering methods to create behavioral gameplay profiles from a First Person Shooter (FPS) game and a MMOG. Bauckhage et al. [1] observed specific patterns in the playtime distribution across five major commercial game titles, and presented an explanation for why the Weibull distribution model provides good fits on various aspects of player behavior (playtime, session frequency, session length, inter-session time). Following this study, Sifa et al. [23] documented the same patterns for over 3000 games, and furthermore noted the presence of groups of games featuring similar aggregate playtime profiles. A major topic of interest in the game industry is player churn analysis. Having analyzed five F2P mobile

game players, Hadiji et al. [9] proposed models to predict player departure, i.e. churn, in order to allow the developers or the studios to take precautions to prevent the players from churning. Runge et al. [19] focused on two F2P games, presenting another method for churn modeling.

1.2 Contribution

This paper takes a step towards addressing the need for recommender systems specialized for games. The first contribution of the work presented two different formulations of archetypal analysis for Top- L recommender task using implicit feedback. These form the first application of recommender systems to digital games. The two models comprise: 1) a factor based model aiming to impute missing values; 2) user-based neighborhood oriented model operating in reduced dimensions. Both systems are evaluated using off-line evaluations following a similar procedure to [4]. The evaluation approach evaluates hit recall ratio that is based on the ranking assigned to a blinded game selected randomly among the games the player in question spends the most of their time on and 100 randomly selected games. The evaluation is then based on where the blinded game occurs in the ranked recommendations. Using our models, we obtain up to 86% recall when predicting mostly played games for players in the cases of Top-5 recommendation and over 97% recall in the cases of Top-30 recommendation.

2 Steam, Data and Pre-processing

Steam is an online, cross-platform game distribution system, with around 75 million active users, about 172 million accounts total, hosting over 3000 games, which makes it an ideal platform for the type of work presented here. The dataset used here was harvested by [23] from public Steam profiles, using the API client provided by Valve. The same dataset is used here to have consistent and comparable results to our previous work. The dataset contains records from over 3200 games and applications, but after running through the preprocessing steps detailed below, the dataset was constrained to 3,007 full games and 6,049,520 Steam players, covering 5,068,434,399 hours of game-play.

The public profiles and the corresponding players were selected from the most populous 3500 communities on Steam, and their IDs are anonymized by random hashing. The data was harvested in the Spring of 2014 and contains information on playtime for the games owned by the user. For some players, the dataset may not cover their full player histories (i.e. still active players), and may bias results towards showing shorter playtimes than they actually are. It is also important to note that the tracking of player behavior in the Steam platform started after March 2009⁴, which eliminates the playtime of the users before this time. A series of preprocessing steps have been performed: all game demos and Software Development Kits (SDKs) were removed, as were games not played by

⁴ <http://forums.steampowered.com/forums/showthread.php?p=10247483>

at least 25 people. Furthermore, there was a small set of games with no playtime information, i.e. games that do not save the information about whether it has been downloaded and not played. These games were also eliminated. It is also important to note that the dataset only covers playtime on the Steam platform. We tested our recommender models that predict the mostly played games using a sample of players of the game Warframe⁵, a F2P combat game (3rd most played F2P game on Steam). Warframe was randomly selected from the 25 most played games in the dataset, to ensure a sufficiently large sample size of players: 772,068 of the 6 million people in the dataset has played Warframe, leading to a combined playtime of over 22 million hours. The game is furthermore a good candidate for the work presented here because it, by the time of writing, is only about one year old, which means there is theoretically less chance of tracking corruption as compared to older games. Of the Warframe players, over 540,000 have played at least 5 games and 100,000 players were randomly sampled from this pool to evaluate our models.

3 Archetypal Analysis

In this section we briefly introduce Archetypal Analysis and its properties. Archetypal Analysis [5] is a matrix decomposition technique based on decomposing the given matrix into a collection of extreme entities, called *archetypes*, and stochastic coefficient vectors to represent each data point as a convex combination of the found archetypes. Archetypal Analysis has been extensively used in the gamemining research to cluster players [6, 24], analyze social group activities [25], group games based on gameplay-interest models [23] and to generate human-like game bots [22]. Formally given an m dimensional data matrix with n data points as $\mathbf{X} \in \mathbb{R}^{m \times n}$ and an integer k , archetypal analysis finds k archetypes $\mathbf{Z} \in \mathbb{R}^{m \times k}$ and a non-negative column stochastic coefficient vectors $\mathbf{A} \in \mathbb{R}^{k \times n}$, that satisfy

$$\|\mathbf{a}_j\|_1 = 1 \quad \forall j \in [1, 2, \dots, n]. \quad (1)$$

Having found the appropriate archetypes, the data points can be represented as convex combinations of the archetypes as

$$\mathbf{x}_j \approx \mathbf{Z}\mathbf{a}_j = \sum_{i=1}^k \mathbf{z}_i a_{ij}. \quad (2)$$

Additionally, in [5] archetypes were defined as convex combinations of the actual data points which can be represented as

$$\mathbf{z}_i = \mathbf{X}\mathbf{b}_i = \sum_{j=1}^n \mathbf{x}_j b_{ji} \quad (3)$$

⁵ <http://store.steampowered.com/app/230410/>

where $\mathbf{B} \in \mathbb{R}^{n \times k}$ are non-negative column stochastic coefficient vectors satisfying the following

$$\|\mathbf{b}_i\|_1 = 1 \quad \forall i \in [1, 2, \dots, k]. \quad (4)$$

Representing this as matrix reconstruction problem, Archetypal Analysis finds the optimal archetypes and the coefficient matrices to reduce the representation error that can be quantified by the Frobenius Norm as

$$\|\mathbf{X} - \mathbf{XBA}\|_F = \|\mathbf{X} - \mathbf{ZA}\|_F = \|\mathbf{E}\|_F = \sqrt{\sum_{u=1}^n \sum_{y=1}^m |e_{uy}|}. \quad (5)$$

Identifying the possible values of k , Cutler and Breiman [5] show that for $k = 1$ the minimizer of (5) is the data mean, whereas for values between 1 and n , i.e. $1 < k < n$, the archetypes are in the data convex hull and finally, for the case where $k = n$, having the data elements as archetypes is the global minimizer of (5). Various methods have been studied to find archetypes that reduce (5) by keeping, relaxing or strengthening the above constraints. Cutler and Breiman [5] have proposed an alternating algorithm that solves convex least squares problems to find optimal matrices \mathbf{A} and \mathbf{B} . Thureau et al. [26] increase the speed of finding the archetypes by constraining the archetypes to be lying in the data convex hull. Restricting the archetypes to be data points, Thureau et al. [27] found the archetypes that maximize the volume of the data simplex. For a detailed explanation of archetypal analysis and its applications we refer the reader to [5, 26, 27].

4 Archetypal Top-L Recommender Systems

In this section we describe how Archetypal Analysis can be used to recommend $L \in \mathbb{N}$ items to users given that we know which items they own. That is, we will describe two different data representation and recommendation approaches to increase recommendation accuracy. As the main intention of recommender engines is to provide the users useful recommendations that are personalized according to their tastes, our scenario is based on proposing the user, called *active user*, a list of games, denoted by \mathcal{L} , that they might be interested in. Formally, representing the playtime data as a game-player matrix $\mathbf{T} \in \mathbb{R}^{m \times n}$, where we have m games and n players, our aim is to come up with a list of games that a player i have not yet played and might be interested in playing, based on the games they have played, which we group under the set $\mathcal{D} = [j \mid \forall t_{ji} > 0]$. It is important to note that the solutions we provide here are inclined to fully observed datasets, where the zero values in the data matrix are not treated as missing values. This is an important aspect to distinguish when we analyze implicit playing behavior measures, such as playtime or login time distributions, number of in-app purchases, number of friends and so on. In fact, unlike the explicit movie ratings, this sensory, or telemetry, data represents the confidence of the user to the particular attribute rather than being a direct indicator of interest [1, 23, 10].

4.1 Factor Oriented Model

The first recommendation model we propose in this work is a factor based model that aims to impute the missing value by refactorization. Namely, having selected an integer k and factorized the playtime matrix \mathbf{T} using Archetypal Analysis we obtain:

$$\mathbf{T} \approx \mathbf{G}^T \mathbf{P}. \quad (6)$$

In this equation \mathbf{G}^T represents the extreme game player profiles that form the *game factor matrix* and \mathbf{P} , the *player factor matrix*, contains the stochastic coefficient vectors that are used to represent each player as a convex combination of the extreme entities in \mathbf{G}^T . Prediction at this point is done similar to the other latent factor models by multiplying the respective (active) player and game factors. Namely the estimation of association between game j and player i is found as

$$\hat{t}_{ji} = \mathbf{g}_j \mathbf{P}_i = \sum_{u=1}^k g_{uj} p_{ui}. \quad (7)$$

As a final step, as shown in (8), our main aim is now to find the top L games for the player i that have the highest estimated values of the associations.

$$\mathcal{L} = \operatorname{argmax}_{l \notin D} \hat{t}_{li} \quad (8)$$

Compared to the neighborhood oriented models, the factor model provides a very fast and a scalable way to make recommendations, as we do not have to calculate similarities between the available users, but directly multiply the corresponding two factor vectors of the active player and the active game.

4.2 Neighborhood Oriented Models

Another use of Archetypal Analysis for Top-L recommender systems is through neighborhood methods in the reduced dimensional spaces. Having found the appropriate archetypes and coefficient vectors, for each user, we can make recommendation based on neighboring items or users in the calculated coefficient space. An obvious advantage of this set of methods is the representation of the players or the games in the reduced dimensions, k -simplex, which reduces the neighborhood calculation time significantly, as $k \ll n$. Initially, we factorize the player-game matrix as in (6), where we obtain the simplex embedding of the players in the matrix \mathbf{P} .

At each recommendation step where we are required to recommend L games to the user i , we first find the most similar players to player i that maximize the similarity, or minimize the distance, in the reduced dimensional player factor space represented by \mathbf{P} . Variety of distance metrics can be used to find similar entities in the reduced dimensional space those include, cosine similarity, Pearson’s correlation index or entropy based similarity measures. Throughout our experiments we used cosine similarity measure to calculate the similarities between users as it provided us a fast and efficient solution through vectorized

operations. Cosine similarity is an angular similarity measure, that is based on finding the cosine value of the angle between two vectors. We can calculate the Cosine similarity between two vectors \mathbf{v}_1 and \mathbf{v}_2 by finding the dot product between the vectors after normalizing them as in

$$sim(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1 \mathbf{v}_2}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|}. \quad (9)$$

Having selected a similarity measure, which we denote as sim , the estimation of the association for recommending any game j can be calculated as a weighted average between the values of the closest (with respect to distance in the reduced space) U players grouped in \mathcal{U} . Formally, we find the estimation of the association between player i and game j as

$$\hat{t}_{ji} = \frac{\sum_{u \in \mathcal{U}} sim(\mathbf{p}_i, \mathbf{p}_u) t_{ju}}{\sum_{u \in \mathcal{U}} sim(\mathbf{p}_i, \mathbf{p}_u)}. \quad (10)$$

Having calculated the estimation for the association values, the top L games are recommended by returning the ones that have the highest predicted playtime association values, as done in (8).

5 Evaluation and Results

Variety of methods in the recommender systems literature have been proposed to evaluate recommender systems [4, 12, 11]. In this study we follow a user based off-line evaluation where we randomly blind one of the mostly played games for each player to form a test set and use the rest of the data set to train our models. We follow an evaluation method similar to [4], that is, our aim is to evaluate the hit recall based on the ranking assigned to the blinded (active) games and randomly selected 100 games for each test case. The addition of the randomly selected games to the evaluation process is made to observe that, given the history of the active player, if the system returns the active game to the user, i.e. if a high association value is assigned to the very game. An ideal recall hit, 100%, will be achieved if all of the blinded games in the test set are returned, or ranked, in the recommended top- L list, and oppositely, 0 % recall will be achieved if non of the games in the test set are returned in any of the recommendation steps. Therefore, for each testing instance we calculate the association between the active player and the 101 games and find out if the recommended top L games contain the blinded game to get a hit. We repeat the same procedure for different numbers of returned games, i.e. top- L games, and report the hit ratio for each L value to obtain the recall curves. We preferred to solely use recall based evaluation so as to simulate the specif game recommendation scenarios where the player is offered to purchase or download a collection of games that are presented without any particular order.

We have tested the factor oriented and the player based neighborhood oriented methods (which we named **AAF** and **AANeP** respectively) to predict

the games that are likely to be played by each Steam player for a relatively long time. Fig. 2 shows the recommendation results from our models with different parametrization where the recall values at the y-axis represents the hit ratio of the sought game and the x-axis represents the number of recommended games.

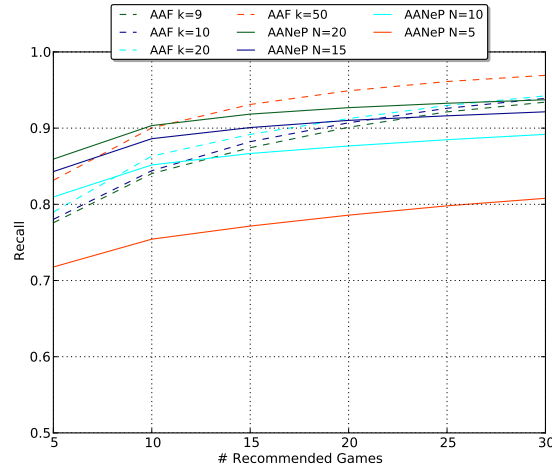


Fig. 2. Top- L Game recommendation results for different parametrization of **AAF** and **AANeP**. The recall value at the y-axis represents the hit ratio of the sought game and the x-axis represents the number of recommended games. Best seen in color.

So as to compare our models with other recommender systems, we used the following four recommendation methods:

- **Rand**: A random recommender that recommends L randomly selected (from uniform distribution) games to each user (among the games that are not played by the user)
- **PurPop**: A popularity based recommender system that recommends users unplayed games that are sorted with respect to the global frequency of the ownership (for free-to-play games) or purchases (for paid games)
- **TimePop**: A popularity based recommender system that recommends users unplayed games that are sorted with respect to the total playtime of the games
- **NN**: An item-based nearest neighbor recommender that estimates player-game associations based on player’s playtime spent on similar games.

The first model is introduced to show how a random model would behave for top- L recommender systems, similar to [4], the following two popularity based recommender models are used to show the effect of global recommendations to find the mostly played games. Finally the last method is used to provide yet another baseline classifier as it has been used commonly in the recommender

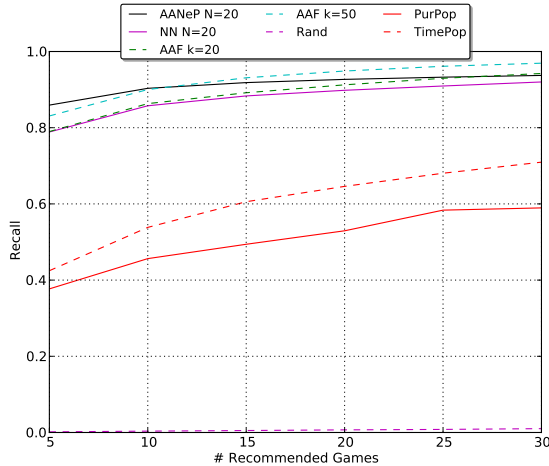


Fig. 3. Top- L game recommendation results for recommending the mostly played games in individual basis. Archetypal Analysis based recommender models performed the best among the methods used in the experiments. Best seen in color

systems literature [20, 10]. Figure 3 illustrates the recall values for all of the tested methods in a single picture where Archetypal Analysis based recommender models performed the best among the methods used in the experiments. We can see that random recommender systems perform poorly when used for top- L recommendation tasks. Additionally, unlike the above random recommender, the results of the popularity based recommender systems show that there are trends for playing games in the steam platform, which indicates, recommending the most popular games increases the recall rates over 50%. Analyzing the results of Archetypal recommender systems we can observe that user based neighborhood oriented method and the factor methods give the best prediction performances among the tested methods reaching up to 86% recall for Top-5 recommendations (for AANeP) and over 97% recall for Top-30 recommendations (for AAF).

6 Discussion and Conclusion

With the steadily increasing number of available digital games, recommender systems have started to gain the interest of the game industry as the issues surrounding information search and selection are becoming increasingly serious. This not only from the viewpoint of the users, who need to find the right games, but also from the perspective of the game developers, who due to high and increasing user acquisition costs, especially for F2P games [21], have a direct need to optimize strategies to find the right users for their games.

In this paper we propose two different recommender methods for Top- L recommender tasks, that are based on archetypal analysis [5] and use implicit feedback via data on playtime and game ownership. It is to the best knowledge of

the authors the first application of recommender systems to digital games. The two models are evaluated against a dataset of 500,000 users of the Steam digital game distribution platform, covering more than 3000 games. For L values of 5, recall rates of 86% are reached for AANeP and 84% for AAF, which outperforms the other four recommender models we tested for evaluation. This shows that archetypal analysis can be used for Top- L recommendation purposes. While multiple values of L are tested and results presented in Fig. 3, the L value of 5 is highlighted here as it forms the lower boundary of the classical 7 ± 2 rule from user interface design [15]. The rule basically states that human short term memory can normally handle 7 ± 2 chunks of information. This means that chunking information can increase the short-term memory capacity, but it also means that showing a user for example 10 images of recommended games (for example using box art), will not allow the user to identify these with a glance. On the contrary, if short-term identification is a goal, a maximum of 5 recommended games at a time would appear to be a safer option. This in turn means that there may be an interest in optimizing recommender systems for games towards L values that fall within the 7 ± 2 rule.

The goal here was to use playtime information across all games played by the users on the Steam platform, to predict a game that the player would play for a long time. This is just one potential goal of game recommender systems. Other goals, that will be investigated in future work, include finding players who will spend a lot of money on in-game purchases or have a high social value. Our future work on game recommender systems will also focus on increasing the size of the sample used here. Additionally, features such as social networking impacts will be investigated. Finally, a practical requirement of game recommender systems when it comes to their adoption by the game industry will be the speed at which they can be executed for large sample sizes. While the models used here are relatively fast to compute, optimization forms another venue for future research.

Acknowledgment

The work reported in this paper was carried out within the Fraunhofer / University of Southampton research project *SoFWIReD*. The authors gratefully acknowledge this support.

References

1. Bauckhage, C., Kersting, K., Sifa, R., Thureau, C., Drachen, A., Canossa, A.: How Players Lose Interest in Playing a Game: An Empirical Study Based on Distributions of Total Playing Times. In: Proc. IEEE CIG (2012)
2. Bauckhage, C., Sifa, R., Drachen, A., Thureau, C., Hadji, F.: Beyond Heatmaps: Spatio-Temporal Clustering using Behavior-Based Partitioning of Game Levels. In: Proc. IEEE CIG (2014)
3. Chambers, C., F.W.S.S., Saha, D.: Measurement-based Characterization of a Collection of On-line Games. In: Proc. of ACM SIGCOMM Conf. on Internet Measurement (2005)

4. Cremonesi, P., Koren, Y., Turrin, R.: Performance of Recommender Algorithms on Top-N Recommendation Tasks. In: Proc. ACM Recsys (2010)
5. Cutler, A., Breiman, L.: Archetypal Analysis. *Technometrics* 36(4), 338–347 (1994)
6. Drachen, A., Sifa, R., Bauckhage, C., Thureau, C.: Guns, swords and data: Clustering of player behavior in computer games in the wild. In: Proc. IEEE CIG (2012)
7. Entertainment Software Association: Essential Facts About the Computer and Video Game Industry (2014), http://www.theesa.com/facts/pdfs/esa_ef_2014.pdf
8. Gartner Group: Gartner Says Worldwide Video Game Market to Total \$93 Billion in 2013 (2013), <http://www.gartner.com/newsroom/id/2614915>
9. Hadji, F., Sifa, R., Drachen, A., Thureau, C., Kersting, K., Bauckhage, C.: Predicting Player Churn in the Wild. In: Proc. IEEE CIG (2014)
10. Hu, Y., Koren, Y., Volinsky, C.: Collaborative Filtering for Implicit Feedback Datasets. In: Proc. IEEE Int. Conf. Data Mining, ICDM. IEEE (2008)
11. Kantor, P., Rokach, L., Ricci, F., Shapira, B.: Recommender Systems Handbook. Springer (2011)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* 42(8), 30–37 (2009)
13. Laratta, D.: The Death of the Video Game Expert (2010), <http://venturebeat.com/2010/04/08/the-death-of-the-video-game-expert/>
14. Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Computing* 7(1), 76–80 (2003)
15. Miller, G.A.: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity to Process Information. *Psychological Review* 63(2), 81–97 (1956)
16. Pittman, D., GauthierDickey, C.: Characterizing Virtual Populations in Massively Multiplayer Oline Role-playing Games. In: Proc. of Int. Conf. on Advances in Multimedia Modeling (2010)
17. PocketGamer.biz: App Store Metrics (2012), <http://www.pocketgamer.biz/metrics/app-store/categories/>
18. Resnick, P., Varian, H.: Recommender Systems. *Communications of the ACM* 40(3), 56–58 (1997)
19. Runge, J., Gao, P., Garcin, F., Faltings, B.: Churn Prediction for High-value Players in Casual Social Games. In: Proc. IEEE CIG (2014)
20. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms. In: Proc. ACM WWW (2001)
21. Seif El-Nasr, M., Drachen, A., Canossa, A.: Game Analytics: Maximizing the Value of Player Data. Springer (2013)
22. Sifa, R., Bauckhage, C.: Archetypal Motion: Supervised Behavior Learning Using Archetypal Analysis. In: Proc. IEEE CIG (2013)
23. Sifa, R., Bauckhage, C., Drachen, A.: The Playtime Principle: Large-scale Cross-games Interest Modeling. In: Proc. IEEE CIG (2014)
24. Sifa, R., Drachen, A., Bauckhage, C., Thureau, C., Canossa, A.: Behavior Evolution in Tomb Raider Underworld. In: Proc. IEEE CIG (2013)
25. Thureau, C., Bauckhage, C.: Analyzing the Evolution of Social Groups in World of Warcraft. In: Proc. IEEE CIG (2010)
26. Thureau, C., Kersting, K., Bauckhage, C.: Convex Non-negative Matrix Factorization in the Wild. In: Proc. ICDM (2009)
27. Thureau, C., Kersting, K., Bauckhage, C.: Yes We Can: Simplex Volume Maximization for Descriptive Web-Scale Matrix Factorization. In: Proc. ACM CIKM (2010)

Preference Learning from Annotated Game Databases

Christian Wirth and Johannes Fürnkranz

Knowledge Engineering, Technische Universität Darmstadt, Germany
{cwirth, fuernkranz}@ke.tu-darmstadt.de

Abstract. In chess, as well as many other domains, expert feedback is amply available in the form of annotated games. This feedback usually comes in the form of qualitative information because human annotators find it hard to determine precise utility values for game states. Therefore, it is more reasonable to use those annotations for a preference based learning setup, where it is not required to determine values for the qualitative symbols. We show how game annotations can be used for learning a utility function by translating them to preferences.

We evaluate the resulting function by creating multiple heuristics based upon different sized subsets of the training data and compare them in a tournament scenario. The results show that learning from game annotations is possible, but our learned functions did not quite reach the performance of the original, manually tuned function. The reason for this failure seems to lie in the fact that human annotators only annotate “interesting” positions, so that it is hard to learn basic information, such as material advantage from game annotations alone.

1 Introduction

For many problems, human experts are able to demonstrate good judgment about the quality of certain courses of actions or solution attempts. Typically, this information is of qualitative nature, and cannot be expressed numerically without selecting arbitrary values. Particularly well-studied form of qualitative knowledge are so-called *pairwise comparisons* or *preferences*. Humans are often not able to determine a precise utility value of an option, but are typically able to compare the quality of two options (e.g., “Treatment a is more effective than treatment b ”). Thurstone’s *Law of Comparative Judgment* essentially states that such pairwise comparisons correspond to an internal, unknown utility scale [14]. Recovering this hidden information from such qualitative preference is studied in various areas such as ranking theory [12] or voting theory [3] Most recently, the emerging field of preference learning [6] studies how such qualitative information can be used in a wide variety of machine learning problems.

Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

In the game of chess, qualitative human feedback is amply available in the form of game notations.

We show how this information can be used in combination with state-of-the-art ranking algorithms to successfully learn an evaluation function. The learning setup is based on the methodology used in Paulsen and Fürnkranz [13], where it has been used for learning evaluation functions from move preferences of chess players of different strengths. This paper briefly summarizes the key results, for details we refer to Wirth and Fürnkranz [18].

In Section 2, we discuss the information that can typically be found in annotated chess games. Section 3 shows how to extract preference information from such data and how to use this information for learning an evaluation function via an object ranking algorithm. In our experimental setup (Section 4), we evaluate the proposed approach with an large scale tournament, discussing the results in Section 5. Section 6 concludes the paper and discusses open questions and possible future work.

2 Game Annotations in Chess

Chess is a game of great interest, which has generated a large amount of literature that analyzes the game. Particularly popular are game annotations, which are regularly published after important or interesting games have been played in tournaments. These annotations reflect the analysis of a particular game by a (typically) strong professional chess player.

Annotated chess games are amply available. For example, the largest database distributed by the company *Chessbase*¹, contains over five million games, more than 66,000 of which are annotated.

Chess players annotate games with alternative lines of play and/or textual descriptions of the evaluation of certain lines or positions. An open format for storing chess games and annotations is the so-called *portable game notation* (PGN) [4].

Most importantly, however, typical events in a chess game can be encoded with a standardized set of symbols. There is a great variety of these *numeric annotation glyphs* (NAG), referring to properties of the positions (e.g., attacking chances, pawn structures, etc.), the moves (e.g., forced moves, better alternatives, etc.), or to external properties of the game (such as time constraints). In this work, we will focus on the most commonly used symbols, which annotate the quality of moves and positions:

- *move evaluation*: Each move can be annotated with a symbol indicating its quality. Six symbols are commonly used:
 - very poor move (??),
 - poor move (?),
 - speculative move (?!),
 - interesting move (!?),

¹ <http://www.chessbase.com/>

- good move (!),
- very good move (!!).
- *position evaluation*: Each move can be annotated with a symbol indicating the quality of the position it is leading to:
 - white has a decisive advantage (+-),
 - white has a moderate advantage (\pm),
 - white has a slight advantage (\pm),
 - equal chances for both sides (=),
 - black has a slight advantage ($\bar{\mp}$),
 - black has a moderate advantage ($\bar{\mp}$),
 - black has a decisive advantage (-+),
 - the evaluation is unclear (∞).

We will denote the set annotation symbols with $\mathcal{A} = \mathcal{A}_P \cup \mathcal{A}_M$ where \mathcal{A}_P are position annotations and \mathcal{A}_M are move annotations. $\mathcal{A}(\mathbf{m})$ are the annotations associated with a given move \mathbf{m} in the annotated game. Move and position evaluations can be organized into a partial order which we denote with the symbol \sqsupset . The move evaluations can be ordered as

$$+- \sqsupset \pm \sqsupset \pm \sqsupset = \sqsupset \bar{\mp} \sqsupset \bar{\mp} \sqsupset -+$$

and the position evaluations as

$$!! \sqsupset ! \sqsupset !? \sqsupset ?! \sqsupset ? \sqsupset ??.$$

Note that, even though there is a certain correlation between position and move annotations (good moves tend to lead to better positions and bad moves tend to lead to worse positions), they are not interchangeable. A very good move may be the only move that saves the player from imminent doom, but must not necessarily lead to a very good position. Conversely, a bad move may be a move that misses a chance to mate the opponent right away, but the resulting position may still be good for the player. For this reason, \sqsupset is partial in the sense that it is only defined on $\mathcal{A}_M \times \mathcal{A}_M$ and $\mathcal{A}_P \times \mathcal{A}_P$, but not on $\mathcal{A}_M \times \mathcal{A}_P$.

In addition to annotating games with NAG symbols, annotators can also add textual comments and variations. These complement the moves that were actually played in the game with alternative lines of play that could have happened or that illustrate the assessment of the annotator. Typically, such variations are short move sequences that lead to more promising states than the moves played in the actual game. Variations can also have NAG symbols, and may contain subvariations.

It is important to note that this feedback is of qualitative nature, i.e., it is not clear what the expected reward is in terms of, e.g., percentage of won games from a position with evaluation \pm . However, according to the above-mentioned relation \sqsupset , it is clear that positions with evaluation \pm are preferable to positions with evaluation \pm or worse ($=, \bar{\mp}, \bar{\mp}, -+$).

As we will see in the following, we will collect preference statements over positions. For this, we have to uniquely identify chess positions. Chess positions

can be efficiently represented in the *Forsyth-Edwards Notation* (FEN), which is a serialized, textual representation of the game board, capturing all data that is required to uniquely identify a chess state.

3 Learning an Evaluation Function from Annotated Games

Our approach of learning from qualitative game annotations is based on the idea of transforming the notation symbols into preference statements between pairs of positions, which we describe in more detail in sections 3.2 Each such preference may then be viewed as a constraint on a utility function for chess positions, which can be learned with state-of-the-art machine learning algorithms such as support-vector machines (Section 3.4). First, however, let us briefly recapitulate a few key notions in the field of preference learning.

3.1 Preference Learning

Preference learning [6] is a newly emerging area that is concerned with the induction of predictive preference models from empirical data. It establishes a connection between machine learning and research fields like preference modeling or decision making.

One can distinguish two main types of preference learning problems: in *label ranking*, the task is to learn which of a set of labeled options are preferred in a given context [9, 15]. *Object ranking* is an alternative setting, where the preferences are defined over a set of objects, which are not designated by labels but characterized via features, and the task is to learn a ranking function for arbitrary objects from this domain [11]. More precisely, the task is to learn a utility function $f(\cdot)$ that allows to order a subset of objects out of a (potentially infinite) reference set \mathcal{Z} . The training data is given in the form of rankings between subsets of these objects $\mathbf{z} \in \mathcal{Z}$, which are usually specified as a vector of values for a given set of features. These rankings over objects can be decomposed into a finite set of pairwise preferences $\mathbf{z}_i \succ \mathbf{z}_j$ specifying that object \mathbf{z}_i should have a higher utility than object \mathbf{z}_j , i.e., that

$$\mathbf{z}_i \succ \mathbf{z}_j \Leftrightarrow f(\mathbf{z}_i) > f(\mathbf{z}_j). \quad (1)$$

The task of the object ranking algorithm is to learn the utility function $f(\cdot)$ which can then be used to rank any given subset of objects in \mathcal{Z} .

3.2 Generating Preferences from Annotations

The training data that are needed for an object ranking algorithm can be generated from game annotations of the type discussed in Section 2.

First, we parse each game in the database \mathcal{G} , replaying all moves so that we are able to generate all occurring positions. For each move \mathbf{m} that has a set

Algorithm 1 Preference generation from position annotations

Require: database of games \mathcal{G} , initial position \mathbf{p}_0

```
1:  $prefs \leftarrow \emptyset$ ,
2: for all  $\mathbf{g} \in \mathcal{G}$  do
3:    $pairs \leftarrow \emptyset$ ,
4:    $seen \leftarrow \emptyset$ 
5:    $\mathbf{p} \leftarrow \mathbf{p}_0$ 
6:   for all  $\mathbf{m} \in \mathbf{g}$  do
7:      $\mathbf{p} \leftarrow \text{MOVE}(\mathbf{p}, \mathbf{m})$ 
8:     if  $\exists \mathbf{a} \in \mathcal{A}_P(\mathbf{m})$  then
9:        $pairs \leftarrow pairs \cup \{(\mathbf{p}, \mathbf{a})\}$ 
10:       $seen \leftarrow seen \cup \{\mathbf{p}\}$ 
11:     end if
12:   end for
13:   for all  $\mathbf{p}' \in seen$  do
14:     for all  $\mathbf{p}'' \in seen$  do
15:       for all  $(\mathbf{p}', \mathbf{a}'), (\mathbf{p}'', \mathbf{a}'') \in pairs$  do
16:         if  $\mathbf{a}' \sqsupset \mathbf{a}''$  then
17:            $prefs \leftarrow prefs \cup \{\mathbf{p}' \succ \mathbf{p}''\}$ 
18:         end if
19:       end for
20:     end for
21:   end for
22: end for
```

of associated position annotation symbols $\mathcal{A}_P(\mathbf{m})$ we associate the annotation symbols with the position \mathbf{p} that results after playing each move.

Algorithm 1 shows the algorithm for generating state preferences. The first loop collects for each game a list of all positions that have a position annotation. In a second loop, all encountered annotated positions are compared. For all pairs of positions $(\mathbf{p}', \mathbf{p}'')$ where \mathbf{p}' received a higher evaluation than \mathbf{p}'' , a corresponding position preference is generated. Position \mathbf{p}' receives a higher evaluation than \mathbf{p}'' if the associated position evaluation \mathbf{a}' prefers white to a stronger degree than the annotation \mathbf{a}'' associated with \mathbf{p}'' . (Position evaluations are always viewed from the white player)

As can be seen, we only compare position preferences within the same game and not across games. This has several reasons. One is, of course, that a comparison of all annotated position pairs in the entire database would be computationally infeasible. However, it is also not clear whether this would be a sensible thing to do.

For example, it is possible that different annotators reach different conclusions for the same position. A decisive advantage $(-+, +-)$ for one player may only be annotated as moderate advantage (\mp, \pm) by another annotator. For this reason, we only compare annotations within the same annotated game.

Concerning move preferences, we follow a comparable procedure. As already mentioned, the quality of a move depends on the possibilities given in a certain state, hence move preferences are restricted to moves applied to the same state. This means when ever we encounter two moves for the same state, annotated with different symbols, we create a preference according to the ranking given by \mathcal{A}_M . By applying the move to the state, we can convert the move preferences to position preferences. This means the preference is then given by a relation between to states directly following a given state by playing different moves.

3.3 Position Evaluations in Computer Chess

In many cases, not only a NAG annotation is available, but also a suggested move sequence that should be followed afterwards. This usually means that not the position reached after the move is preferable, but the position at the end of the line. Consequently, in such cases we also considered following these variations until the end of the line.

It should also be noted that chess engines are usually only evaluating quiet positions, where no capturing move is obviously preferred. For example, when the opponent initiated a piece-trading move sequence, the trade should be completed before evaluating the position. This is why such a setup has been suggested in previous works on reinforcement learning in chess [2]. Hence, we further extended the algorithm so that it may also perform a quiescence search.

3.4 SVM-based ranking

Once we have a set of position preferences of the form $\mathbf{p}_i \succ \mathbf{p}_j$, we can use them for training a ranking support vector machine, as proposed in [10]. As stated above (1), the key idea is to reinterpret the preference statements as constraints on an evaluation function. If this function f is linear, i.e., it is a weighted sum

$$f(\mathbf{p}) = \sum_{\phi} w_{\phi} \cdot \phi(\mathbf{p}) \quad (2)$$

of features ϕ , the latter part is equivalent to

$$\begin{aligned} f(\mathbf{p}_i - \mathbf{p}_j) &= \sum_{\phi} w_{\phi} \cdot \phi(\mathbf{p}_i - \mathbf{p}_j) \\ &= \sum_{\phi} w_{\phi} \cdot (\phi(\mathbf{p}_i) - \phi(\mathbf{p}_j)) > 0 \end{aligned} \quad (3)$$

This essentially corresponds to the training of a classifier on the pairwise differences $\bar{\mathbf{p}}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ between positions \mathbf{p}_i and \mathbf{p}_j with an unbiased hyperplane. The pairwise ranking information can thus be converted to binary training data in the form of feature distance vectors $\bar{\mathbf{p}}_{ij}$.

We selected SVMs for our experiments because they learn a linear evaluation function, which can be easily plugged into the chess program, and which can

Table 1. Features used in the linear evaluation function of the CUCKOO chess engine.

| Feature Type | # Features | Description |
|----------------------------|------------|---|
| <i>material difference</i> | 1 | Difference in the sum of all piece values per player. |
| <i>piece square</i> | 6 | Position dependent piece values by static piece/square tables. |
| <i>pawn bonus</i> | 1 | Bonus for pawns that have passed the enemy pawn line. |
| <i>trade bonus</i> | 2 | Bonus for following the “when ahead trade pieces, when behind trade pawns” rules. |
| <i>castle bonus</i> | 1 | Evaluates the castling possibility. |
| <i>rook bonus</i> | 1 | Bonus for rooks on (half-) open files. |
| <i>bishops scores</i> | 2 | Evaluating the bishops position by attack possibilities, if trapped and relative positioning. |
| <i>threat bonus</i> | 1 | Difference in the sum of all piece values under attack. |
| <i>king safety</i> | 1 | Evaluates the kings position relative to the rooks. |

be quickly evaluated. This is important for a good performance of the chess program because the more positions it can evaluate the deeper it can search. Finally, the default evaluation function of the chess program is also linear, so that it should, in principle, be possible to achieve a good performance with learned linear functions.

3.5 Related Work

The basic idea of learning from annotated games was first introduced by [8], where a hill-climbing approach was used to optimize Kendall’s τ rank correlation measure.

Our approach to using a preference-based object ranking approach for learning an evaluation function essentially follows [13, 16], where a similar algorithm was applied to the problem of learning evaluation functions of different playing strengths. In that case, the training preferences were obtained by assuming that the move actually played by a player had been preferred over all other moves that had not been played.

Of course, approaches to learning from (expert) demonstration, most notably inverse reinforcement learning [1], are also applicable to learning from chess databases, but our focus is on learning from annotations. IRL assumes (near) perfect demonstrations, which is hardly available in many domains, as opposed to the partial, qualitative evaluations that can be found in the game annotations.

4 Experimental Setup

To investigate the usefulness of preference data, we train chess evaluation functions based on preferences generated from annotated chess games (Section 4.1), and employ them in the strong open source chess engine CUCKOO². All states are

² <http://web.comhem.se/petero2home/javachess/>

represented by the heuristic features created by the position evaluation function. CUCKOO’s evaluation function uses the 16 features, shown in Table 1.

Training a linear kernel model allows us to simply extract the feature weights for the linear sum function. The quality of the preferences can then be analyzed by comparing the playing strength of our re-weighted chess engine. In this section, we describe the basic experimental setup, the results of the experiments will be describe in the subsequent section 5.

4.1 Chess Database

As a data source we used *Mega Database 2012*, which is available from *Chessbase*. To the authors’ knowledge, this is the largest database of professionally annotated chess games available. The annotations are commonly, but not exclusively, provided by chess grandmasters.

In the more than 5 million games contained in the database, we found 86,767 annotated games,³ from which we computed 5.05 million position preferences and 190,000 move preferences using the algorithms sketched in section 3.

4.2 Evaluation

The learned evaluation functions have been evaluated in several round robin tournaments, where each learned function plays multiple games against all other functions. All tournaments are played using a time control of two seconds per move.

All results are reported in terms of Elo ratings [5], which is the commonly used rating system for rating chess players. It also allows reporting upper and lower bounds for the playing strength of each player. For calculating the Elo values, a base value of 2600 was used, because this is the rating for CUCKOO as reported by the *Computer Chess Rating List*⁴. It should be noted that computer engine Elo ratings are not directly comparable to human Elo ratings, because they are typically estimated on independent player pools, and thus only reflect relative strengths within each pool.

5 Results

In this section, we report the results from the explained approach. Upon analyzing these results, we noticed that a possible reason for the weak performance could be that annotated positions are not representative for all positions that may be encountered during this game. We investigate this hypothesis in Section 5.2, where we show how the results can be improved by adding positions where one side has a very clear, decisive advantage.

³ In addition to the 66k games with textual annotations, there are a few more that only contain a few annotation symbols, which we included, but which are not officially counted as “annotated”.

⁴ <http://www.computerchess.org.uk/ccr1/>

5.1 Learning from Preferences

We generated both move and position preferences for training set sizes of 10%, 25%, 50%, and 100% of all games. We divided the data into k non-overlapping folds, each holding $\frac{1}{k}$ -th of the data (e.g., 4 folds with 25%), and trained an evaluation function for each fold. The players that represent different values of k then, before each game, randomly selected one of the evaluation functions for their size. The second and third column of Table 2 show the number of preferences in the training data for each configuration, averaged over all folds of the same size (we will return to columns four and five in Section 5.2).

We first optimized various parametrizations of the experimental setup. In particular, we optimized the C -parameter of the SVM by testing three values: $C = 1$, $C = 0.01$ and $C = .00001$.

Our second objective was to compare different ways of incorporating position evaluations. As described in Section 3.3, one can either directly use the position to which the annotation is attached, attach the annotation to the end of the line provided by the annotator, or perform a quiescence search as is commonly done in reinforcement learning of evaluation functions.

It turns out that following the annotated side lines, but without using quiescence search performs best in every instance. Hence, we decided to run all subsequent experiments with this setup.

These optimal parameter settings were then used in a large tournament that compares the performance of different types of preference information. Figure 1 shows the results. Note that the absolute Elo values presented there are not comparable with the baseline value, because they are estimated on a different player pool. This means the new evaluation function does not outperform the original one. For the moment, only consider the right-most three groups in each graph, namely those representing learning from position preferences, learning from move preferences and learning from both. Clearly, learning from only move preferences yields much worse results than learning from position preferences. Moreover, adding move preferences to the position preferences does not further increase the performance.

Clearly, one has to keep in mind that the number of move preferences is considerably smaller than the number of position preferences, so that the values are already different with respect to training set sizes. However, we also see from the

Table 2. Average Number of Generated Preferences (in Millions).

| #games | Preferences | | | |
|--------|-------------|----------|-----------|-----------|
| | Move | Position | Augmented | Surrender |
| 10% | 0.02M | 0.50M | 0.34M | 0.05M |
| 25% | 0.04M | 1.26M | 0.85M | 0.12M |
| 50% | 0.09M | 2.53M | 1.71M | 0.24M |
| 100% | 0.19M | 5.05M | 3.41M | 0.48M |

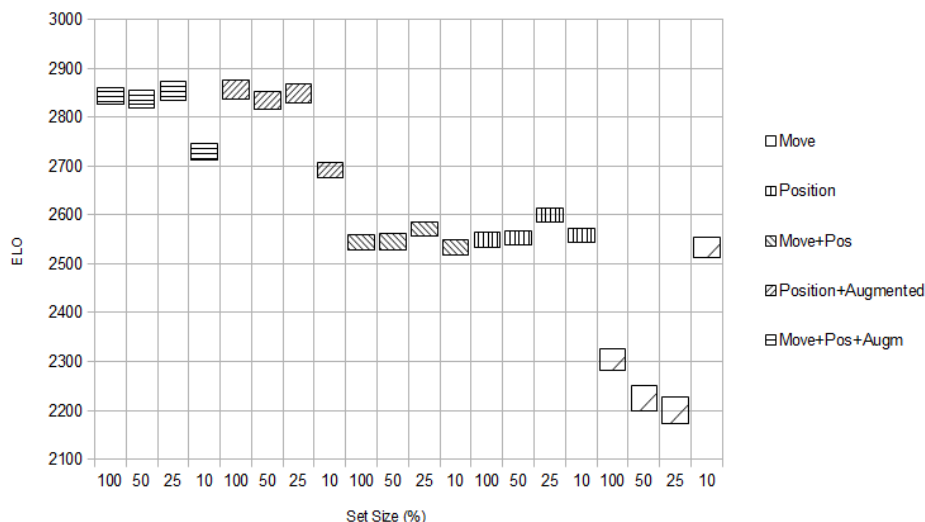


Fig. 1. Comparison of players learned from different types of preference information for four training set sizes with the square height as confidence interval.

results that learning from position preferences is already saturated in the sense that there are only minor performance variations between the different training set sizes, i.e., more position preferences do not further improve performance. The same seems to be the case when adding move preferences to the position preferences. Thus, we are inclined to conclude that the move preferences do not add qualitatively different information but just add more of the same information.

5.2 Augmenting Position Preferences

After a first batch of experiments, we noticed a considerable performance gap between the performance of the original CUCKOO chess program and the one using our learned evaluation function. In particular, we noticed that fundamental concepts like the material value of the different pieces tends to be underestimated by the learned heuristics, resulting in a lower playing strength.

A possible reason for this could be that human players tend to only annotate “interesting” positions, i.e., positions where the evaluation is not entirely clear. Thus, the program may miss picking up that, e.g., it is in general really bad to be a queen down because such positions are too clear to be annotated.

In order to test this hypothesis, we tried to augment our preference data with preferences that involve a very clear material advantage for one side. For each position annotated as a decisive advantage (+-, -+, we generated a new position by removing a random piece, excluding pawns and kings, of the inferior side. The idea is that if we remove a piece from an already lost position, the resulting position should be extremely bad. Thus, such positions are hopefully

more extreme than those that are usually annotated in game databases. This technique increased the amount of state preferences by 60%-70%, as can be seen from the fourth column of Table 2.

We compared the augmented preference sets to non-augmented sets in a tournament with 100 games per pairing. Figure 1 shows the results. We can clearly see that the augmented preference sets always outperform all other settings. We can also observe a similar saturation as with position preferences, but here it only occurs after 25% of the preferences have been seen. The learned function is still not able to reach the performance of the original one.

Finally, we also tried to enhance the amount of preferences even further by including position where a player resigned the game, because most chess games do not end with checkmate. We inserted such final position at the top or bottom of the preference order \sqsubset , with the reasoning that these positions are so hopeless that their qualitative evaluation should be worse than $-+$ (or better than $+-$ respectively).⁵

Including such surrenders increased the amount of available position preferences by ca. 5% (cf. column five in Table 2). However, these additions did not further increase the performance.

6 Conclusion

We have been able to confirm, that useful information can be learned from annotations but that the result does not reach the performance of a hand-tuned function. We also showed that this is not a problem of insufficient training data.

One problem that we identified is that human annotators only tend to focus on close and interesting positions. We could show that an augmentation of the training information with artificial preferences that result from generating bad positions by removing pieces from already losing positions leads to a significant increase in the quality of the learned function, confirmed by an evaluation of the resulting feature weights. Thus, we may conclude that important training information is missing from human annotations. One could also speculate that the observed performance differences might be due to different scales used by different annotators. However, we did not generate preferences across different games, only within individual games. Thus, we cannot suffer from this problem as long as each expert is consistent with herself.

We conclude that annotations should be complemented with other phases of learning (e.g., phases of traditional, reinforcement-based self-play).

This is also what we plan to explore in the near future. We are currently thinking of ways for combining learning from game annotations with preference-based reinforcement learning algorithms that are currently under investigation [7, 17]. In that way, the information learned from game annotations can be fine-tuned with the agent's own experience (or vice versa), a strategy that seems typical for human players.

⁵ We used a few crude heuristics for excluding cases where a game was decided by time controls or similar.

References

- [1] Abbeel, P., Ng, A.Y.: Inverse reinforcement learning. In: Encyclopedia of Machine Learning, pp. 554–558. Springer-Verlag (2010)
- [2] Baxter, J., Tridgell, A., Weaver, L.: Knightcap: A chess program that learns by combining td(λ) with game-tree search. In: Proceedings of the 15th International Conference on Machine Learning, (ICML-98). pp. 28–36. Morgan Kaufmann (1998)
- [3] Coughlin, P.J.: Probabilistic Voting Theory. Cambridge University Press (2008)
- [4] Edwards, S.J.: Portable game notation (1994), <http://www6.chessclub.com/help/PGN-spec>, accessed on 14.06.2012
- [5] Elo, A.E.: The Rating of Chessplayers, Past and Present. Arco, 2nd edn. (1978)
- [6] Fürnkranz, J., Hüllermeier, E. (eds.): Preference Learning. Springer-Verlag (2010)
- [7] Fürnkranz, J., Hüllermeier, E., Cheng, W., Park, S.H.: Preference-based reinforcement learning: A formal framework and a policy iteration algorithm. Machine Learning 89, 123–156 (2012), Special Issue of Selected Papers from ECML PKDD 2011
- [8] Gomboc, D., Marsland, T.A., Buro, M.: Evaluation function tuning via ordinal correlation. In: Advances in Computer Games, IFIP, vol. 135, pp. 1–18. Springer US (2004)
- [9] Hüllermeier, E., Fürnkranz, J., Cheng, W., Brinker, K.: Label ranking by learning pairwise preferences. Artificial Intelligence 172, 1897–1916 (2008)
- [10] Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02). pp. 133–142. ACM Press (2002)
- [11] Kamishima, T., Kazawa, H., Akaho, S.: A survey and empirical comparison of object ranking methods. In: [6], pp. 181–201
- [12] Marden, J.I.: Analyzing and Modeling Rank Data. CRC Press (1995)
- [13] Paulsen, P., Fürnkranz, J.: A moderately successful attempt to train chess evaluation functions of different strengths. In: Proceedings of the ICML-10 Workshop on Machine Learning and Games (2010)
- [14] Thurstone, L.L.: A law of comparative judgement. Psychological Review 34, 278–286 (1927)
- [15] Vembu, S., Gärtner, T.: Label ranking algorithms: A survey. In: [6], pp. 45–64
- [16] Wirth, C., Fürnkranz, J.: First steps towards learning from game annotations. In: Workshop Proceedings - Preference Learning: Problems and Applications in AI at ECAI 2012. pp. 53–58 (2012)
- [17] Wirth, C., Fürnkranz, J.: EPMC: Every visit preference Monte Carlo for reinforcement learning. In: Proceedings of the 5th Asian Conference on Machine Learning, (ACML-13). JMLR Proceedings, vol. 29, pp. 483–497. JMLR.org (2013)
- [18] Wirth, C., Fürnkranz, J.: On learning from game annotations. IEEE Transactions on Computational Intelligence and AI in Games (2014), in press

Deriving the Employee-perceived Application Quality in Enterprise IT Infrastructures using Information from Ticketing Systems*

Susanna Schwarzmann, Thomas Zinner, and Matthias Hirth

University of Würzburg, Institute of Computer Science, Würzburg, Germany

The need for a less complex maintenance of applications and the IT infrastructure for huge enterprises lead to the centralization of applications and services within data centers. Employees at sites and branches are connect to data centers via the Internet using a thin-client architectures resulting in additional failure sources beside the end devices, namely the transport network and hardware components in the data centers. To provide a good application quality to the employers using a multitude of different applications and access networks has thus become a complex task [2].

In order to evaluate the quality of an application, subjective metrics like Quality of Experience (QoE) [1] are often used. Ongoing research in the field of QoE typically tries to understand the impact of technical systems on the subjective perception of specific applications. Main influence factors are deduced and appropriate models allowing an estimation of the QoE for varying parameters like bandwidth, packet loss, or jitter are developed. The QoE for applications like web browsing, video streaming, VoIP, and office products are well understood. This, however, does not hold for enterprise applications like resource planning and management or data warehouse applications, which are not covered by current research. Time-consuming user surveys in the employers working environment highly affect the day-to-day business and thus are not practicable. Nevertheless, a profound knowledge of the application quality and availability is required to enable good conditions of work and a high working efficiency. For that, enterprises may rely on support systems like a hotline or a ticketing system. Particular the latter is a huge database collecting complaints and problems of the users over a long period of time and thus are an interesting starting point to identify performance problems.

Using this data source, we propose an approach to automatically identify tickets indicating problematic applications and reflecting the user experience. To this end, our approach first groups similar tickets and afterwards tags the resulting groups with adequate keywords. For the grouping process, we rely on the information from the free-text fields of the tickets, which include a summary and a detailed description of the reported issue, and calculate the lexicographical distance between the tickets using the Jaccard index [3]. The keywords for the

* *Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes.* In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

groups are based on word frequencies within the groups. The tagged groups can finally be evaluated further to identify issues in the IT system.

We evaluate the accuracy of our approach using 12,000 tickets accumulated in June 2013 at the ticketing system of a company. These tickets were manually categorized in tickets covering application performance issues (303 performance tickets), respectively tickets addressing other issues and serve as gold standard data for the categorization results. The performance of the approach is measured by two different metrics, (1) the overall share of correctly classified tickets and (2) the share of performance relevant tickets in groups tagged as performance relevant. A parameter study was conducted to investigate the impact of different Jaccard similarity thresholds on the misclassification rate (Type I and Type II errors). Based on the specific threshold, the ratio of correctly classified performance tickets varied between 44 % and 57 %, whereas, the number of false classified non-performance tickets was between 55 and 325. Hence, a higher hit ratio also results in more manual overhead for checking the tickets within the performance ticket groups and removing the wrongly classified tickets.

Even though not all performance tickets can be detected by our algorithm, the number of correctly classified tickets is sufficient to draw conclusions about temporal performance problems. To this end, we compared the number of actual performance tickets per day with the correctly classified number of performance tickets per day identified using our approach. Both time series show similar trends, although the identified number of performance tickets per day is always lower than the actual number. Nevertheless, daily trends are preserved and presented approach detected at minimum 20% of the daily performance tickets. Consequently, the algorithm cannot be used if the exact number of tickets is required, however, it is possible to identify trends temporal occurring performance problems. The root course of the issue can then be evaluated further by technical staff using the tickets in the performance ticket groups.

The preliminary results of the proposed algorithm are promising, but a lot of optimization potential remains. In the next steps, the impact of other similarity metrics, e.g., the Cosine-measure, will be evaluated. Further, more sophisticated methods for evaluating ticket similarity will be integrated, e.g., considering n-grams or content based evaluations.

Acknowledgment This work is supported by the Deutsche Forschungsgemeinschaft (DFG) under Grants HO TR 257/41-1 “Trade-offs between QoE and Energy Efficiency in Data Centers”. The authors alone are responsible for the content.

References

1. Qualinet White Paper on Definitions of Quality of Experience (2012) Version 1.2. March 2013.
2. T. Hofffeld et al. Challenges of QoE Management for Cloud Applications. *IEEE Communications Magazine*, April 2012.
3. W. Cohen et al. A comparison of string distance metrics for name-matching tasks. In *Proc. Workshop on Information Integration on the Web*, Acapulco, Mexico, 2003.

Learning to Predict Component Failures in Trains

Sebastian Kauschke¹, Immanuel Schweizer², Michael Fiebrig⁴, and Frederik Janssen³

¹ skauschke@gmail.com

Technische Universität Darmstadt, Germany

² schweizer@cs.tu-darmstadt.de

Telecooperation Group

Technische Universität Darmstadt, Germany

³ janssen@ke.tu-darmstadt.de

Knowledge Engineering Group

Technische Universität Darmstadt, Germany

⁴ michael.fiebrig@dbschenker.eu

DB Schenker

Head of Technical Management

Components Locomotives & Wagons (L.RBA 2 (A))

Abstract. Trains of *DB Schenker Rail AG* create a continuous logfile of diagnostics data. Within the company, methods to use this data in order to increase train availability and reduce costs are researched. An interesting and promising application is the prediction of train component failure.

In this paper, we developed and evaluated a method that utilizes the diagnostic data to predict future component failures. To do so, failure codes were aggregated and a flexible labeling scheme is introduced. In an extensive experiment section, three different failure types are examined, a combination of them is evaluated, and different parametrizations are inspected in more detail.

The results indicate that a prediction for all of the different types indeed is possible starting from days up to weeks ahead of the failure. However, the level of data-quality and its quantity still have to be increased considerably to yield high quality models.

1 Introduction

Each year, up to 400 million tons of goods are transported by *DB Schenker Rail AG*⁵. To achieve this, a fleet of over 3500 trains is available. Reliability is a crucial issue, since delivery dates have to be met. Complications may lead to delays and increase the cost of a transport. Mechanical failures of the train itself is one of the main reasons for not reaching destinations in time. Failures of a mechanical component are costly, since not only the component itself has to be replaced or repaired, but also the train is standing

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

⁵ www.rail.dbschenker.de

still and consequently has to be transported to a repair station, which causes difficulties with the global train schedule. Among others, these effects create additional costs.

To improve reliability, mechanical failure needs to be avoided as far as possible. Trains are regularly maintained according to a maintenance schedule, but since a train is a complex machine, not every component is checked each time. There are different kinds of inspections, some are smaller and occur more often and are used for small checkups. Others are large inspections that occur in longer intervals but with much more in-depth examination. Scheduled maintenance alone is therefore not able to provide constant monitoring of component status.

In other industries there have been successful implementations of constant monitoring and failure prediction such as logfile evaluation and engine temperature monitoring [1, 2]. It was shown that imminent failures can be detected early enough to avoid breakdowns. Fixed maintenance intervals can be replaced by on-demand maintenance, hence reducing cost and increasing reliability. The availability of the fleet will improve, allowing more goods to be transported. In this paper, we follow that path and present an approach to achieve failure prediction suited to the data provided by *DB Schenker Rail AG*.

The idea is to utilize the events occurring in the train’s computer systems. The various soft- and hardware systems of a modern train provide a continuous stream of such events, including status changes and errors. This is referred to as diagnostics data. Given such data, a method to predict failure of various kinds of components is developed. The approach uses the frequency of certain types of log entries in a variable time window before the actual failure to label a dataset and then train a classifier using the labeled data. The resulting model then is able to predict such failures on new data.

The method incorporates in-depth knowledge from engineers of *DB Schenker Rail AG*. The experts helped significantly to accelerate the data mining process involved by reducing the amount of data to a feasible subset. The topic has been discussed in more detail in the diploma thesis of Sebastian Kauschke [3]. In this paper, only a subset of the work presented there will be shown.

In the next section, the approach is detailed with a focus on the employed pre- and postprocessing of the data. Then, the experimental setup is sketched (Section 3 and the results are shown in Section 4. The following section provides related work and in Section 6 the paper is concluded and future research is given.

2 Prediction of Failures based on Diagnostic-Code Frequency

The proposed approach uses the frequency of diagnostic-code occurrence as a measure to detect anomalies in the train’s behavior and predict failures. The existing diagnostics data is used to create the features. We start by giving some details on the diagnostic data. Then a hypothesis is proposed, followed by a detailed description of the preprocessing of the data.

2.1 Diagnostics Data

Since the 1990s trains are equipped with on-board computers that connect and control the various systems. A train of the so-called ”Baureihe 185” (*BR185*) for example

has 37 main systems including a total of 70 subsystems. Each of the activities in those systems and subsystems is recorded in the diagnostics data file. The diagnostic messages include, e.g., protocol messages, events, warnings, and errors as well as analog measurements. A total of over 6900 types of diagnostic messages exist for this train type.

Each diagnostic message consists of the diagnostic-code, the system and subsystem this code belongs to, timestamps when it occurred first and when it vanished (diagnostic messages may span a certain amount of time), as well as the *environment data*. The *environment data* can include status variables, analog measurements, or simply plain text.

Please note that the data provided to conduct this research suffers from low coverage in regard to the recorded timespan which made high-quality predictions a true challenge.

A feature consists of an aggregation of past occurrences of a certain diagnostic-code in a specific time-frame. This way, for each day and train a feature vector is produced that incorporates information about how often diagnostic-codes appeared in that time-frame. Those vectors are then labeled in two categories: *normal* and *warning* where *normal* is representing a point in time where no failure was imminent, and *warning* is denoting an imminent failure.

A classifier is then trained on the labeled data, and 5-fold cross-validation is used to evaluate the classifier.

2.2 Hypothesis

In the following we propose a hypothesis concerning the dependency between diagnostic-codes and failure events.

Hypothesis 1 *Failure of components of the train are preceded by an increased appearance of specific diagnostic-codes in a certain time-frame before the failure occurs. The failure can be predicted by detecting which diagnostic-codes show this kind of increase and how much they increase opposed to a situation with no imminent failure.*

Based on this hypothesis a method to predict failures is created by the use of machine learning. To proof the hypothesis it is necessary to answer the questions to what extent a time-frame before the breakdown must be examined, and how far in advance a failure can be predicted. Those values will be discovered experimentally.

2.3 Preparations

The available data spans 18 months in time and includes over 3.7 million diagnostic-codes in a total of 187 trains. In that timeframe the majority of failures happened, most of them being of little interest for this research. To decide which failures are of interest, the following criteria have to be met:

- the component has failed because of deterioration
- the component needs to be attached to one of the systems sending diagnostic messages

Table 1. Exemplary occurrences of diagnostic-codes

| Day: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|---|---|---|---|---|---|---|---|
| CodeA: | 1 | 0 | 2 | 1 | 1 | 3 | 2 | 0 |
| CodeB: | 1 | 1 | 4 | 3 | 2 | 4 | 2 | 2 |
| CodeC: | 2 | 4 | 2 | 1 | 1 | 2 | 4 | 3 |

- the failing component causes a certain amount of cost or affects the trains ability to move
- the date of the failure needs to be known

At *DB Schenker Rail AG* a database including all repair station activities exists. It is mainly used for accounting purposes, but in this case it was used to search for activities that indicate failures which fit the required criteria. The database contains information about the dates the failures happened and on which trains they happened.

From the failures meeting the criteria, the three most frequent ones in the given timeframe were chosen:

- *ASG instand setzen* (repair motor control, found 142 times)
- *LZB instand setzen* (repair guiding system, found 126 times)
- *LZB Empfangsantenne einstellen* (repair antenna of guiding system, found 93 times)

2.4 Preprocessing

At first a decision has to be made which diagnostics-codes are relevant for the failure to be predicted. In an ideal environment, all existing diagnostics-codes (over 6900) would be used as features at first. This feature set can later be reduced by methods that single out the necessary features for the specific failure prediction.

In this case an expert was questioned and stated only diagnostic-codes of the system the failing component belongs to should be examined. With this information, the 30 most frequent diagnostics-codes of the relevant systems in the timeframe before the failure are chosen to be the features. The frequency of each code occurrence per train per day is calculated and will be used for the further steps.

2.5 Aggregation of features

A feature consists of the frequency of a diagnostic-code $CodeX$.

In Table 1 an exemplary amount of occurrences per day is shown for $CodeA$, $CodeB$, and $CodeC$.

For each day and each diagnostic-code a vector A_x is calculated. In the following example $x = 3$ is chosen:

$$A_x = \begin{pmatrix} CodeA \\ CodeB \\ CodeC \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 2 \end{pmatrix}$$

This Vector sums up the occurrences of all diagnostics-codes on day x . To achieve a time-span evaluation, a vector V is built, accumulating the vectors A_{x-v}, \dots, A_x by averaging the values. For $x = 5$ and $v = 3$ this is:

$$V_{x,v} = \begin{pmatrix} (0 + 2 + 1 + 1)/4 \\ (1 + 4 + 3 + 2)/4 \\ (4 + 2 + 1 + 1)/4 \end{pmatrix} = \begin{pmatrix} 1 \\ 2.5 \\ 2 \end{pmatrix}$$

2.6 Labeling

Each of the three failure-types will be handled as a separate classification problem. For every day x and every train a vector $V_{x,v}$ is calculated as described in Section 2.5, covering the v days preceding x . For each case of a failure a warning-timeframe is defined. The warning-timeframe ends with the day of the failure (S) and is of length w .

The vectors are labeled the following way:

1. For all failure dates: vectors in the range of $S - w, \dots, S$ that belong to the defunct train are labeled as *warning*.
2. All other vectors are labeled as *normal*

The vectors consist of a total of 30 features.

2.7 Postprocessing

Since the diagnostics data used for this experiment is incomplete and not all the days in the 18 month time interval are covered for every train, there is a necessity to filter out days that have no recorded entries at all. This means, if there is no entry on a specific day for a train, the data on this day will be marked as *invalid*. It will be excluded from vector generation and will not be used for training and evaluation.

This is especially problematic for the labeling process, since 50 % - 75 % of the time there are no or few data recordings available for the *warning* vectors. When the coverage of data for a specific failure date is lower than 50 %, all the vectors in the *warning*-timespan are marked as *invalid* and not used for training and evaluation.

3 Experimental Setup

In the following the algorithms used in the experiments are summarized. Then, a total of five different experiments are defined.

3.1 Training and Evaluation

The WEKA Suite [7] is used for training and evaluation. The classifiers *JRip*, *J48*, *RandomForest*, and *SMO* are chosen for classification. This selection ensures that most symbolic algorithms are present and that also a statistical classifier is employed.

The WEKA parameterization of the classifiers were set as follows:

Table 2. v and w values for experiment 1 and 2

| (a) Exp. 1: Motor Control Failure | | | | (b) Exp. 2: Guiding System Failure | | | |
|-----------------------------------|-----|-----|--------------------|------------------------------------|-----|-----|--------------------|
| Configuration | v | w | number of failures | Configuration | v | w | number of failures |
| a | 5 | 3 | 49 | a | 5 | 3 | 85 |
| b | 10 | 5 | 46 | b | 10 | 5 | 87 |
| c | 30 | 10 | 40 | c | 30 | 10 | 89 |

JRip: *Folds: 3, minNo: 2, Optimizations: 2.*

J48: *confidenceFactor: 0.25, minNumObj: 2, subtreeRaising: true.*

RandomForest: *maxDepth: unlimited, numFeatures: unlimited, numTrees: 10.*

SMO: *C: 300, Tolerance: 0.001; Epsilon: 1E-12, Kernel: Polykernel (Cache Size: 250007; E=5).*

To verify the results, 5-fold cross-validation was used. The main performance indicators are the *precision* and *recall* values of the *warning* class. Usually, *accuracy* is a good measurement for the performance of a classifier, but in this case *accuracy* is over 97 % in most cases. The reason for this are the unbalanced classes. The *warning* class is a minority class by a factor of up to 100. Another measure that reflects the performance of the classifier is the area under curve (*AUC*) value. We decided to include this evaluation measure instead of regular accuracy as it is capable to incorporate unbalanced classes.

3.2 Five Experiments

A total of five experiments were conducted during our study: One experiment for each of the chosen failure types, and one experiment for a combination of all three failure types. The fourth experiment is used to show that a discrimination between the three types of failure is possible. In the last experiment a wider range of the v and w values is examined, to show where the peak potential can be achieved.

Experiment 1: Motor Control Failure This experiment is based on the failure of the *Antriebssteuergerät (ASG)*, which is the motor control unit of the train. It manages power distribution to the four electric engines. If it fails, it is often the case that one of the two power converters shuts down, which results in a 50 % power loss. This may stop the train from moving, depending on how heavy it is loaded.

For this type of failure, 142 instances have been recorded. After postprocessing the labeled data, only a certain number of those is still considered *valid* according to the criteria described in Section 3. This depends on how the timeframes are chosen for the w and v to build the vectors. Three combinations of v and w were chosen (see Table 2 (a)).

Table 3. v and w values for experiment 3 and 4

| (a) Exp. 3: Guiding System Antenna Failure | Configuration | | | (b) Exp. 4: Combining all three Failure Types | Configuration | | |
|--|---------------|-----|--------------------|---|---------------|------------|--------------------|
| | v | w | number of failures | | v | w | number of failures |
| a | 5 | 3 | 17 | 30 | 10 | 97 (total) | |
| b | 10 | 5 | 17 | | | | |
| c | 30 | 10 | 20 | | | | |

Experiment 2: Guiding System Failure This experiment is based on the failure of the *Linienförmige Zugbeeinflussung (LZB)*, which is one of the guiding systems used by *Deutsche Bahn*. It allows to coordinate the positions of all trains and achieve faster driving speeds. If it fails, the train driver is forced to rely on other guiding systems, which can reduce overall speed or cause other problems.

For the guiding system failure, 126 instances were recorded. Only a certain amount of those is still considered *valid* after postprocessing, according to the criteria described above that are dependent on w and v . For this experiment, also three combinations of v and w were chosen (cf. Table 2 (b)).

Experiment 3: Guiding System Antenna Failure The *LZB* has an antenna, which provides it with the information it needs to work. If the antenna fails, the *LZB* will cease to function properly, and the same restrictions as in experiment 2 will apply.

93 instances of antenna failure have been recorded. After postprocessing a major amount is considered *invalid*. The following combinations of v and w were chosen (see Table 3 (a)).

Experiment 4: Combining all three Failure Types In this experiment a multi-class approach is used to show that discriminating between the three types of failures also is possible. To achieve this, the feature generation and labeling process are altered.

In the feature generation step, the number of features is increased. The required features for the three failure types partly overlap, so a mixture of features is used to allow every failure to be predicted adequately.

The labeling process is adapted in the following way:

1. For all motor control failure dates: vectors in the range of the *warning* timeframe are labeled *warning1*.
2. For all guiding system failure dates: vectors in the range of the *warning* timeframe are labeled *warning2*.
3. For all guiding system antenna failure dates: vectors in the range of the *warning* timeframe are labeled *warning3*.
4. All other vectors are labeled as *normal*

As the $v = 30$ and $w = 10$ combination of the previous experiments showed the best results, for this experiment only this combination is used (Table 3 (b)).

Table 4. Results for Motor Control Failure

| (a) $v = 5, w = 3$ | | | | | (b) $v = 10, w = 5$ | | | | |
|--------------------|--------------|--------------|-------------|--------------|---------------------|--------------|--------------|-------------|--------------|
| Classifier | Precision | Recall | F1-Score | AUC | Classifier | Precision | Recall | F1-Score | AUC |
| JRip | 0.333 | 0.076 | 0.12 | 0.534 | JRip | 0.592 | 0.229 | 0.33 | 0.623 |
| J48 | 0.167 | 0.010 | 0.02 | 0.513 | J48 | 0.678 | 0.150 | 0.25 | 0.609 |
| RandomF. | 0.610 | 0.171 | 0.27 | 0.820 | RandomF. | 0.835 | 0.380 | 0.52 | 0.904 |
| SMO | 0.517 | 0.148 | 0.23 | 0.573 | SMO | 0.612 | 0.226 | 0.33 | 0.612 |

| (c) $v = 30, w = 10$ | | | | |
|----------------------|--------------|--------------|-------------|--------------|
| Classifier | Precision | Recall | F1-Score | AUC |
| JRip | 0.728 | 0.675 | 0.70 | 0.838 |
| J48 | 0.752 | 0.577 | 0.65 | 0.905 |
| RandomF. | 0.895 | 0.667 | 0.76 | 0.968 |
| SMO | 0.757 | 0.487 | 0.59 | 0.742 |

Experiment 5: Extended Timeframes The last experiment is conducted to show which parameter combination of v and w achieves the highest *AUC*. *RandomForest* is used as the only classifier as it showed the best overall performance in all previous experiments. The failure type of experiment 1 is used for the evaluation. This choice is somewhat arbitrary and we believe that the parameters are indeed subject to change among different experimental setting, but, however, for demonstration purposes and due to space restrictions, we had to choose one of the above experiments.

The former experiments showed the highest results in the (c) configuration, which was the largest settings of v and w . To allow a better overview, if the performance can be further increased, a grid of *AUC* values is generated, with v values ranging from 20 to 100, and w values from 10 to 50, each in incremental steps of 10.

4 Results

In this section the results of the experiments will be shown and discussed. Note that for both *precision* and *recall*, the values are shown for the *warning* class. As the *AUC* is insensitive to imbalanced class distributions, we preferred this type of measure over regular *accuracy*.

4.1 Results Experiment 1: Motor Control Failure

In Table 4 the *RandomForest* classifier shows the overall highest performance. For all classifiers performance increases come along with larger timespan v and warning time w . *RandomForest* reaches *AUC* values of up to 0.904 (0.76 F1-score) in configuration (c). *JRip*, *J48* and *SMO* deliver similar *precision* values, but lack *recall* and their *AUC* values are lower than *RandomForest*.

All classifiers show a significant increase of *recall* with each step up in timespans, while *precision* interestingly does also increase.

Table 5. Results for Guiding System Failure

| (a) $v = 5, w = 3$ | | | | | (b) $v = 10, w = 5$ | | | | |
|--------------------|--------------|--------------|-------------|--------------|---------------------|--------------|--------------|-------------|--------------|
| Classifier | Precision | Recall | F1-Score | AUC | Classifier | Precision | Recall | F1-Score | AUC |
| JRip | 0.351 | 0.077 | 0.13 | 0.536 | JRip | 0.500 | 0.224 | 0.31 | 0.624 |
| J48 | 1.000 | 0.071 | 0.13 | 0.537 | J48 | 0.586 | 0.148 | 0.24 | 0.633 |
| RandomF. | 0.684 | 0.154 | 0.25 | 0.818 | RandomF. | 0.832 | 0.345 | 0.49 | 0.926 |
| SMO | 0.403 | 0.172 | 0.24 | 0.585 | SMO | 0.620 | 0.384 | 0.47 | 0.691 |

| (c) $v = 30, w = 10$ | | | | |
|----------------------|--------------|--------------|-------------|--------------|
| Classifier | Precision | Recall | F1-Score | AUC |
| JRip | 0.750 | 0.690 | 0.72 | 0.843 |
| J48 | 0.789 | 0.693 | 0.74 | 0.895 |
| RandomF. | 0.942 | 0.701 | 0.80 | 0.983 |
| SMO | 0.773 | 0.810 | 0.79 | 0.903 |

Table 6. Results for System Antenna Failure

| (a) $v = 5, w = 3$ | | | | | (b) $v = 10, w = 5$ | | | | |
|--------------------|--------------|--------------|-------------|--------------|---------------------|--------------|--------------|-------------|--------------|
| Classifier | Precision | Recall | F1-Score | AUC | Classifier | Precision | Recall | F1-Score | AUC |
| JRip | 0.568 | 0.292 | 0.39 | 0.668 | JRip | 0.581 | 0.443 | 0.5 | 0.741 |
| J48 | 1.000 | 0.069 | 0.13 | 0.546 | J48 | 0.652 | 0.155 | 0.25 | 0.591 |
| RandomF. | 0.706 | 0.167 | 0.27 | 0.858 | RandomF. | 0.800 | 0.412 | 0.54 | 0.957 |
| SMO | 0.556 | 0.278 | 0.37 | 0.639 | SMO | 0.556 | 0.361 | 0.44 | 0.680 |

| (c) $v = 30, w = 10$ | | | | |
|----------------------|--------------|--------------|-------------|--------------|
| Classifier | Precision | Recall | F1-Score | AUC |
| JRip | 0.766 | 0.695 | 0.73 | 0.867 |
| J48 | 0.822 | 0.550 | 0.66 | 0.845 |
| RandomF. | 0.916 | 0.649 | 0.76 | 0.959 |
| SMO | 0.732 | 0.669 | 0.70 | 0.834 |

4.2 Results Experiment 2: Guiding System Failure

The results shown in Table 5 are similar to those of experiment 1. With a *AUC* value of up to 0.983 and a F1-score of 0.8 in the widest timeframe (configuration (c)), the results of *RandomForest* are the highest of all classifiers. *SMO* achieves the highest *recall* value in all configurations, but the *precision* of 77.7 % is lower than *RandomForests* 94.2 %. *J48* and *JRip* show similar results with *AUC* values of 69 %.

The significant increase of the *recall* value with each step up is also evident for all the classifiers. However, *J48* has its best *precision* at the (a) configuration.

4.3 Results Experiment 3: System Antenna Failure

RandomForest shows high values in *recall* and *precision* in Table 6 which is also shown by an *AUC* value of 0.959 for configuration (c). *JRip* achieves the highest *recall* value of

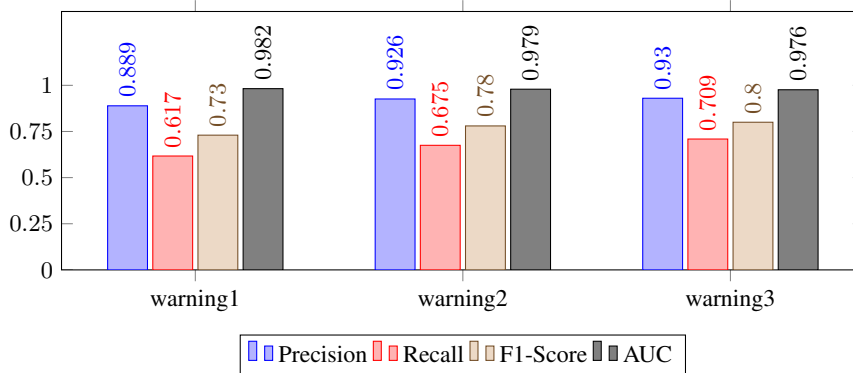


Fig. 1. Results for Experiment 4: RandomForest Classifier; $v = 30$, $w = 10$

Table 7. Experiment 4: Confusion Matrix for RandomForest

| | classified as | | | |
|----------|---------------|------------|------------|------------|
| | normal | warning1 | warning2 | warning3 |
| normal | 19753 | 25 | 16 | 7 |
| warning1 | 128 | 209 | 2 | 0 |
| warning2 | 107 | 1 | 226 | 1 |
| warning3 | 44 | 0 | 0 | 107 |

69.5 % and the second highest *AUC* value of 0.867 while outperforming *RandomForest* in F1-score at least in configuration (a). *J48* has a similar *AUC* of 0.845 and *SMO* has 0.834.

The significant increase of *recall* with bigger timeframes is also present here. All classifiers except *J48*, which shows a similar trend as before, achieve their best results in configuration (c).

4.4 Results Experiment 4: Combining all three Failure Types

Figure 1 shows that all of the three different warning types could be predicted with a high *precision* of 88.9 %, 92.6 % and 97.6 %, respectively while also the *recall* was above 60 %. The F1-score never falls below 0.7. Therefore, we can conclude that our hypothesis posed in Section 2.2 seems to be valid. In the confusion matrix depicted in Table 7 it can be seen that only a total of 1.3 % of all *normal* cases are classified incorrectly and that the actual warnings are predicted quite accurately. However, about a third of them is falsely classified as *normal*, which explains the *recall* values of 60 % to 70 %.

4.5 Results Experiment 5: Extended Timeframes

In the last experiment, different configurations for the days taken into account (v) and the number of days before the failure that are labeled as failure (w) were examined.

Table 8. Results for Extended Timeframes

| (a) AUC | | | | | (b) F1-score | | | | | | |
|---------|--------------|-------|-------|-------|--------------|-------|--------------|--------------|-------|--------------|--------------|
| | w=50 | w=40 | w=30 | w=20 | w=10 | | w=50 | w=40 | w=30 | w=20 | w=10 |
| v=100 | 0.991 | 0.995 | 0.994 | 0.984 | 0.982 | v=100 | 0.999 | 0.998 | 0.998 | 0.999 | 0.998 |
| v=90 | 0.992 | 0.995 | 0.99 | 0.992 | 0.989 | v=90 | 0.998 | 0.998 | 0.998 | 0.998 | 0.998 |
| v=80 | 0.999 | 0.996 | 0.993 | 0.995 | 0.992 | v=80 | 0.999 | 0.999 | 0.998 | 0.999 | 0.999 |
| v=70 | 0.997 | 0.997 | 0.996 | 0.995 | 0.992 | v=70 | 0.998 | 0.998 | 0.998 | 0.998 | 0.999 |
| v=60 | 0.992 | 0.989 | 0.995 | 0.99 | 0.99 | v=60 | 0.997 | 0.997 | 0.998 | 0.998 | 0.998 |
| v=50 | 0.99 | 0.989 | 0.988 | 0.99 | 0.972 | v=50 | 0.996 | 0.996 | 0.997 | 0.998 | 0.998 |
| v=40 | 0.982 | 0.984 | 0.991 | 0.988 | 0.981 | v=40 | 0.995 | 0.995 | 0.996 | 0.997 | 0.997 |
| v=30 | 0.978 | 0.979 | 0.978 | 0.976 | 0.968 | v=30 | 0.994 | 0.994 | 0.994 | 0.996 | 0.996 |
| v=20 | 0.966 | 0.97 | 0.973 | 0.972 | 0.967 | v=20 | 0.991 | 0.992 | 0.993 | 0.994 | 0.995 |

Table 8 (a) shows that an increase of the v and w values can further improve the AUC . The highest result is given at $v = 80$ and $w = 50$ with an AUC of 0.999 compared to the original value achieved in experiment 1 (cf. Section 4.1) which was 0.968. Interestingly a different picture manifests when inspecting the F1-score. Here, there is no single best configuration but a total of seven ones achieved the highest value (cf. Table 8 (b)). Among them, also the best one for AUC is present, but, however, it seems that F1-score is not so sensitive to these two parameters.

5 Related Work

This section will provide an overview of work that was used directly or as an inspiration for the methods described in this paper. There are certain parallels, so some of the methods were adapted and adjusted to fit the specific problems. However, related work is rather rare as usually the algorithms are tied to a specific problem and it is hard to generalize to arbitrary scenarios.

Fulp, Fink and Haack [1] proposed a method based on the evaluation of system logfiles to predict hard disc failure. The logfile data was used to train a support-vector machine to recognize sequences or patterns in the messages which implied an impending failure. With a *sliding window* method subsequences of the log were evaluated and classified as *fail* or *non-fail*. The training data consisted of the actual system log of a linux computing cluster. The results showed promising results with up to 73 % recognition rate up to two days ahead of the failure.

The work of by Létourneau, Famili and Matwin [4] originated from the aircraft domain. The authors examined the problematics of large amounts of data generating systems in an airplane. They addressed the issues of data gathering, labeling, and model integration and presented an approach to learn models from the data to predict issues with components of the aircraft.

In a Phd. thesis of Lipowsky [5], the author focused on condition monitoring of gas turbines. The differences between handling gradually occurring degradations and spontaneous failures were elaborated. An integrated system to deal with both cases

was developed, one based on a least-squares solution, the other based on nonlinear optimization.

6 Conclusions

In this paper we proposed a method to predict component failures based on system logs. The results show, that such a type of prediction indeed is possible. However, the effort to reach this goal is high. For every type of failure a separate classifier has to be trained, although the results of experiment 4 (cf. Section 4.4) show that a combined classification is possible. Nevertheless, such a combination still is prone to result in a reduced classification performance compared to treating each problem separately. Also, and perhaps most importantly, the requirement for methods such as proposed in this paper, namely data quality is not yet met. To assure a consistent data environment for the training, the data needs to be complete for the whole fleet. The data used for this research had huge gaps in the recorded timespan. This is a problem *DB Schenker Rail* has to solve, before methods like this can be used in a real-world environment. We also assume that the prediction quality will significantly increase given the data quality is improved.

Also, the problem of imbalanced classes is certainly present in domains such as failure prediction as usually cases of failure are quite rare compared to the regular cases where everything is fine. The results show that the *RandomForest* classifier seems to work well on imbalanced data, but, if tackled appropriately the performance will even increase.

For future work it is planned to carefully tune the parameters of the machine learning algorithms for each single problem. Also, given the data quality is enhanced, the approach has to be re-run on the new data. Another interesting topic is to inspect the effects of the values v and w also for the other experiments and figure out whether or not similar trends are present.

References

1. Fulp, E.W., Fink, G.A., & Haack, J.N., Predicting Computer System Failures Using Support Vector Machines. *WASL'08 Proceedings of the First USENIX conference on Analysis of system logs*, 2008.
2. Guo, P., & Bai, N., Wind Turbine Gearbox Condition Monitoring With AAKR And Moving Window Statistic Methods. *Energies* 2011, 4, 2077-2093., 2011.
3. Kauschke, S., Nutzung Bahnbezogener Sensordaten Zur Vorhersage Von Wartungszyklen, Diploma Thesis, TU Darmstadt, Knowledge Engineering Group, http://www.ke.tu-darmstadt.de/lehre/arbeiten/diplom/2014/Kauschke_Sebastian.pdf, 5 2014.
4. Létourneau, S., Famili, F. & Matwin, S., Data Mining For Prediction of Aircraft Component Replacement. *IEEE Intelligent Systems Jr., Special Issue on Data Mining*, p. 59-66, 1999.
5. Lipowsky, H., *Entwicklung Und Demonstration Eines Integrierten Systems Zur Zustandsüberwachung Von Gasturbinen*, Phd. Thesis, Stuttgart, Institut für Luftfahrtantriebe, 2010.
6. Swets, J., *Signal Detection And Recognition By Human Observers*, New York, Wiley, 1964.
7. Witten, I.H., Frank, E., & Hall, M., *Data Mining - Practical Machine Learning Tools And Techniques*. Burlington: Morgan Kaufmann Publishers, 3rd edition, 2011.

Route Planning with Real-Time Traffic Predictions

Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik

TU Dortmund University, Dortmund, Germany,
{firstname.lastname}@tu-dortmund.de

Abstract. Situation dependent route planning gathers increasing interest as cities become crowded and jammed. We present a system for individual trip planning that incorporates future traffic hazards in routing. Future traffic conditions are computed by a Spatio-Temporal Random Field based on a stream of sensor readings. In addition, our approach estimates traffic flow in areas with low sensor coverage using a Gaussian Process Regression. The conditioning of spatial regression on intermediate predictions of a discrete probabilistic graphical model allows to incorporate historical data, streamed online data and a rich dependency structure at the same time. We demonstrate the system with a real-world use-case from Dublin city, Ireland.

Resubmission of: [12] Predictive Trip Planning - Smart Routing in Smart Cities. In: Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference, vol. 1133, pp. 331–338. CEUR-WS.org (2014)

Keywords: trip planning, real-time traffic model, traffic flow estimation

1 Introduction

The incentive for the creation of smart cities is the increase of living quality and performance of the city. This is often accompanied with various mobile phone apps or web services to bring new services to the people of a city – advertising events, spreading city information or guiding people to their destinations by providing smart trip planning based on the city’s spirit.

With the unpleasant trend of growing congestion in modern urban areas, smart route planing becomes an essential service in the smart city development. Existing trip planning systems consider current traffic hazards and historical speed profiles which are recorded by personal position traces and mobile phone network data [19].

The fast moving traffic situations in urban areas demand for a thorough routing that incorporates as fresh information about the city’s infrastructure as

Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

possible. This current work, originally presented in [12], presents an approach to *situation dependent trip planning* that incorporates real time information gained from smart city sensors and combines this data with a model for estimating future traffic situations for route calculation. The proposed system provides three components: (1) an interactive web-based user interface that is based on the popular *OpenTripPlanner* project [16]. The web interface allows for users to specify start and target location and triggers the route planning and provides a REST-ful service (REpresentation State Transfer, introduced in [18]) interface to integrate such services into mobile applications. (2) A real-time backend engine, based on the *streams* framework [3], which provides data stream processing for various types of data. We provide input adapters for *streams* to read and process SCATS data [1] emitted from automatic traffic loops (city sensors). This allows us to maintain an up-to-date view of the city’s current traffic state. (3) A sophisticated dynamic traffic model that is integrated into the backend stream engine and which provides traffic flow estimation at unobserved locations at future times.

The combination of these components is a trip planner that incorporates the latest traffic state information as well as using a fine-grained future traffic flow estimation for urban trip planning. We test our trip planner in a use case scenario in the city of Dublin. The city is amongst the most jammed cities in Europe. The city holds about 966 SCATS sensors, each providing current traffic flow and vehicle speed at the sensor location.

The paper is structured as follows. In the second section we describe the general architecture of the presented system regarding the input and output of the trip planner, the data analysis and the stream processing connecting middleware. The third section deals with the application of our proposed trip planner to a use case in Dublin, Ireland. In the fourth section we provide a discussion of the work together with future directions. The fifth section presents related work.

2 General Architecture

We give an overview of the system developed to address the veracity, velocity and sparsity problems of urban traffic management. The system has been developed as part of the INSIGHT project. This section describes the input and output of the system, the individual components that perform the data analysis, and the stream processing connecting middleware.

2.1 System Components

As already noted in the introduction, we built the system aiming real time streaming capabilities. Based on the *streams* framework, the core engine is a data flow graph that models the data stream processing of the incoming SCATS data. This graph can easily be defined by means of the *streams* XML configuration language and features the integration of custom components directly into the

data flow graph. As can be seen in Figure 1, this data flow graph contains the SCATS data source as well as several nodes that represent preprocessing operations. A crucial component within that stream processing is our Spatio-Temporal Random Field (STRF) implementation¹, which is used in combination with the sensor readings to provide a model for traffic flow prediction.

With the *service layer* API provided by *streams*, we export access to the traffic prediction model to the OpenTripPlanner component. The OpenTripPlanner provides the interface to let the user specify queries for route planning. Based on a given query (v, w) with a starting location v and a destination w , it computes the optimal route $v \rightarrow p_0 \dots p_k \rightarrow w$ based on traffic costs. Here we plug in a cost-model for the routing that is based on the traffic flow estimation and the current city infrastructure status. This cost-model is queried by OpenTripPlanner using the *service layer* API.

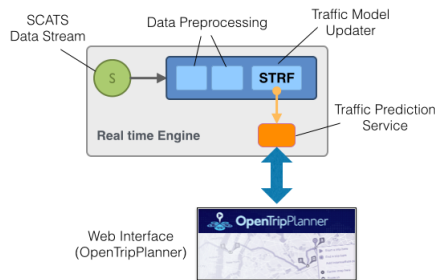


Fig. 1. A general overview of the components of the predictive trip planning system. The real time engine continuously computes traffic condition forecasts and exports the prediction service to the OpenTripPlanner. Best viewed in color.

2.2 Traffic Model

The key component of our system is the traffic model. It combines two machine learning methods in a novel way, in order to achieve traffic flow predictions for nearly arbitrary locations and points in time. This traffic model addresses multiple facets of the trip planning problem:

- sparsity of stationary sensor readings among the city,
- velocity of real-time traffic readings and computation, and
- veracity of future traffic flow predictions.

Based on a stream of observed sensor measurements, a Spatio-Temporal Random Field [17] estimates the future sensor values, whereas values for non-sensor

¹ The C++ implementation of STRF and the JNI interface can be found at: <http://sfb876.tu-dortmund.de/strf>

locations are estimated using Gaussian Processes [14]. To the best of the authors knowledge, streamed STRF+GP prediction has not been considered until now and is therefore a novel method for traffic modelling.

Spatio-Temporal Random Field for Flow Prediction In order to model the temporal dynamics of the traffic flow as measured by the SCATS sensors (Figure 3), a Spatio-Temporal Random Field is constructed. The intuition behind STRF is based on sequential probabilistic graphical models, also known as linear chains, which are popular in the natural language processing community. There, consecutive words or corresponding word features are connected to a sequence of labels that reflects an underlying domain of interest like entities or part of speech tags. If a sensor network, represented by a spatial graph $G_0 = (V_0, E_0)$, is considered that generates measurements over space and time, it is appealing to identify the joint measurement of all sensors with a single word in a sentence and connect those structures to form a temporal chain $G_1 - G_2 - \dots - G_T$. Each part $G_t = (V_t, E_t)$ of the temporal chain replicates the given *spatial graph* G_0 , which represents the underlying physical placement of sensors, i.e., the spatial structure of random variables that does not change over time. The parts are connected by a set of spatio-temporal edges $E_{t-1;t} \subset V_{t-1} \times V_t$ for $t = 2, \dots, T$ and $E_{0;1} = \emptyset$, that represent dependencies between adjacent snapshot graphs G_{t-1} and G_t , assuming a Markov property among snapshots, so that $E_{t;t+h} = \emptyset$ whenever $h > 1$ for any t . The resulting spatio-temporal graph G , consists of the snapshot graphs G_t stacked in order for time frames $t = 1, 2, \dots, T$ and the temporal edges connecting them: $G := (V, E)$ for $V := \cup_{t=1}^T V_t$ and $E := \cup_{t=1}^T \{E_t \cup E_{t-1;t}\}$.

Finally, G is used to induce a generative probabilistic graphical model that allows us to predict (an approximation to) each sensors maximum-a-posterior (MAP) state as well as the corresponding marginal probabilities. The full joint probability mass function is given by

$$p_{\theta}(\mathbf{X} = \mathbf{x}) = \frac{1}{\Psi(\theta)} \prod_{v \in V} \psi_v(\mathbf{x}) \prod_{(v,w) \in E} \psi_{(v,w)}(\mathbf{x}).$$

Here, \mathbf{X} represents the random state of all sensors at all T points in time and \mathbf{x} is a particular assignment to \mathbf{X} . It is assumed that each sensor emits a discrete value from a finite set \mathcal{X} . By construction, a single vertex v corresponds to a single SCATS sensor s at a fixed point in time t . The potential function of an STRF has a special form that obeys the smooth temporal dynamics inherent in spatio-temporal data.

$$\psi_v(\mathbf{x}) = \psi_{s(t)}(\mathbf{x}) = \exp \left\langle \sum_{i=1}^t \frac{1}{t-i+1} \mathbf{Z}_{s,i}, \phi_{s(t)}(\mathbf{x}) \right\rangle$$

The STRF is therefore parametrized by the vectors $\mathbf{Z}_{s,i}$ that store one weight for each of the $|\mathcal{X}|$ possible values for each sensor s and point in time $1 \leq i \leq T$. The function $\phi_{s(t)}$ generates an indicator vector that contains exactly one 1 at the position of the state that is assigned to sensor s at time t in \mathbf{x} and

zero otherwise. For a given data set, the parameters \mathbf{Z} are fitted by regularized maximum-likelihood estimation.

As soon as the parameters are learned from the data, predictions can be computed via MAP estimation,

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}_{V \setminus U} \in \mathcal{X}} p_{\theta}(\mathbf{x}_{V \setminus U} \mid \mathbf{x}_U), \quad (1)$$

where $U \subset V$ is a set of spatio-temporal vertices with known values. The nodes in U are termed observed nodes. Notice that $U = \emptyset$ is a perfectly valid choice that yields the most probable state for each node, given no observed nodes. To compute this quantity, the sum-product algorithm [10] is applied, often referred to as loopy belief propagation (LBP). Although LBP computes only approximate marginals and therefore MAP estimation by LBP may not be perfect [8], it suffices our purpose.

Gaussian Process Model for Flow Imputation Based on the discrete estimates of the STRF, we model the junction based traffic flow values within a Gaussian Process regression framework, similar to the approach in [14]. In the traffic graph each junction corresponds to one vertex. To each vertex v_i in the graph, we introduce a latent variable f_i which represents the true traffic flow at v_i . The observed traffic flow values are conditioned on the latent function values with Gaussian noise ϵ_i : $y_i = f_i + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

We assume that the random vector of all latent function values follows a Gaussian Process (GP), and in turn, any finite set of function values $\mathbf{f} = f_i : i = 1, \dots, M$ has a multivariate Gaussian distribution with mean and covariances computed with mean and covariance functions of the GP. The multivariate Gaussian prior distribution of the function values \mathbf{f} is written as $P(\mathbf{f} \mid \mathbf{X}) = \mathcal{N}(0, K)$, where K is the so-called kernel and denotes the $M \times M$ covariance matrix, zero mean is assumed without loss of generality.

For traffic flow values at unmeasured locations u , the predictive distribution can be computed as follows. Based on the property of GP, the vector of observed traffic flows (v at locations $-u$) and unobserved traffic flows (f_u) follows a Gaussian distribution

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_u \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} \hat{K}_{-u,-u} + \sigma^2 I & \hat{K}_{-u,u} \\ \hat{K}_{u,-u} & \hat{K}_{u,u} \end{bmatrix} \right), \quad (2)$$

where $\hat{K}_{u,-u}$ are the corresponding entries of \hat{K} between the unobserved vertices u and observed ones $-u$. $\hat{K}_{-u,-u}$, $\hat{K}_{u,u}$, and $\hat{K}_{-u,u}$ are defined equivalently. I is an identity matrix of size $|-u|$.

Finally the conditional distribution of the unobserved traffic flows are still Gaussian with the mean m and the covariance matrix Σ : $m = \hat{K}_{u,-u}(\hat{K}_{-u,-u} + \sigma^2 I)^{-1} \mathbf{y}$, $\Sigma = \hat{K}_{u,u} - \hat{K}_{u,-u}(\hat{K}_{-u,-u} + \sigma^2 I)^{-1} \hat{K}_{-u,u}$.

Since the latent variables \mathbf{f} are linked together in a graph \mathcal{G} , it is obvious that the covariances are closely related to the network structure: the variables

are highly correlated if they are adjacent in \mathcal{G} , and vice versa. Therefore we can employ graph kernels [23] to denote the covariance functions $k(x_i, x_j)$ among the locations x_i and x_j , and thus the covariance matrix.

The work in [14, 13] describes methods to incorporate knowledge on preferred routes in the kernel matrix. Lacking this information, we decide for the commonly used regularized Laplacian kernel function $K = [\beta(L + I/\alpha^2)]^{-1}$, where α and β are hyperparameters. L denotes the combinatorial Laplacian, which is computed as $L = D - A$, where A denotes the adjacency matrix of the graph \mathcal{G} . D is a diagonal matrix with entries $d_{i,i} = \sum_j A_{i,j}$.

2.3 OpenTripPlanner

OpenTripPlanner (OTP) is an open source initiative for route calculation. The traffic network for route calculation is generated using data from OpenStreetMap and (eventually) public transport schedules. Thus, OpenTripPlanner allows route calculation for multiple modes of transportation including walking, bicycling, transit or its combinations. However, vehicular routing is possible, but for data quality reasons in OpenStreetMap concerning the turning restrictions [20] it is not advisable. The default routing algorithm in OTP is the A* algorithm which utilizes a cost-heuristic to prune the Dijkstra search.

OpenTripPlanner consists of two components an API and a web application which interfaces the API using RESTful services. The API loads the traffic network graph, and calculates the routes. The web application provides an interactive browser based user interface with a map view. A user of the trip planner can form a trip request by selecting a start and a target location on the map.

2.4 The streams Framework

The need for real time capabilities in today’s data processing and the steady decrease of latency from data acquisition to knowledge extraction or information use from that data led to a growing demand for general purpose stream processing environments. Several such frameworks have evolved – *Storm*, *Kafka* or Yahoo!’s *S4* engine are among the most popular open-source approaches to streaming data. They all feature slightly different APIs and come with slightly different philosophies. Focusing on a more middle-layer approach is the *streams* framework proposed in [3], which aims at providing a light-weight high-level abstraction for defining data flow networks in an easy-to-use XML configuration. It comes with its own execution engine, but also features the transparent execution of data flow graphs on existing engines such as *Storm*. We base our decision for the *streams* framework on its recent applications that highlight its high throughput capabilities [5] and the built-in data mining operators [2].

SCATS Data Processing with streams Within the *streams* framework, a data source is represented as a sequences of data items, which in turn are sets of key-value pairs, i.e. event attributes and their values. Processes within a *streams*

data flow graph consume data items from streams and apply functions onto the data. The data flow graph for manipulation, analysis and filtering of the streams is formulated in an XML-based language that *streams* provides. A sample XML configuration is given in Figure 2.

```

<container>
  <stream id="scats:data" url="http://..." class="eu.insight.input.ScatsStream" />
  <process input="scats:data">
    <!-- .. custom functions .. -->
    <eu.insight.data.DataNormalization />
    <eu.insight.traffic.TrafficEstimator id="predictor" />
  </process>
</container>

```

Fig. 2. XML representation of a streams container with a source for SCATS data and a process that applies a normalization to each data item and then forwards it to a traffic estimation processor.

The process setup of Figure 2 defines a single data source that provides a stream of SCATS sensor data. A *process* is attached to this source and continuously reads items from that source. For each of the data item, it applies a sequence of custom functions (so called *processors*) that reflect data transformations or other actions on the items. In the example above, we include a SCATS specific *DataNormalization* step as well as our custom *TrafficEstimator* implementation directly into the data flow graph.

Service Level API The *streams* runtime provides a simple RMI-based service invocation of data flow components that do provide remote services. The *TrafficEstimator* defines such a remote interface and is automatically registered as a service with identifier “*predictor*”. This allows service methods of that estimator to be asynchronously called from outside the data flow graph, i.e. from within our modified *OpenTripPlanner* component.

The service method that is defined by the *TrafficEstimator* is exactly the cost-retrieval function that is required within the A^* algorithm of the *OpenTripPlanner*:

$$getCost(x, y, t)$$

where x and y are the longitude and latitude of the location and t is the time at which the traffic flow for (x, y) shall be predicted.

3 Empirical Evaluation

In this section we present the application of our proposed trip planner to a use case in Dublin, Ireland. We used real data streams obtained from the SCATS sensors of Dublin city. The stream was collected between January and April

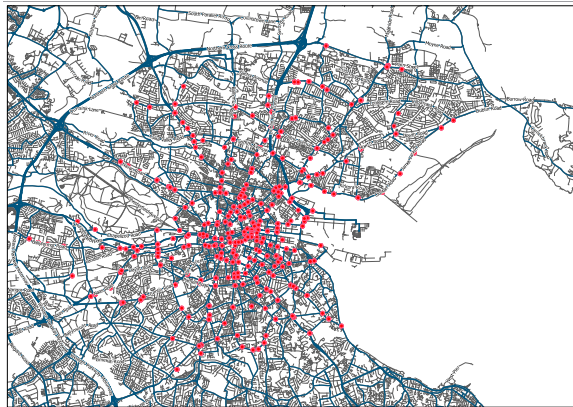


Fig. 3. Locations of SCATS sensors (marked by red dots) within Dublin, Ireland. Best viewed in color.

2013 and comprises ≈ 9 GB of data. The SCATS dataset includes 966 sensors, see Figure 3 for their spatial distribution among the traffic network. SCATS sensors transmit information on traffic flow every six minutes. The data set is publicly available².

For the experiments in Dublin, the traffic network is generated based on the OpenStreetMap³ data. In the preprocessing step the network is restricted to a bounding window of the city size. Next, every street is split at any junction in order to retrieve street segments. In result we obtain a graph that represents the traffic network. The SCATS locations are mapped to their nearest neighbours within this street network.

In the preprocessing step the sensor readings are aggregated within fixed time intervals. We tested various intervals and decided for 30 minutes, as lower aggregates are too noisy, caused by traffic lights and sensor fidelity.

The spatial graph G_0 that is required for the STRF is constructed as k -nearest-neighbor (k NN) graph of the SCATS sensor locations. In what follows, a 7NN graph is used, since a smaller k induces graphs with large disconnected components and a larger k leads to more complex models without improving the performance of the method. The fact that no information about the actual street network is used to build G_0 might seem counterintuitive, but undirected graphical models like STRF do not use or rely on any notion of flow. They rather make use of conditional independence, i.e. the state of any node v can be computed if the states of its neighboring nodes are known. Thus, the k NN graph can capture long-distance dependencies that are not represented in the actual street network connectivity. The maximum traffic flow value that is measured by each SCATS sensor in each 30-minutes-window is discretized into one of 6 consecutive intervals. A separate STRF model for each day of the week is

² Dublin SCATS data: <http://www.dublinked.ie>

³ OpenStreetMap: <http://www.openstreetmap.org>

constructed and each day is further partitioned into 48 snapshot graphs, since we can divide a day into 48 blocks of 30 minutes length. The model parameters are estimated on SCATS data between January 1 and March 31 2013 and evaluated using data from April 2013.

The evaluation data is streamed as observed nodes into the STRF which computes a new conditioned MAP prediction (Equation 1) for all unobserved vertices of the spatio-temporal graph G whenever time proceeds to the next temporal snapshot. The discrete predictions are then de-discretized by taking the mean of the bounds of the corresponding intervals and subsequently forwarded to the Gaussian Process which uses these predictions to predict values at non-sensor locations. Notice that although the discretization with subsequent de-discretization seems inconvenient at a first glance, it allows the STRF to model any non-linear temporal dynamics of the sensor measurements, i.e. the flow at a fixed sensor might change instantly if the sensor is located close to a factory at shift changeover.



Fig. 4. Results of route calculations for fixed start and target at different timestamps (from left to right: 7:00, 8:00, 8:30). Best viewed in color.

Application of Gaussian Processes requires a joint multivariate Gaussian distribution among the considered random variables. In our case, these random variables denote the traffic flow per junction. Literature on traffic flow theory [11, 4] tested traffic flow distributions and supports a hypothesis for a joint log-normal distribution. We test our dataset for this hypothesis. Thus, we apply the Mardia [15] normality test to the preprocessed data set. The test checks multivariate skewness and kurtosis. We apply the implementation of the Mardia test contained in the R package MVN [9]. The tests confirmed the hypothesis that the recorded traffic flow (obtained from the SCATS system) is lognormal distributed. Thus, application of Gaussian Processes to log-transformed traffic flow values is possible. The hyper-parameters for the GP are chosen in advance using a grid search. Best performance was achieved with $\alpha = 1/2$ and $\beta = 1/2$. The STRF provides complete knowledge on future sensor readings which is necessary for our GP. As the STRF model performs well [17], we set the noise among the sensor data in our GP to a small variance of 0.0001. For easy tractability, we set up the GP to model about 5000 locations among the city of Dublin.

The OpenTripPlanner creates a query for the costs at a particular coordinate in space-time. The query is transmitted from the route calculation to the traffic model. There, the query is matched to the discrete space. The spatial coordinates

are encoded in the WGS84 reference system. To avoid precision problems during the matching between the components, the spatial coordinate is matched with a nearest neighbour method using a KDTree data structure. The nearest neighbour matching offers also the possibility to query costs for arbitrary locations. The timestamp of the query is discretized to one of the 48 bins we applied in the STRF.

We apply our trip planner for a particular Monday in data set (8th April 2013) and compute routes from a fixed start to a fixed target at different time stamps. Figure 4 shows that different routes are calculated depending on the traffic situation. Congested street segments are avoided and different routes are suggested.

4 Related Work

Previous sections already discussed related approaches. Here, we present briefly recent work on dynamic cost estimation for trip planning in smart cities. Recent work [24] predicts the travel time of routes. As their work evaluates a particular predefined route based on recorded GPS traces, it has a related but different scope. In contrast to [25], our approach combines the STRF with a GP for estimation of costs at unobserved locations. The approach in [6] addresses travel time forecasts based on the delays in the public transportation system. Main drawback of their method is that buses have extra lanes at most junctions and their movement follows a regular pattern. The inclusion of traffic loop readings was motivated in their section on future work. The dynamic traffic flow estimation is a major problem in traffic theory. Common approach is the usage of a k-Nearest Neighbour algorithm which calculates traffic flow estimates as weighted average of the k nearest observations [7]. In contrast, our approach models future traffic flow values based on their temporal patterns, correlations and dependencies. Foremost, our model requires less memory as k-NN which has to store all previously seen sensor values for continuous traffic flow estimation. Another paper that compares two prediction models for traffic flow estimation is presented in [21]. By combining a Gauss Markov Model with a Gaussian Process, their work provides a faster model which is suitable for near time predictions (as required for automatic signal control). The model estimates future values by consecutive application of the model. In contrast, the hereby presented work estimates all future time slices at once. In result, we built valuable trip planner application on top of the traffic estimation model and highlighted its usability. Improvement of the estimation method, and comparison of estimation accuracy is subject for future work.

5 Discussion and Future Work

Within this paper we presented a novel approach for trip planning in highly congested urban areas. Our approach computes intelligent routes that avoid traffic hazards in advance. The proposed trip planner consists of a continuous traffic

model based on real-time sensor readings and a web based user interface. We combined the real-time traffic model and the trip calculation with a streaming backbone. We applied the trip planner to a real-world use case in the city of Dublin, Ireland.

Our traffic model combines latest advances in traffic flow estimation. On the one hand, prediction of future sensor values is performed with a spatio-temporal random field, which is trained in advance. Based on these estimates, the traffic flow for unobserved locations is performed by a Gaussian Process Regression. We successfully applied the Regularized Laplacian Kernel. In literature, also other kernels have been successfully applied to the problem, [13, 22]. Exploration of different kernel methods is subject for future research.

Besides the SCATS data also other data sources provide useful information for dynamic cost estimation. The integration of bus travel times or user generated (crowdsourcing and social network) data in our model is possible by dynamically changing the traffic network (in case of road blockages) or introducing dynamic weights (in case of a accident or flooding on a street segment). Future studies need to explore these directions.

Acknowledgements This work is funded by the following projects: EU FP7 INSIGHT (318225); the Deutsche Forschungsgemeinschaft within the CRC SFB 876 Providing Information by Resource-Constrained Data Analysis, A1 and C1.

References

1. SCATS. *Sydney Coordinated Adaptive Traffic System*, Available: <http://www.scats.com.au/> [Last accessed: 27 June 2013] (2013)
2. Bockermann, C., Blom, H.: Processing Data Streams with the RapidMiner Streams-Plugin. In: Proceedings of the 3rd RapidMiner Community Meeting and Conference (2012)
3. Bockermann, C., Blom, H.: The streams framework. Tech. Rep. 5, TU Dortmund University (12 2012), <http://jwall.org/streams/tr.pdf> [Last accessed: 28 November 2013]
4. Davis, G.: estimation theory approach to monitoring and updating average daily traffic. Tech. Rep. mn/rc 97-05, minnesota department of transportation, office of research administration (january 1997)
5. Gal, A., Keren, S., Sondak, M., Weidlich, M., Blom, H., Bockermann, C.: Grand challenge: The techniball system. In: Proceedings of the 7th ACM International Conference on Distributed Event-based Systems. pp. 319–324. DEBS '13, ACM, New York, NY, USA (2013)
6. Gasparini, L., Bouillet, E., Calabrese, F., Verscheure, O., O'Brien, B., O'Donnell, M.: System and analytics for continuously assessing transport systems from sparse and noisy observations: Case study in dublin. In: Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. pp. 1827–1832 (2011)
7. Gong, X., Wang, F.: Three Improvements on KNN-NPR for Traffic Flow Forecasting. In: Proceedings of the 5th International Conference on Intelligent Transportation Systems. pp. 736–740. IEEE Press (2002), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1041310&tag=1

8. Heinemann, U., Globerson, A.: What cannot be learned with bethe approximations. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence. Barcelona, Spain (2011)
9. Kormaz, S.: MVN: Multivariate Normality Tests (2013), <http://CRAN.R-project.org/package=MVN>, r package version 1.0
10. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory* 47(2), 498–519 (2001)
11. Lay, G.: Handbook of Road Technology, Fourth Edition. taylor & francis (2009)
12. Liebig, T., Piatkowski, N., Bockermann, C., Morik, K.: Predictive trip planning - smart routing in smart cities. In: Proceedings of the Workshops of the EDBT/ICDT 2014 Joint Conference (EDBT/ICDT 2014), Athens, Greece, March 28, 2014. vol. 1133, pp. 331–338. CEUR-WS.org (2014)
13. Liebig, T., Xu, Z., May, M.: Incorporating mobility patterns in pedestrian quantity estimation and sensor placement. In: Nin, J., Villatoro, D. (eds.) Citizen in Sensor Networks, Lecture Notes in Computer Science, vol. 7685, pp. 67–80. Springer Berlin Heidelberg (2013)
14. Liebig, T., Xu, Z., May, M., Wrobel, S.: Pedestrian quantity estimation with trajectory patterns. In: Flach, P.A., Bie, T., Cristianini, N. (eds.) Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science, vol. 7524, pp. 629–643. Springer Berlin Heidelberg (2012)
15. Mardia, K.V.: Measures of multivariate skewness and kurtosis with applications. *Biometrika* 57, 519–530 (1970)
16. McHugh, B.: The opentripplanner project. Tech. Rep. Metro RTO Grant Final Report, TriMet (August 2011), <http://portlandtransport.com/documents/OTP\%20Final\%20Report\%20-%20Metro\%202009-2011\%20RTO\%20Grant.pdf>
17. Piatkowski, N., Lee, S., Morik, K.: Spatio-temporal random fields: compressible representation and distributed estimation. *Machine Learning* 93(1), 115–139 (2013)
18. Richardson, L., Ruby, S.: RESTful Web Services. O’Reilly Series, O’Reilly Media, Incorporated (2007), <http://books.google.de/books?id=XUaErakHsoAC>
19. Schäfer, R.P.: IQ Routes and HD Traffic: Technology Insights About Tomtom’s Time-dynamic Navigation Concept. In: Proceedings of the the 7th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering. pp. 171–172. ESEC/FSE ’09, ACM, New York, NY, USA (2009)
20. Scheider, S., Possin, J.: Affordance-based individuation of junctions in open street map. *Journal of Spatial Information Science* 4(1), 31–56 (2012)
21. Schnitzler, F., Liebig, T., Mannor, S., Morik, K.: Combining a gauss-markov model and gaussian process for traffic prediction in dublin city center. In: Proceedings of the Workshop on Mining Urban Data at the International Conference on Extending Database Technology. p. (to appear) (2014)
22. Selby, B., Kockelman, K.M.: Spatial prediction of traffic levels in unmeasured locations: applications of universal kriging and geographically weighted regression. *Journal of Transport Geography* 29, 24–32 (May 2013)
23. Smola, A., Kondor, R.: Kernels and regularization on graphs. In: Proc. Conf. on Learning Theory and Kernel Machines. pp. 144–158 (2003)
24. Wang, Y., Zheng, Y., Xue, Y.: Travel time estimation of a path using sparse trajectories. In: KDD 2014. ACM (August 2014), <http://research.microsoft.com/apps/pubs/default.aspx?id=217493>
25. Yang, B., Guo, C., Jensen, C.S.: Travel cost inference from sparse, spatio temporally correlated time series using markov models. *Proc. VLDB Endow.* 6(9), 769–780 (July 2013), <http://dx.doi.org/10.14778/2536360.2536375>

Integrating Input from Human Experts into Prototype-based Classifier Learning

Petra Perner

Institute of Computer Vision and Applied Computer Sciences, IBaI
PSF 301114, 04251 Leipzig, Germany
pperner@ibai-institut.de

Abstract. An expert is able to tell the system developer in many image-related tasks what a prototypical image should look like. Usually he will choose several prototypes for one class, but he cannot provide a good and large enough sample set for the class to train a classifier. Therefore, we mapped his technical procedure into a technical system based on proper theoretical methods that assist him in acquiring the knowledge about his application and furthermore in developing a classifier for his task. This system helps him to learn about the clusters and the borderlines of the clusters even when the data are very noisy as is the case for microscopic cell images in drug discovery, where it is unclear if the drug will produce the expected result on the cell parts.

We describe in this paper the necessary functions that a prototype-based classifier should have. We also use the expert's estimated similarity as a new knowledge piece and based on that we optimize the similarity. The test of the system was carried out on a new application on microscopic cell image analysis - the study of the internal mitochondrial movement of cells. The aim was to discover the different dynamic signatures of mitochondrial movement. Three results of this movement were expected: tubular, round, and dead cells. Based on our results we can show the success of the developed method.

Keywords: Internal Mitochondrial Movement, Cell Biology, Similarity Measure, Case-Based Reasoning, Prototype-Based Classification, Knowledge Acquisition, Feature Subset Selection, Prototype Selection, Adjustment Theory

1 Introduction

Prototypical classifiers have been successfully studied for medical applications by Schmidt and Gierl [1], by Perner [2] for image interpretation and by Nilsson and Funk [3] on time-series data. The simple nearest-neighbor approach [4], as well as hierar-

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

chical indexing and retrieval methods [5], have been applied to the problem. It has been shown that an initial reasoning system could be built up based on prototypical cases. The systems are useful in practice and can acquire new cases for further reasoning during utilization of the system. Prototypical images are a good starting point for the development of an automated image classifier [6]. This knowledge is often collected by human experts in image catalogues. We describe, based on a task for the study of the internal mitochondrial movement of cells [7], how such a classifier in combination with image analysis can be used for incremental knowledge acquisition and automatic classification. The work enhances our previous work on a prototype-based classifier [2] by introducing the expert's estimated similarity as a new knowledge piece and a new function that adjusts this similarity and the automatically calculated similarity by the system in order to improve the system accuracy. The test of the system is done on a new application on cell image analysis - the study of the internal mitochondrial movement of cells.

The classifier is set up based on prototypical cell appearances in the image such as for e.g. „healthy cell“, „dead cell“, and „cell in transition stage“. For these prototypes are calculated image features based on a random set theory that describes the texture on the cells. The prototype is represented then by the feature-value pair and the class label. These settings are taken as initial classifier settings, in order to acquire the knowledge about the dynamic signatures.

The importance of the features and the feature weights are learned by the proto-class-based classifier [2]. After the classifier is set up each new cell is then compared by the proto-class-based classifier and the similarity to the prototypes is calculated. If the similarity is high the new cell gets the label of the prototype. If the similarity to the prototypes is too low, then there is evidence that the cell is in transition stage and a new prototype has been found. With this procedure we can learn the dynamic signature of the mitochondrial movement.

In Section 2 we present the methods for our prototype-based classifier. The material is described in Section 3 for the internal mitochondrial movement of cells. In Section 4 is presented the methodology for the knowledge acquisition based on a prototype-based classification. Results are given in Section 5 and finally in Section 6 conclusions are presented.

2 ProtoClass Classifiers

A prototype-based classifier classifies a new sample according to the prototypes in the data base and selects the most similar prototype as output of the classifier. A proper similarity measure is necessary to perform this task, but in most applications there is no a-priori knowledge available that suggests the right similarity measure. The method of choice to select the proper similarity measure is therefore to apply a subset of the numerous similarity measures known from statistics to the problem and to select the one that performs best according to a quality measure such as, for example, the classification accuracy. The other choice is to automatically build the similarity metric by learning the right features and feature weights. The latter one we chose as one option to improve the performance of our classifier.

When people collect prototypes to construct a dataset for a prototype-based classifier, it is useful to check if these prototypes are good prototypes. Therefore a function is needed to perform prototype selection and to reduce the number of prototypes used for classification. This results in better generalization and a more noise-tolerant classifier. If an expert selects the prototypes, this can result in bias and possible duplicates of prototypes causing inefficiencies. Therefore a function to assess a collection of prototypes and identify redundancy is useful.

Finally, an important variable in a prototype-based classifier is the value used to determine the number of closest cases and the final class label.

Consequently, the design-options for the classifier to improve its performance are prototype selection, feature-subset selection, feature weight learning and the 'k' value of the closest cases (see Figure 1).

We assume that the classifier can start in the worst case with only one prototype per class. By applying the classifier to new samples the system collects new prototypes. During the lifetime of the system it will chance its performance from an oracle-based classifier, which will classify the samples roughly into the expected classes, to a system with high performance in terms of accuracy.

In order to achieve this goal we need methods that can work on a low number of prototypes and on large number of prototypes. As long as we have only a few prototypes feature subset selection and learning the similarity might be the important features the system needs. If we have more prototypes we also need prototype selection.

For the case with a low number of prototypes we chose methods for feature subset selection based on the discrimination power of features. We use the feature based calculated similarity and the pair-wise similarity rating of the expert and apply the adjustment theory [11] to fit the similarity value more to the true value.

For a large number of prototypes we choose a decremental redundancy-reduction algorithm proposed by Chang [8] that deletes prototypes as long as the classification accuracy does not decrease. The feature-subset selection is based on the wrapper approach [9] and an empirical feature-weight learning method [10] is used. Cross validation is used to estimate the classification accuracy. A detailed description of our prototype-based classifier ProtoClass is given in [2]. The prototype selection, the feature selection, and the feature weighting steps are performed independently or in combination with each other, in order to assess the influence these functions have on the performance of the classifier. The steps are performed during each run of the cross-validation process.

The classifier schema shown in Figure 1 is divided in the design phase (Learning Unit) and the normal classification phase (Classification Unit). The classification phase starts after we have evaluated the classifier and determined the right features, feature weights, the value for 'k' and the cases.

Our classifier has a flat data base instead of a hierarchical one that makes it easier to conduct the evaluations.

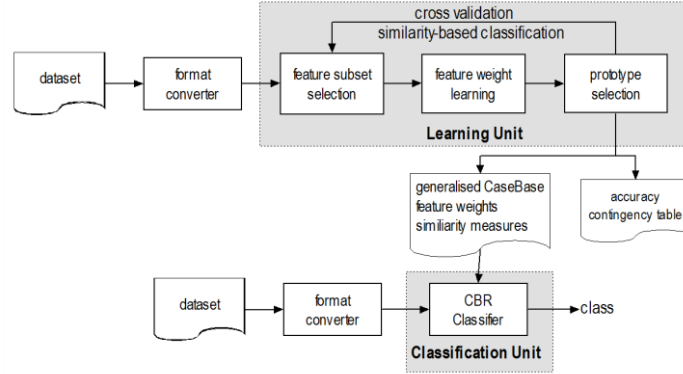


Fig. 1. Prototype-based Classifier

2.1 Classification Rule

Assume we have n prototypes that represent m classes of the application. Then, each new sample is classified based on its closeness to the n prototypes. The new sample is associated with the class label of the prototype that is the closest one to sample.

More precisely, we call $x'_n \in \{x_1, x_2, \dots, x_i, \dots, x_n\}$ a closest case to x if $\min d(x_i, x) = d(x'_n, x)$, where $i=1, 2, \dots, n$.

The rule chooses to classify x into category C_l , where x'_n is the closest case to x and x'_n belongs to class C_l with $l \in \{1, \dots, m\}$.

In the case of the k -closest cases we require k samples of the same class to fulfill the decision rule. As a distance measure we can use any distance metric. In this work we used the city-block metric.

The pair-wise similarity measure Sim_{ij} among our prototypes shows us the discrimination power of the chosen prototypes based on the features.

The calculated feature set must not be the optimal feature subset. The discrimination power of the features must be checked later. For a low number of prototypes we can let the expert judge the similarity $SimE_{ij}$ between $i, j \in \{1, \dots, n\}$ the prototypes. This gives us further information about the problem which can be used to tune the designed classifier.

2.2 Using Expert's Judgment on Similarity and the Calculated Similarity to Adjust the System

Humans can judge the similarity $SimE_{ij}$ among objects on a rate between 0 (identity) and 1 (dissimilar). We can use this information to adjust the system to the true system parameters [11].

Using the city-block distance as distance measure, we get the following linear system of equations:

$$SimE_{ij} = \frac{1}{N} \sum_{l=1}^N a_l |f_{il} - f_{jl}| \quad (1)$$

with $i, j \in \{1, \dots, n\}$, f_{il} the feature l of the i -th prototype and N the number of features.

The feature a_l is the normalization of the feature to the range $\{0,1\}$ with $a_l = \frac{1}{|f_{\max,l} - f_{\min,l}|}$ that is calculated from the prototypes. That this is not the true range of the feature value is clear since we have a low number of samples. The factor a_l is adjusted closer to the true value by the least square method using expert's $SimE_{ij}$:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n \left(SimE_{ij} - \frac{1}{N} \sum_{l=1}^N a_l |f_{il} - f_{jl}| \right)^2 \Rightarrow Min! \quad (2)$$

with the restriction $0 \leq a_l \leq \frac{1}{|f_{\max,l} - f_{\min,l}|}$.

3 Methodology

Figure 2 summarizes the knowledge-acquisition process based on prototyped-based classification.

We start with one prototype for each class. This prototype is chosen by the biologist based on the appearance of the cells. It requires that the biologist has enough knowledge about the processes going on in cell-based assays and can decide what kind of reaction the cell is showing.

The discrimination power of the prototypes is checked first based on the feature values measured from the cells and the chosen similarity measure. Note that we calculated a large number of features for each cell. However, using many features does not mean that we will achieve a good discrimination power between the classes. It is better to come up with one or two features for small sample sizes in order to ensure a good performance of the classifier. The expert manually estimates the similarity between the prototypes and inputs these values into the system. The result of this process is the selection of the right similarity measure and the right number of features. With this information is set up a first classifier and applied to real data.

Each new data gets associated with the label of the classification. Manually we evaluate the performance of the classifier. The biologist gives the true or gold label for the sample seen so far. This is kept into a data base and serves as gold standard for further evaluation. During this process the expert will sort out wrongly classified data. This might happen because of too few prototypes for one class or because the samples should be divided into more classes. The decision what kind of technique should be applied is made based on the visual appearance of the cells. Therefore, it is necessary to display the prototypes of the classes and the new samples. The biologist sorts these samples based on the visual appearance. That this is not easy to do by humans is clear and needs some experiences in describing image information [6]. However, it is a standard technique in psychology, in particular in gestalts psychology, and known as categorizing or

card sorting. As a result of this process we come up with more prototypes for one class or with new classes and at least one prototype for these new classes.

The discrimination power needs to get checked again based on this new data set. New features, a new number of prototypes or a new similarity measure might be the output. The process is repeated as long as the expert is satisfied with the result. As a result of the whole process we get a data set of samples with true class labels, the settings for the proto-class-based classifier, the important features and the real prototypes. The class labels represent the categories of the cellular processes going on in the experiment. The result can now be taken as a knowledge acquisition output. Just for discovering the categories or the classifier can now be used in routine work at the cell-line.

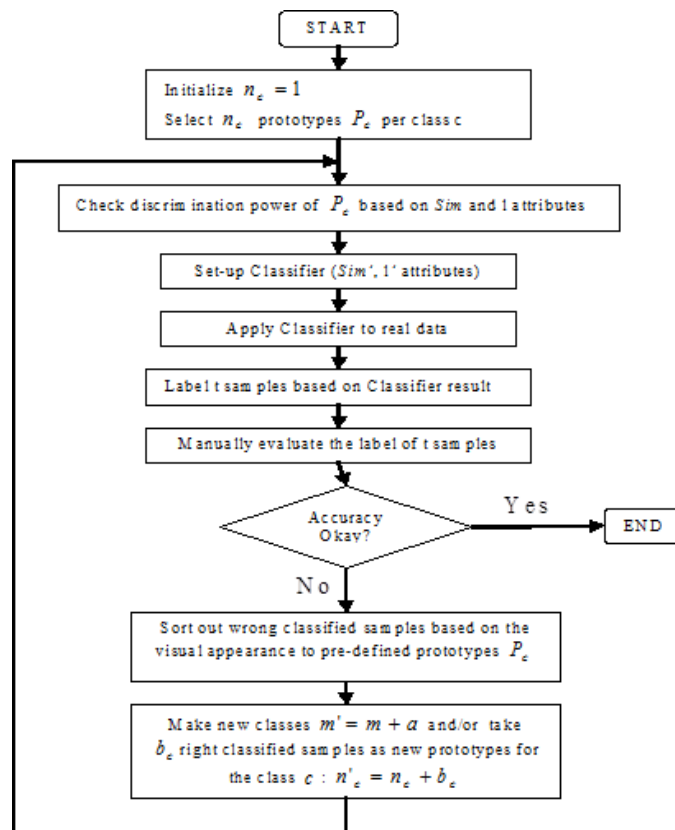


Fig. 2. Methodology for Prototype-based Classification

4 THE APPLICATION

After the assay has been set up, it is not quite clear what the appearances of the different phases of a cell are. This has to be learnt during the use of the system.

Based on their knowledge the biologists set up several descriptions for the classification of the mitochondria. They grouped these classes into the following classes: tubular cells, round cells and dead cells. For the appearance of these classes see images in Figure 3.

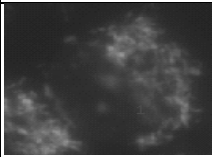
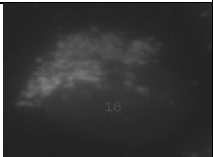
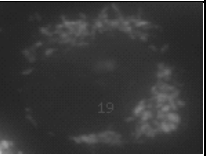
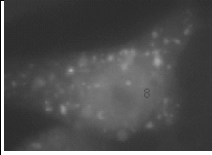
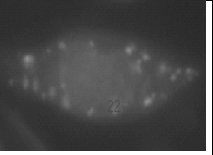
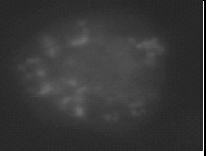

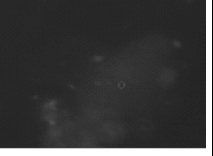
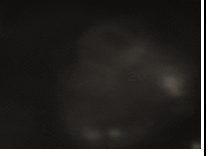
| Class Tubular | | |
|---|---|--|
| B10_1 | B10_18 | B10_19 |
|  |  |  |
| Class Round | | |
| B03_8 | B03_22 | B03_26 |
|  |  |  |
| Class Death | | |
| B03_11 | B06_0 | B06_20 |
|  |  |  |

Fig. 3. Sample Images for three Classes (top Class Tubular, middle Class Round, bottom Class Death)

Then prototypical cells were selected and the features were calculated with the software tool *CellInterpret* [12]. The expert rated the similarity between these prototypical images.

Our data set consist of 223 instances with the following class partition: 36 instances of class *Death*, 120 instances of class *Round*, 47 instances of class *Tubular*, and 114 features for each instance.

The expert chose for each class a prototype shown in Figure 4. The test data set for classification has then 220 instances. For our experiments we also selected 5 prototypes pro class respectively 20 prototypes pro class. The associate test data sets do not contain the prototypes.

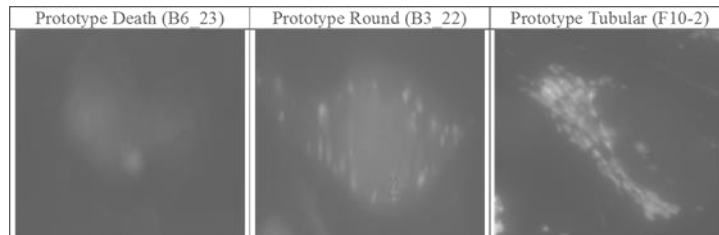


Fig. 4. The Prototypes for the classes Death, Round and Tubular

5 Results

Figure 5 shows the accuracy for classification based on different numbers of prototypes for all features and Fig. 6 shows the accuracy for a test set based on only the three most discriminating features. The test shows that the classification accuracy is not so bad for only three prototypes, but with the number of prototypes the accuracy increases. The selection of the right subset of features can also improve the accuracy and can be done based on the method presented in Section 2 for a low number of samples. The right chosen number of closest cases k can also help to improve accuracy, but cannot be applied if we only have three prototypes or less in the data base.

Figure 7 shows the classification results for the 220 instances started without adjustment meaning the weights are equal to one (1;1;1) and with adjustment based on expert's rating where the weights are (0.00546448; 0.00502579; 0.00202621) as an outcome of the minimization problem.

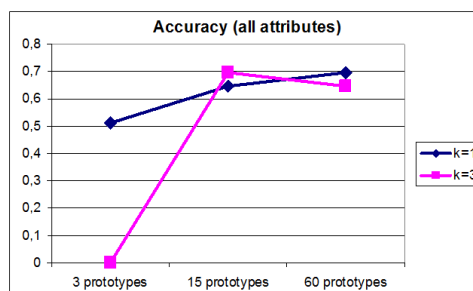


Fig. 5. Accuracy versus Prototypes and for two different feature subsets; Accuracy for different number of prototypes using all features

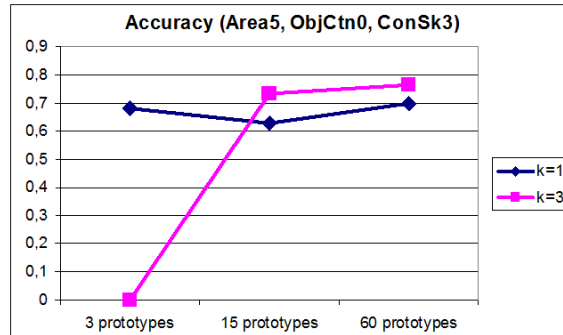


Fig. 6. Accuracy versus Prototypes and for two different feature subsets; Accuracy for different number of prototypes using 3 features (Area5, ObjCtn0, ConSk3)

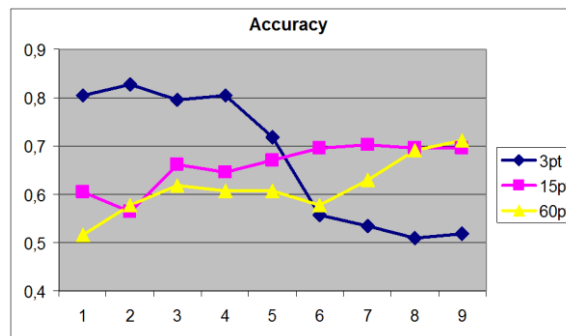


Fig. 7. Accuracy depending on choice of features (k=1)

Table 1. Difference values between 3 Prototypes using the 3 features (ObjCnt0, ArSig0, ObjCnt1) and the judged difference values by the expert

| | B6_23 | B03_22 | F10_2 |
|--------|----------------------|----------------------|----------------------|
| B6_23 | 0 | 0,669503257 (0,8) | 0,989071038 (0,6) |
| B03_22 | 0,669503257 (0,8) | 0 | 0,341425705 (0,9) |
| F10_2 | 0,989071038 (0,6) | 0,341425705 (0,9) | 0 |

Table 1 shows the difference values of three prototypes and in clips the judged difference values by the expert. The result shows that accuracy can be improved by applying the adjustment theory and especially the class specific quality is improved by applying the adjustment theory (see Fig. 8).

The application of the methods for larger samples set did not bring any significant reduction in the number of prototypes (see Fig. 9) or in the feature subset (see Fig. 10).

The prototype selection method reduced the number of prototypes only by three prototypes. We take it as an indication that we have not yet the enough prototypes and that the accuracy of the classifier can be improved by collecting more prototypes.

In Summary, we have shown that the chosen methods are valuable methods for a prototype-based classifier and can improve the classifier performance. For future work we will do more investigations on the adjustment theory as a method to learn the importance of features based on a low number of features and for feature subset selection for a low number of samples.

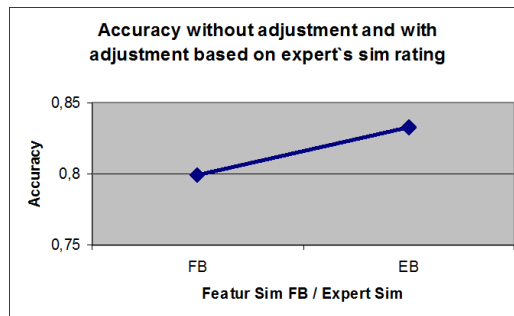


Fig. 8. Accuracy with and without adjustment theory

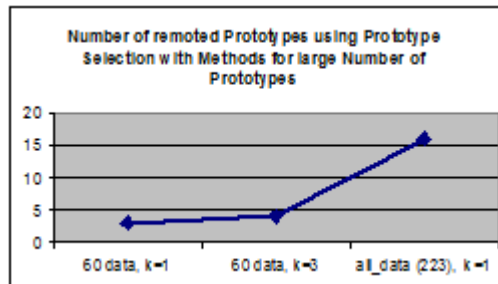


Fig. 9. Number of removed Prototypes

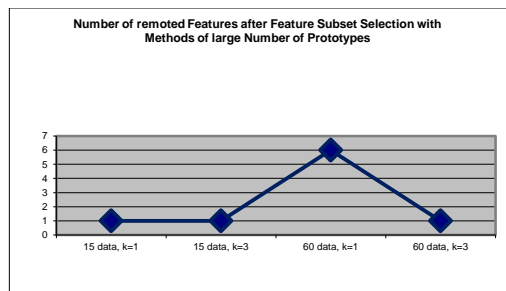


Fig. 10. Number of removed Features after Feature Subset Selection

6 CONCLUSIONS

We have presented our results on a prototype-based classification. Such a method can be used for incremental knowledge acquisition and classification. Therefore the classifier needs methods that can work on low numbers of prototypes and on on large numbers of prototypes. Our result shows that feature subset selection based on the discrimination power of a feature is a good method for low numbers of prototypes. The adjustment theory in combination with an expert similarity judgment can be taken to learn the true feature range in case of few prototypes. If we have a large number of prototypes an option for prototype selection is needed that can check for redundant prototypes.

The system can start to work on a low number of prototypes and can instantly collect samples during the usage of the system. These samples get the label of the closest case. The system performance improves the more prototypes the system has in its data base. That means an iterative process of labeled sample collection based on prototype based classification is necessary, followed by a revision of these samples after some time, in order to sort out wrongly classified samples until the system performance has been stabilized.

The test of the system is done on a new application on cell image analysis, the study of the internal mitochondrial movement of cells.

References

1. R. Schmidt and L. Gierl, "Temporal Abstractions and Case-Based Reasoning for Medical Course Data: Two Prognostic Applications," in *Machine Learning and Data Mining in Pattern Recognition, MLDM2001*, edited by P. Perner, Inai 2123, Springer-Verlag: Berlin Heidelberg, p. 23-34, 2001.
2. Perner, P.: *Prototype-Based Classification*, Applied Intelligence 28, 238-246 (2008)
3. M. Nilsson and P. Funk, "A Case-Based Classification of Respiratory Sinus Arrhythmia," in *Advances in Case-Based Reasoning, ECCBR 2004*, edited by P. Funk and P.A. Gonzalez Calero, Inai 3155, Springer-Verlag: Berlin Heidelberg, p. 673-685, 2004.
4. D.W. Aha, D. Kibler, and M.K. Albert, "Instance-based Learning Algorithm," *Machine Learning*, 6(1):37-66, 1991.
5. P. Perner, "Incremental Learning of Retrieval Knowledge in a Case-Based Reasoning System," in: K.D. Ashley and D.G. Bridge (Eds.), *Case-Based Reasoning - Research and Development*, Springer Verlag 2003, LNAI 2689, pp. 422-436
6. Sachs-Hombach, Kl.: *Bildbegriff und Bildwissenschaft*. In: Gerhardus, D., Rompza, S. (Eds.) kunst - gestaltung - design, Heft 8, pp. 1-38, Verlag St. Johann, Saarbrücken (2002)
7. Krausz E., Prechtel, St., Stelzer, E.H.K., Bork, P., Perner, P.: *Quantitative Measurement of dynamic time dependent cellular events*. Project Description (May 2006)
8. Chang, C.-L.: Finding Prototypes for Nearest Neighbor Classifiers. *IEEE Trans. on Computers*. C-23 (11) (1974)
9. Perner, P.: *Data Mining on Multimedia Data*. LNCS, vol. 2558, Springer Verlag (2002)

10. Little, S., Colantonio, S., Salvetti, O., Perner, P.: Evaluation of Feature Subset Selection, Feature Weighting, and Prototype Selection for Biomedical Applications. *J. Software Engineering & Applications* 3, 39-49 (2010)
11. Niemeier; W.: *Ausgleichsrechnung*, de Gruyter, Berlin New York (2008)
12. Perner, P.: *Novel Computerized Methods in System Biology -Flexible High-Content Image Analysis and Interpretation System for Cell Images*. In: Perner, P. Salvetti, O. (Eds.) *Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry*, MDA 2008, Inai, vol. 5108, pp. 139-157, Springer Verlag (2008)

Efficient Identification of Subspaces with Small but Substantive Clusters in Noisy Datasets (Extended Abstract)*

Frank Höppner

Ostfalia University of Applied Sciences
Dept. of Computer Science, D-38302 Wolfenbüttel, Germany

Abstract We propose an efficient filter approach (called ROSMULD) to rank subspaces with respect to their clustering tendency, that is, how likely it is to find areas in the respective subspaces with a (possibly slight but substantive) increase in density. Each data object votes for the subspace with the most unlikely high data density and subspaces are ranked according to the number of received votes. Data objects are allowed to vote only if the density significantly exceeds the density expected from the univariate distributions. Results on artificial and real data demonstrate efficiency and effectiveness of the approach.

1 Subspace Filtering

Data analysis typically starts with visualization and exploration of the data. Cluster analysis is a valuable tool to identify representative or prototypical cases that stand for a whole group of similar records in the dataset. However, for high-dimensional datasets that have not been collected with a specific analysis goal in mind, it is unlikely that the data nicely collapses into a small number of well-separated clusters. In fact, the whole data or large portions of it may not group at all. And it is quite likely that such groups manifest only in a low-dimensional subspace rather than having most attributes interacting with each other. In this work we consider an efficient approach to identify those subspaces of the dataset that disclose substantive clusters even though they may be small in size and hidden in a lot of noisy data.

While standard clustering algorithms consider all attributes as being (equally) relevant, *subspace clustering* interlocks the search for the appropriate subspace and the clusters themselves within the same algorithm [4,6]. The downside is that the notion of a cluster is strongly connected to the choice of the clustering algorithm, but the literature does not offer a subspace version for every clustering approach. Embedding the clustering algorithm into a search

* Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

for the best subspace may come at prohibitive computational costs. This work is in line with the few filter approaches that exist in the literature (e.g. [1,3]) which limit themselves to the efficient identification of promising subspaces only, leaving the further cluster analysis to subsequent steps.

When searching for potentially small clusters in a noisy environment, we face various problems: (1) If the clusters are relatively small, global correlation measures may respond to them only marginally such that the chosen thresholds are not passed. (2) Density variations in single variables alone may cause high-dimensional spots look dense (but do not establish an worthwhile high-dim. cluster). (3) Any kind of density estimation involves some kind of threshold selection (e.g. the sampling area) and the impact of the selection may be easily underestimated. (4) Many weapons to reduce runtime (e.g. subsampling) do not apply successfully if a clusters size is only a small fraction of the noise.

The new ROSMULD algorithm (**r**anking of subspaces by the **m**ost **u**nlikely high **l**ocal **d**ensity) overcomes these difficulties. By means of a rank-order transformation, all attributes become uniformly distributed, which eliminates density variations in single attributes. For each data point the subspace with the most surprisingly high data density is identified. Only if this density exceeds the expected density significantly, the data object votes for the respective subspace. Thresholds are automatically derived from the desired sensitivity (e.g. a cluster should have at least a density f times higher than the background noise). An exhaustive search for the most suprising subspace is avoided by employing new bounds on the used interestingness measure (without loosing completeness of the search).

ROSMULD successfully identifies subspaces with very small clusters and does not report any interesting subspace if the attributes are mutually independent. It performs also well on data sets with prominent and well-separated clusters. Compared to subspace clustering algorithms (cf. comparison in [5]) ROSMULD performs very competetive. For further details we refer to [2].

References

1. C. Baumgartner, K. Kailing, H.-P. Kriegel, P. Krüger, and C. Plant. Subspace Selection for Clustering High-Dimensional Data. In *ICDM*, 2004.
2. F. Höppner. A subspace filter supporting the discovery of small clusters in very noisy datasets. *Proc. 26th Int. Conf. on Scientific and Statistical Database Management - SSDBM '14*, 2014.
3. K. Kailing, H. Kriegel, P. Kröger, and S. Wanka. Ranking interesting subspaces for clustering high dimensional data. In *PKDD*, volume 2838, pages 241–252, 2003.
4. H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, Mar. 2009.
5. E. Müller, S. Günemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB*, 2(1):1270–1281, 2009.
6. K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong. A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery*, 26(2):332–397, Feb. 2012.

Subspace Search for Community Detection and Community Outlier Mining in Attributed Graphs

Emmanuel Müller

emmanuel.mueller@kit.edu

emmanuel.mueller@ua.ac.be

Karlsruhe Institute of Technology, Germany

University of Antwerp, Belgium

Attributed graphs are widely used for the representation of social networks, gene and protein interactions, communication networks, or product co-purchase in web stores. Each object is represented by its relationships to other objects (*edge structure*) and its individual properties (*node attributes*). For instance, social networks store friendship relations as edges and age, income, and other properties as attributes. These relationships and properties seem to be dependent on each other and exploiting these dependencies is beneficial, e.g. for community detection and community outlier mining. However, state-of-the-art techniques highly rely on this dependency assumption. In particular, *community outlier mining* [2] is able to detect an outlier node if and only if connected nodes have similar values in all attributes. Such assumptions are generally known as homophily [4] and are widely used. However, looking at multivariate spaces, one can observe that not all given attributes have high dependencies with the graph structure. For example, social properties such as income or age have strong dependencies with the graph structure of social networks [4]. In contrast, properties such as gender are rather independent from it. Consequently, recent graph mining algorithms degenerate for multivariate attribute spaces that lack dependency with the graph structure in some of the attributes. This calls for a general pre-processing step that selects subspaces, i.e. subsets of the attributes, showing dependencies with the graph.

This talk covers several methods for the selection of such relevant subspaces in attributed graphs:

As first method, *ConSub* [3] proposes the statistical selection of *congruent subspaces*, i.e. subsets of attributes showing a dependency with the graph structure. A core challenge in selecting these subspaces lies in the modeling of dependence between graph structure and attribute values. Further, one has to ensure that congruent subspaces are selected only if there is sufficient evidence on this dependence. *ConSub* addresses all those problems by: (1) a novel measure for the degree of congruence between a set of node attributes and a graph by means

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

of edge counts and attribute values; and (2) a comparison of edge counts in subgraphs constrained by attribute value ranges in a Monte Carlo processing. The congruence measure exploits these dependencies between random subgraphs and their attribute subspaces and *ConSub* selects attribute subsets featuring those dependencies in multivariate attribute spaces. This selection can serve as general pre-processing step for algorithms that rely on the homophily assumption on attributed graphs.

As second method, *FocusCO* [1] incorporates the user preference into the selection of relevant subspaces in attributed graphs. *FocusCO* considers communities and community outliers based on user preference. This *focused* mining is of particular interest in attributed graphs, where users might not be concerned with all but a few available attributes. As different attributes induce different clusters and outliers in the graph, the user should be able to steer the subspace selection accordingly. As such, the user controls the mining by providing a set of exemplar nodes (perceived similar by the user) from which *FocusCO* infers *attribute weights* of relevance that capture the user-perceived similarity. The essence of user preference is captured by those attributes with large weights, i.e. the *focus attributes*, which form the basis for the discovery of focused clusters and outliers.

To illustrate the applicability of common graph mining tasks and in order to evaluate these selection schemes, community detection and community outlier mining is used. The methods are evaluated on several synthetic and real world graphs, in particular on a novel benchmark graph for attributed graphs that has been derived from a case study on the Amazon co-purchase network [5]. The selection of congruent subspaces clearly enhances outlier detection by measuring outlierness scores in selected subspaces only. Furthermore, focused attributes enable a more user-oriented mining of community structures. Experiments show that both approaches outperform traditional full space approaches and as general pre-processing steps they enhance the later data mining steps on attributed graphs.

References

1. Bryan, P., Akoglu, L., Iglesias, P., Müller, E.: Focused clustering and outlier detection in large attributed graphs. In: ACM SIGKDD (2014)
2. Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., Han, J.: On community outliers and their efficient detection in information networks. In: ACM SIGKDD. pp. 813–822 (2010)
3. Iglesias, P., Müller, E., Laforet, F., Keller, F., Böhm, K.: Statistical selection of congruent subspaces for mining attributed graphs. In: IEEE ICDM. pp. 647–656 (2013)
4. McPherson, M., Lovin, L.S., Cook, J.M.: Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology* 27(1), 415–444 (2001)
5. Müller, E., Iglesias, P., Mülle, Y., Böhm, K.: Ranking outlier nodes in subspaces of attributed graphs. In: Workshop on Graph Data Management at IEEE ICDE (2013)

Vertex Similarity - A Basic Framework for Matching Geometric Graphs

Ayser Armiti and Michael Gertz

Institute of Computer Science, Heidelberg University, Germany
{`ayser.armiti,gertz`}@informatik.uni-heidelberg.de

Abstract. Solutions to the graph matching problem play an important role in many application domains, such as chemistry, proteomics, or image processing. Especially in these domains, graphs have geometric properties that describe the positions of the vertices in some 2- or 3-dimensional space. Several exact and approximate approaches have been proposed to address the problem of matching graphs, which is known to be NP-hard in general. For this, most approaches depend on the concept of vertex similarity to iteratively increase the matching quality.

In this paper, we study the vertex similarity problem for geometric graphs. We formally define such a problem and prove that its complexity is NP-hard. For geometric graphs in 2D, we propose an approximate solution with polynomial runtime. For this, we utilize techniques underlying attributed cyclic string matching and customized edit operations that consider spatial properties and labeling information. In our evaluations, we show that our approach outperforms existing vertex similarity approaches in terms of classification accuracy and matching quality.

1 Introduction

Searching for and exploring similar objects is an important task in many application domains, such as in social networks, biology, or pattern recognition. For such domains and many more, graphs are used as a powerful data structure for the representation of objects and object relationships. By this, searching for similar objects turns to be the tasks of finding similar (sub)graphs, which is estimated by a *graph matching* algorithm [9]. Approaches to graph matching search for correspondences between the vertices of two graphs such that the matched vertices have similar labels and connectivity.

The problem of *inexact graph matching*, which is matching graphs in the presence of noise and outliers, has been shown to be NP-hard [25]. As a consequence, most of the approaches to solving the matching problem focus on finding

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

approximate solutions, e.g., [11, 12]. These approaches follow an iterative *continuous optimization* procedure. The objective function used by such optimization procedures depends on the concept of *vertex similarity* [7, 26].

For many graph matching algorithms, vertex similarity means the similarity of the labels assigned to the vertices and edges [25]. But in application domains such as chemistry, proteomics, or image processing, vertex similarity is much more complex and subjective. For such disciplines, it is critical to study the *spatial properties* of vertices, in addition to their labels and connectivity [3].

In this paper, we study the vertex similarity problem for geometric graphs. We build on our previous work [5, 6] and prove that such a problem is NP-hard in general. For geometric graphs in 2D space (as common in many application domains), we propose a novel approach to estimate the similarity between different vertices. Our solution is based on the concept that two vertices are similar when their direct neighboring vertices are similar. For this, we propose to extract a feature for each vertex based on the properties of its neighborhood. We use string edit distance to compute the similarity of two vertices using their features. To realize this, we propose customized edit operations that utilize spatial properties and labeling information. Comprehensive empirical studies using real geometric graph datasets from different application domains are used to demonstrate the accuracy of our proposed approaches compared to related work.

The remainder of the paper is organized as follows. In Section 2, we survey related work. Section 3 discusses the problem settings. In Section 4, we propose our novel approaches to solve vertex similarity for geometric graphs in 2D space. In Section 5, we present experimental results of our proposed approach. Finally, Section 6 summarizes the paper.

2 Related Work

Several graph matching algorithms use the Euclidean distance to estimate the similarity between real-valued labels that are assigned to the vertices and edges [19, 25]. For geometric graphs, the coordinates of the vertices cannot be simply treated as real-valued attributes since they are measured with respect to the particular reference axis frame for each graph. This makes the Euclidean distance incapable of estimating the spatial distance between vertices of two graphs under a geometric transformation. In addition to this, pure structural graph matching approaches cannot be applied to geometric graphs because they do not consider the spatial properties of a graph. Several other approaches have been proposed to solve the vertex similarity problem for non-geometric graphs [27, 28]. However, for geometric graphs, little can be found in literature. We believe that this is a consequence of the complexity of the problem in the case of geometric graphs, which will be discussed in later sections. In the following, we discuss current approaches to estimate the similarity between vertices of geometric graphs and divide them into two classes: global and local approaches.

The global approaches extract a feature for each vertex using the overall graph structure. Algorithms that utilize graph spectra are classified as global

approaches. The main idea is to extract a feature for each vertex based on the values of the Eigenvectors. Such features are then used by the Hungarian algorithm for graph matching [22]. To use the same concept for geometric graphs, the spectra of the *weighted* adjacency or the weighted Laplacian matrices are used. The weight of an entry represents the length of an edge, which is computed using the Euclidean distance between the coordinates of its incident vertices. Then, eigen-decomposition is used to generate a spectral feature for each vertex, which is represented by the values of the Eigenvectors with respect to that vertex. Since graphs with different numbers of vertices create different numbers of Eigenvectors, the spectral features for the vertices are truncated by keeping the values with respect to the most dominant Eigenvectors [26], i.e., the Eigenvectors that correspond to the largest Eigenvalues. Based on this, the distance between two vertices equals the Euclidean distance between their spectral features. A major drawback of the spectral approach is that it cannot handle labeling information. Also, such an approach is sensitive to differences in the number of vertices, the structure of the graph, and the lengths of the edges.

Another global vertex similarity approach is based on the *landmark distance* concept [8]. First, a set of vertices from each graph is selected as landmarks. Then, every vertex from the graph is represented by a feature vector containing the distances to the landmarks. The distance is measured as the length of the shortest path between the vertex and a landmark. Then, the distance between two vertices is computed using the Manhattan distance between their landmark-based features. The basis of such an approach is the selection of landmarks for each graph. Cheong *et al.* [8] propose to use four landmarks as the extreme vertices in the boundaries of the graph, i.e., peripheral vertices. However, such an approach is incapable of matching graphs that differ in the number of vertices.

To overcome the problems of the global approaches, local features are extracted from the neighborhood of each vertex. One of the earliest approaches to estimate the similarity of different vertices is the *histogram-based* approach [10, 13, 21]. A histogram is created from the spatial properties of the neighborhood of each vertex. It stores the pair-wise relationships between the edges that are incident to that vertex, which consists of the ratio of the lengths of the edges in addition to the angle between them. As a result, the local-feature is a 2D histogram of edge lengths and angle values. Based on this, the distance between two vertices is estimated by the distance between their geometric histograms, which is computed by the χ^2 or the Bhattacharyya distances. Unfortunately, histogram approaches face problems in binning and normalization, especially when dealing with real-valued attributes, i.e., the edge length and the angle value.

Notice that the above approaches extract features that are invariant to geometric transformations. Another approach to solve vertex similarity is to use *geometric hashing* based on the coordinates of the vertices [24]. The basis of this approach is to create several local frames for the neighborhood of each vertex, which are defined again by that vertex and its direct neighbors. Then, the coordinates of the vertices in the neighborhood of a vertex are measured with respect to each local frame. After that, hashing is used to speed up the search for the

local frame that best estimates the distance between two vertices. Geometric hashing is efficient in the case of matching vertices that have a homogeneous transformation, i.e., rigid transformation. But, in the case of inexact matching, such an approach fails to estimate the similarity of the vertices.

3 General Problem Setting

In our framework, we consider (non-)planar, labeled, undirected geometric graphs that do not contain self-loops or multi-edges.

Definition 1. (Geometric Graph) A labeled undirected geometric graph $G = (V, E, l, c)$ consists of a finite set of vertices V , a finite set of edges $E \subseteq V \times V$, a labeling function $l : \{V \cup E\} \rightarrow \Sigma$, assigning a label to every vertex and every edge from a label alphabet Σ , and a function $c : V \rightarrow \mathbb{R}^d$, assigning a coordinate in \mathbb{R}^d to every vertex.

Without loss of generality and throughout the rest of this paper, we represent a geometric graph G as $G = (V, E)$. The size $|G|$ of a graph is the number of vertices in G . The degree of a vertex v , denoted $\deg(v)$, is the number of vertices that are directly connected to v . The set of direct neighboring vertices of a vertex v is denoted by $N(v)$.

In our framework, we follow a local-based vertex similarity approach, which has been proved to give good results for general non-geometric graphs [19, 25]. It is based on the concept that two vertices are similar when their neighbors are similar. For our framework, we call the neighborhood of a vertex its *signature*.

Definition 2. (Vertex Signature) Given a vertex v_i in a graph $G = (V, E)$, the vertex signature $S(v_i)$ is a subgraph $G' = (V', E')$ of G such that $V' = \{v_i \cup \{v_j | (v_i, v_j) \in E\}\}$. For each vertex $v_j \in V'$, $v_j \neq v_i$, there exists an edge $(v_i, v_j) \in E'$. v_i is called the **root vertex** of $S(v_i)$.

After defining the meaning of locality, the similarity between two vertices is estimated by computing the similarity of their vertex signatures. A function that quantifies the similarity between two vertex signatures must satisfy geometric transformations, i.e., two vertex signatures are considered spatially identical if there is a geometric transformation (rotation, translation, and scaling) that makes the coordinates of one vertex signature identical to the coordinates of the other [16]. In addition to this, and for many scientific applications, two similar objects are often represented by two non-identical graphs. In pattern recognition applications, acquisition methods often introduce noise in the number of vertices and their locations. Also, the structure and connectivity of vertices often vary between graphs representing similar objects. As a result, two vertex signatures representing similar vertices have differences in the number of neighbors, labeling information, the lengths of the edges, and the distances between the neighboring vertices. This leads to the concept of *inexact vertex similarity*, which will be detailed in the following section.

3.1 Vertex Edit Distance

To compute the inexact similarity between two vertex signatures, we adopt the *edit distance* concept that is been used in matching strings and graphs [20, 23]. It is defined as the minimum amount of changes that is needed to make a string or a graph identical to another. We call the edit distance of two vertex signatures the *vertex edit distance (VED)*. For two vertex signatures $S(v)$ and $S(u)$, the key idea of the VED is to delete some vertices and edges from $S(v)$, re-label some other vertices and edges, change the coordinates of some vertices, and insert some vertices and edges into $S(u)$ such that the two vertex signatures become identical. For this, we adopt three *edit operations*: substitution (re-label), insertion, and deletion. A sequence of edit operations that transfer one vertex signature to be identical to another is called an *edit path*. Obviously, there are many possible edit paths from one vertex signature to another. As a result, the VED is defined as the distance with the minimum cost of all of them:

Definition 3. (Vertex Edit Distance) Let $\phi(S(v), S(u))$ be the set of all geometric transformations between the coordinates of $S(v)$ and $S(u)$, $\Upsilon_{\phi_i}(S(v), S(u))$ be the set of all edit paths between $S(v)$ and $S(u)$ after applying the geometric transformation ϕ_i , then the vertex edit distance is defined as:

$$d(v, u) = \min_{\phi_i \in \phi(S(v), S(u)), p_j \in \Upsilon_{\phi_i}(S(v), S(u))} cost(p_j) \quad (1)$$

where $cost(p_j)$ is the total cost of all edit operations of the path p_j

The cost of an edit path depends on the cost of its edit operations, which we define as the following. The cost of a substitution between two vertices is defined by the Euclidean distance between their coordinates, the distance between their labels, and the substitution costs of their edges. The substitution cost between two edges is defined as the distance between their labels in addition to the distance between their lengths. The cost of vertex insertion or deletion equals to a constant α .

Lemma 1. *The problem of vertex edit distance for geometric graphs in the \mathbb{R}^d space is NP-hard such that $d \geq 2$.*

In the following, and without loss of generality, we give a sketch proof for the above lemma in the case of unlabeled geometric graphs.

Proof Sketch: For two unlabeled geometric graphs, we reduce the problem of inexact point set matching to the problem of vertex edit distance. Let $P = \{p_1, p_2, \dots, p_n\}$ and $Q = \{q_1, q_2, \dots, q_m\}$ be two point sets in \mathbb{R}^d , $d \geq 2$. The two point sets are reduced to two vertex signatures in polynomial time as follows. All points from the point set P become vertices directly connected to a dummy root vertex v_p . The coordinate of v_p is computed as the center of the point set P . In the same way, the points from Q create a vertex signature with a dummy root vertex v_q . The optimal match between P and Q is the optimal mapping of the neighbors of v_p and the neighbors v_q . Non-matched points (vertices) represent

the insertion and deletion operations. A substitution operation is indicated by a correspondence from one point to another.

The problem of inexact point set matching in the \mathbb{R}^d space is proved to be NP-hard [4] where $d \geq 2$. As result, the problem of computing the optimal solution of vertex edit distance for geometric graphs in \mathbb{R}^d space is also NP-hard. \square

4 Vertex Similarity for 2D Geometric Graphs

In this section, we propose an approximate solution to the VED problem for geometric graphs in 2D space, which can be computed in $O(mn \log n)$, such that n and m are the numbers of edges for two vertex signatures. For this, we require that the edges of a vertex signature are sorted in counter-clockwise order around the root vertex. As a result, the feature that is extracted from each vertex signature is a cyclic string that is defined as follows:

Definition 4. (Spatial Feature) Given a geometric graph $G = (V, E, l, c)$, the spatial feature F_v for the vertex signature $S(v)$ is a **cyclic string** $F_v = [f_1, f_2, \dots, f_n]$, $n = \deg(v)$, such that each token f_i is defined as:

$$f_i := (|e_i|, \angle_{e_i e_{i-1}}, l(e_i), l(v_i))$$

where $|e_i|$ denotes the length of the edge e_i , $\angle_{e_i e_{i-1}}$ denotes the angle between the edges e_i and e_{i-1} in counter-clockwise order, $l(e_i)$ is the label of edge e_i , and $l(v_i)$ is the label of the neighboring vertex incident to edge e_i .

The feature is created by selecting an edge from the vertex signature and going over the rest of the edges in a counter-clockwise order. For a vertex signature of n edges, there will be n different ways to represent its feature. However, all of them are considered equivalent with a cyclic shift from one to another.

Once the spatial features are represented as cyclic strings, the VED between two vertex signatures is estimated by the *cyclic string edit distance* (CS) [17], which is a natural extension to the string edit distance. A naive approach to solve it runs in $O(nm^2)$, where n and m are numbers of edges for two vertex signatures. This is done by applying the algorithm by Wagner and Fisher [23] to the first spatial feature and all cyclic shifts of the second one. Maes in [17] proposed a faster solution to the cyclic string edit distance that runs in time $O(nm \log m)$. So, the cyclic string edit distance gives an approximate solution to the VED problem with a runtime complexity of $O(nm \log m)$.

To utilize the CS approach, we define three edge edit operations: substitution, insertion, and deletion. We propose edit operations that combine spatial attributes and labeling information. In the following we discuss two sets of edit operations. The first one computes the edit operations based on the absolute values of the edge length and the angel value. The second one uses a polar distance based on the lengths of two edges and the angle between them.

Edit operations using the Manhattan distance

Given two vertex signatures $S(v)$ and $S(u)$ such that $n = |S(v)|$ and $m = |S(u)|$, let edge $e_i \in S(v)$, edge $e_j \in S(u)$, $f_i = (|e_i|, \angle e_i e_{i-1}, l(e_i), l(v_i))$, $f_j = (|e_j|, \angle e_j e_{j-1}, l(e_j), l(u_j))$, then, the substitution $\gamma(f_i \rightarrow f_j)$ is defined as:

$$\gamma(f_i \rightarrow f_j) := d_L(f_i, f_j) + d_S(f_i, f_j) \quad (2)$$

In the case of labeled graphs, the function $d_L(f_i, f_j)$ computes the distance between the label of edge e_i and the label of e_j in addition to the distance between the labels of the vertices that are incident to them, i.e., v_i and u_j . The function $d_S(f_i, f_j)$ calculates the spatial distance based on the angle and the edge length. For an edge e , let θ_e and l_e denote the angle and edge length, as defined earlier in Definition 4. The function d_S is formally defined as follows:

$$d_S(f_i, f_j) := \frac{|\theta_{e_i} - \theta_{e_j}|}{2\pi} + \left| \frac{l_{e_i}}{\sum_{k=1}^n l_{e_k}} - \frac{l_{e_j}}{\sum_{k=1}^m l_{e_k}} \right| \quad (3)$$

The angles and edge lengths at a vertex signature are normalized, as can be seen by the denominators used in the above equation. An angle is normalized by 2π since the sum of angles at a local signature sums up to this value. Also, an edge length is normalized by the sum of edge lengths at a local signature. For example, for a local signature $S(v)$, the edge length normalization factor is $\frac{l_{e_i}}{\sum_{k=1}^n l_{e_k}}$, where n is the number of edges connected to v .

In the following, we define the insertion and deletion operations. Let λ represent the null (non-existent) edge, then the insertion $\gamma(\lambda \rightarrow f_i)$ and deletion $\gamma(f_i \rightarrow \lambda)$ with respect to f_i are defined as follows:

$$\gamma(\lambda \rightarrow f_i) = \gamma(f_i \rightarrow \lambda) := c(f_i) + \left(\frac{\theta_{e_i}}{2\pi} + \frac{l_{e_i}}{\sum_{k=1}^n l_{e_k}} \right) \quad (4)$$

The cost of edge insertion or deletion is computed based on the angle value, edge length, and labeling information. For labeled graphs, the function c defines the cost of inserting or deleting the label assigned to that edge in addition to the label assigned to its incident vertex.

For unlabeled graphs, the cost of an edit operation lies in the range $[0,2]$. This is because each of the angle value and edge length is normalized to the range $[0,1]$. For labeled graphs, the range increases depending on the range of the function d_L for the substitution operation and c for the insertion and deletion.

Edit operations using polar coordinate

The second set of edit operations shares many similarities with the previously defined edit operations. However, the spatial distance between two vertex signatures is computed based on the polar distance between the neighboring vertices

of two vertex signatures. Given two vertex signature $S(v)$ and $S(u)$, let edge $e_i \in S(v)$ and edge $e_j \in S(u)$. The substitution cost $\gamma(f_i \rightarrow f_j)$ is defined as:

$$\gamma(f_i \rightarrow f_j) = d_L(f_i, f_j) + d_S(f_i, f_j) \quad (5)$$

In the case of labeled graphs, the function $d_L(f_i, f_j)$ computes the distance between the label of edge e_i and the label of e_j . It also computes the distance between the label of the neighboring vertex connected to e_i to the label of the one connected to e_j . The function $d_S(f_i, f_j)$ calculates the spatial distance based on the angles and the lengths of the edges. For an edge e , let θ_e denote the angle between e and the previous edge in a counter-clockwise order, and let l_e denote the edge length. The function d_S is defined as:

$$d_S(f_i, f_j) = \sqrt{l_{e_i}^2 + l_{e_j}^2 - 2 l_{e_i} l_{e_j} \cos(|\theta_{e_i} - \theta_{e_j}|)} \quad (6)$$

The substitution cost is defined as the distance needed for the neighboring vertex of edge e_i to align with the neighboring vertex of e_j . This can be seen as the polar distance between them such as the polar axis for each vertex is the edge that precedes it in the counter-clockwise order. Analogously, we define the insertion and deletion operations. Let λ represent the null (non-existent) edge. Then, the insertion $\gamma(\lambda \rightarrow f_i)$ and deletion $\gamma(f_i \rightarrow \lambda)$ with respect to f_i are defined as:

$$\gamma(\lambda \rightarrow f_i) = \gamma(f_i \rightarrow \lambda) = c(f_i) + l_{e_i} \quad (7)$$

The cost of edge insertion or deletion is computed based on the edge length (l_{e_i}). For labeled graphs, the function c defines the cost of inserting or deleting the label assigned to the edge e_i in addition to the label assigned to the neighboring vertex connected to e_i .

5 Evaluation

In this section, our proposed approach to the vertex similarity problem and its usage for graph matching is empirically evaluated. We use three different data sets: 1) Chinese characters [1], 2) the COIL-100 image data set [18], and 3) the CMU house and hotel image data sets [2]. Besides coming from different application domains, our data sets vary in many aspects such as the size of the data set, the number of classes (in case geometric graphs have been assigned to classes), as well as the number of vertices and edges.

We compare our algorithms (**CSv1**), which uses the first set of edit operations, and (**CSv2**), which uses the second set of edit operations, with three other approaches: a graph spectral approach (**SP**) [22, 26], a geometric histogram of the pair-wise relations between the edges in the neighborhood of a vertex (**GH**) [14, 21], and a unary vertex distance function based on only the coordinates of the vertices (**CO**) such as neither global nor local structural information is used. We test such an approach to evaluate the effect of using the coordinates of the vertices on their similarities.

To evaluate the different approaches, we embed them in a unified graph matching algorithm. It consists of two steps. First, a vertex-to-vertex distance matrix is created using any of the previous approaches. Second, the Hungarian algorithm [15] is used to select the best match between the two graphs. Since all the approaches use the Hungarian algorithm for graph matching, the differences in the matching results are affected only by the approach that is used to estimate the similarity between two vertices.

To evaluate the performance of a vertex similarity approach, we use two criteria. The first one is the effect of vertex similarity on a graph similarity metric. This is evaluated by embedding the graph matching algorithm in a classification task. The higher the classification accuracy the better the vertex similarity approach. To create a graph distance metric, we follow a graph edit distance approach. This means that the distance between two graphs consists of the cost of the match between them, i.e., the substitution cost, in addition to the cost of inserting the unmatched vertices. The second criterion is the selectivity power, which means that a vertex similarity approach reflects the similarity notion of an application domain. This is measured by the quality of the match computed by the graph matching algorithm.

5.1 Graph Similarity and Classification

In this section, we evaluate the relation between different vertex similarity approaches and graph similarity in general. To measure this, we test the different approaches in a graph classification task. In our experiments, we used the first nearest neighbor classifier (1-NN) based on the similarities of the graphs. For this experiment, we use the COIL-100 data set [18], which consists of images of 100 different objects taken at different degrees. A geometric graph is then extracted from each image. From 3900 graphs, we select 2900 for training, 29 graphs for each object. For testing, we select 1000 graphs, 10 graphs for each object. We also use the Chinese data set [1], which contains a total of 9384 characters that belong to 6 different fonts, i.e., 1564 characters from each font. A test data set of 1564 graphs is extracted from the Dotum Korean font. The remaining five fonts build a training data set of 7820 graphs. Ideally, for a query character, its most similar character from the train data set should have the same Unicode.

Figure 1(a) shows the classification accuracies for the COIL-100 data set for the different approaches. The lowest classification accuracy is for the **SP** approach. This is because spectral approaches are sensitive to the changes in the number of vertices in addition to their spatial properties. We conclude that a local-based vertex similarity approach is better than a global-based one. The best classification accuracy is for the **CSv2+CS** approach, followed by the **CSv2** approach. Notice that the use of invariant spatial features by our approaches (**CSv1** and **CSv2**) gives better results than using the coordinates of the vertices (**CO**). However, combining both of them gives the best result.

The classification accuracy for the Chinese data set is shown in Figure 1(b). Also, for this data set, the best results is for the **CO+CSv2** approach. On the

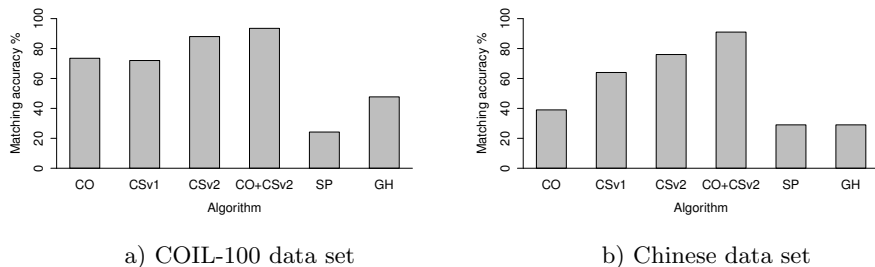


Fig. 1: Classification accuracy for different vertex similarity approaches.

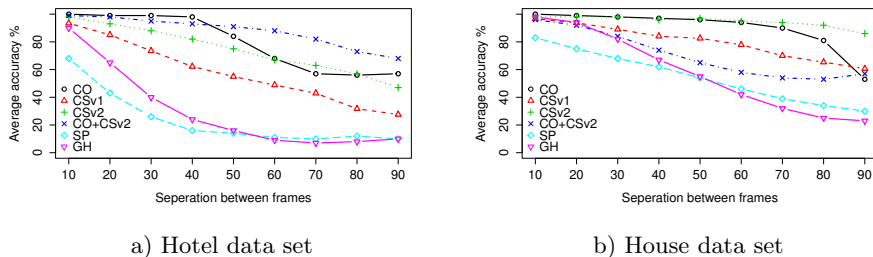
other hand, the **CSv1** and **CSv2** approaches are much better than **CO** alone. The lowest classification accuracy is for **SP** and **GH**.

From these two data sets we conclude that using invariant spatial features is better than using only the coordinates of the vertices. However, still the coordinates of the vertices can be used to give good graph matching results for many applications. Also, using the second set of edit operations, i.e., **CSv2**, gives better results than the first set, i.e., **CSv1**. This is justified since **CSv1** gives the same weight for the differences in the angle value and the edge length.

5.2 Graph Matching

In this section, we evaluate the quality of the match computed by the graph matching algorithm. Higher matching quality indicates higher selectivity power for a vertex similarity approach. We use the matching accuracy to estimate the quality of a match. It is defined as the number of correct matches, computed by a matching algorithm, over the actual total number of correct matches. For this test, we use the CMU hotel and house data sets. They contain images for a toy house and hotel, subjected to rotation in 3D. For each data set, we match all images spaced at 10, 20, 30, 40, 50, 60, 70, 80, and 90 in the rotation sequence, and compute the average matching accuracy.

From Figures 2(a) and 2(b), one can see that for all the approaches, the matching accuracy decreases when the distance in the rotation sequence between the images increases. This is a consequence of the increase in the structural differences between the geometric graphs. The lowest matching accuracy is for the **SP** approach, which is sensitive to the changes in the structure of the graphs. The best matching accuracy is for the **CSv2** and **CO**. However, **CO+CSv2** is not always better than using each of the single approaches alone. Also, the matching accuracy of **CSv2** is better than the one of **CSv1**. This means that using the polar distance gives better results than using just the absolute values of the length of the edge and the angle value.



a) Hotel data set

b) House data set

Fig. 2: Matching quality for the CMU hotel/house data sets.

6 Conclusions

In this paper, we discussed the problem of vertex similarity for geometric graphs. Our focus is on local-based vertex similarity approaches, which use the properties of the neighborhoods of the vertices to estimate their similarities. One of the main results that we introduced is the sketch proof that the problem of vertex similarity for geometric graphs is NP-hard in general. On the other side, we proposed an algorithm to approximate the similarity between vertices for geometric graphs in 2D space. Our solution utilizes the property that the direct neighbors of a vertex has a total order, which is a consequence of the embedding of the neighboring vertices in 2D space. To find the similarity between two vertices, first, a spatial feature is extracted, which is a cyclic string of the lengths of the edges in addition to the angles between them. After that, the cyclic string edit distance is used to estimate the similarity of different vertices. For this, we proposed edit operations that utilize spatial properties and labeling information. We demonstrated the accuracy of our approach using different real-world data sets from image processing and character recognition. We also showed that our approach compares favorably to existing vertex similarity techniques.

References

1. CJK Fonts: Chinese, Japanese, and Korean Fonts. <http://bookr-mod.googlecode.com/files/cjk-fonts-1.zip>. Accessed: 01/12/2011.
2. CMU house and hotel data sets. <http://vasc.ri.cmu.edu/idb/html/motion>. Accessed: 16/02/2012.
3. C. Aggarwal and H. Wang. *Managing and mining graph data*. Springer, 2010.
4. T. Akutsu, K. Kanaya, A. Ohya, and A. Fujiyama. Point matching under non-uniform distortions. *Discrete Applied Mathematics*, 127(1):5–21, 2003.
5. A. Armiti and M. Gertz. Efficient Geometric Graph Matching Using Vertex Embedding. In *SIGSPATIAL*, pages 234–243, 2013.
6. A. Armiti and M. Gertz. Geometric Graph Matching and Similarity: A Probabilistic Approach. In *SSDBM*, pages 27:1–27:12, 2014.
7. T. Caetano, J. McAuley, L. Cheng, Q. Le, and A. Smola. Learning graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(6):1048–1058, 2009.

8. O. Cheong, J. Gudmundsson, H. Kim, D. Schymura, and F. Stehn. Measuring the Similarity of Geometric Graphs. *Experimental Algorithms*, pages 101–112, 2009.
9. D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty Years Of Graph Matching In Pattern Recognition. *IJPRAI*, 18(3):265–298, 2004.
10. X. Gao, B. Xiao, D. Tao, and X. Li. Image categorization: Graph edit distance+edge direction histogram. *Pattern Recognition*, 41:3179–3191, 2008.
11. X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Anal. Appl.*, 13(1):113–129, 2010.
12. S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(4):377–388, 1996.
13. B. Huet and E. Hancock. Inexact graph retrieval. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 40–44, 1999.
14. B. Huet and E. R. Hancock. Relational Object Recognition from Large Structural Libraries. *Pattern Recognition*, 35:1895–1915, 2002.
15. H. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
16. M. Kuramochi and G. Karypis. Discovering Frequent Geometric Subgraphs. In *ICDM*, pages 258–265, 2002.
17. M. Maes. On a cyclic string-to-string correction problem. *Information Processing Letters*, 35(2):73–78, 1990.
18. S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). Technical report, Feb 1996.
19. K. Riesen and H. Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, 27(7):950–959, 2009.
20. A. Sanfeliu and K.-S. Fu. A Distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Syst., Man, Cybern., Syst.*, 13(3):353–362, 1983.
21. N. Thacker, P. Riocreux, and R. Yates. Assessing the completeness properties of pairwise geometric histograms. *Image and Vision Computing*, 13(5):423–429, 1995.
22. S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):695–703, 1988.
23. R. Wagner and M. Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173, 1974.
24. X. Wang, D. Shasha, B. Shapiro, I. Rigoutsos, and K. Zhang. Finding Patterns in Three-Dimensional Graphs: Algorithms and Applications to Scientific Data Mining. *IEEE Trans. on Knowl. and Data Eng.*, 14(4):731–749, July 2002.
25. Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou. Comparing Stars: On Approximating Graph Edit Distance. *PVLDB*, 2(1):25–36, 2009.
26. Y. Zhu, L. Qin, J. X. Yu, Y. Ke, and X. Lin. High efficiency and quality: large graphs matching. In *CIKM*, pages 1755–1764, 2011.
27. G. Wang, B. Wang, X. Yang, and G. Yu. Efficiently Indexing Large Sparse Graphs for Similarity Search. *IEEE Trans. Knowl. Data Eng.*, 24(3):440–451, 2012.
28. C. Xiao, X. Lin, X. Zhao, and W. Wang. Efficient Graph Similarity Joins with Edit Distance Constraints. In *ICDE*, pages 834–845, 2012.

Towards an Integer Approximation of Undirected Graphical Models

Nico Piatkowski and Katharina Morik

Artificial Intelligence Group, TU Dortmund University, 44227 Dortmund, Germany
{nico.piatkowski, katharina.morik}@cs.tu-dortmund.de
<http://www-ai.cs.tu-dortmund.de>

Data analytics for streaming sensor data brings challenges for the resource efficiency of algorithms in terms of execution time and the energy consumption simultaneously. Fortunately, optimizations which reduce the number of CPU cycles also reduce energy consumption. When reviewing the specifications of processing units, one finds that integer arithmetic is usually cheaper in terms of instruction latency, i.e. it needs a small number of clock cycles until the result of an arithmetic instruction is ready. This motivates the reduction of CPU cycles in which code is executed when designing a new, resource-aware learning algorithm. Beside clock cycle reduction, limited memory usage is also an important factor for small devices.

Outsourcing parts of data analysis from data centers to ubiquitous devices that actually measure data would reduce the communication costs and thus energy consumption. If, for instance, a mobile medical device or smartphone can build a probabilistic model of the usage behavior of its user, energy models can be made more accurate and power management can be more efficient. The biggest hurdle in doing this, are the heavily restricted computational capabilities of very small devices—some do not even have a floating point processor. Consequently, computationally simple machine learning approaches have to be considered. Low complexity of machine learning models is usually achieved by independence assumptions among features or labels. In contrast, the joint prediction of multiple dependent variables based on multiple observed inputs is an ubiquitous subtask in real world problems from various domains. Probabilistic graphical models are well suited for such tasks, but they suffer from the high complexity of probabilistic inference.

In the extended abstract at hand, we show how to write the joint probability mass function of undirected graphical models as rational number, if the parameters are integers. More details on the integer parametrization of undirected graphical models can be found in [1]. Inference algorithms and a new optimization scheme are proposed, that allow the learning of integer parameters without the need for any floating point computation. This opens up the opportunity of running machine learning tasks on very small, resource-constrained devices. To be more precise, based only on integers, it is possible to compute approximations to marginal probabilities, to maximum-a-posteriori (MAP) assignments and maximum likelihood estimate either via an approximate closed form solution or an integer variant of the stochastic gradient descent (SGD) algorithm. It turns

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. BEECKS (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

out that the integer approximations use less memory and deliver a reasonable quality while being around twice as fast as their floating point counterparts. To the best of our knowledge, there is nothing like an integer undirected model so far.

Many approximate approaches to probabilistic inference based on belief propagation were proposed in the last decade. Unfortunately, most of these methods are by no means suited for embedded or resource constrained environments. In contrast to these approaches, the model class that is proposed in the paper at hand has the same asymptotic complexity as the vanilla inference methods, but it uses cheaper operations. Inspired by work from the signal processing community, the underlying model class is restricted to the integers, which results in a reduced runtime and energy savings, while keeping a good performance. This new approach should not be confused with models that are designed for integer state spaces, in which case the state space \mathcal{X} is a subset of the natural numbers or, more generally, is a metric space. Here, the state space may be an arbitrary discrete space without any additional constraints.

In their book on graphical models, Wainwright and Jordan stated that "It is important to understand that for a general undirected graph the compatibility functions ψ_C need not have any obvious or direct relation to marginal or conditional distributions defined over the graph cliques. This property should be contrasted with the directed factorization, where the factors correspond to conditional probabilities over the child-parent sets." This raises hope that we might find meaningful probabilistic models, even when we restrict the model parameters to be integers. For excluding every floating point computation, the identification of integer parameters is not enough. That is, the computations for training and prediction have to be based on integer arithmetic.

The first step towards integer models is directly related to the above statement. Strictly speaking, the parameter domain Ω is restricted to the set of integers \mathbb{N} and a new potential function is defined as

$$\bar{\psi}_C(x_C) := 2^{(\theta_C, \phi_C(x))} = \exp(\ln(2)\langle \theta_C, \phi_C(x) \rangle).$$

Considering parameters $\theta \in \mathbb{R}^d$ of a model that has potential function $\psi_C(x_C)$, it is easy to see that replacing $\psi_C(x_C)$ with $\bar{\psi}_C(x_C)$ does not alter the marginal probabilities as long as the parameters are scaled by $1/\ln 2$. By this, it is possible to convert integer parameters that are estimated with $\bar{\psi}_C(x_C)$ to $\psi_C(x_C)$ (and vice versa), without altering the resulting probabilities. Notice that $\bar{\psi}_C(x_C)$ can be computed by logical bit shift operations which consume less clock cycles than the corresponding transcendental functions required to compute $\psi_C(x_C)$.

Acknowledgements This work has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Data Analysis", project A1.

References

1. Piatkowski, N., Sangkyun, L., Morik, K.: The integer approximation of undirected graphical models. In: De Marsico, M., Tabbone, A., Fred, A. (eds.) 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM). SciTePress (2014)

A Purely Geometric Approach to Non-Negative Matrix Factorization

Christian Bauckhage

B-IT, University of Bonn, Bonn, Germany
Fraunhofer IAIS, Sankt Augustin, Germany
<http://mmprec.iais.fraunhofer.de/bauckhage.html>

Abstract. We analyze the geometry behind the problem of non-negative matrix factorization (NMF) and devise yet another NMF algorithm. In contrast to the vast majority of algorithms discussed in the literature, our approach does not involve any form of constrained gradient descent or alternating least squares procedures but is of purely geometric nature. In other words, it does not require advanced mathematical software for constrained optimization but solely relies on geometric operations such as scaling, projections, or volume computations.

Keywords: latent factor models, data analysis.

1 Introduction

Non-negative matrix factorization (NMF) has become a popular tool of the trade in areas such as data mining, pattern recognition, or information retrieval. Ever since Paatero and Tapper [14] and later Lee and Seung [11] published seminal papers on NMF and its possible applications, the topic has attracted considerable research that produced a vast literature. Related work can be distinguished into two main categories: either reports on practical applications in a wide range of disciplines or theoretical derivations of efficient algorithms for NMF.

The work reported here belongs to the latter category. However, while our technique scales to very large data sets, our focus is not primarily on efficiency. Rather, our main goal is to expose a new point of view on NMF and to show that it can be approached from an angle that, to the best of our knowledge, has not been widely considered yet.

In order for this paper to be accessible to a wide audience, we first review the NMF problem, its practical appeal, established algorithms for its computation, and known facts about its complexity. Readers familiar with matrix factorization for data analysis might want to skip this introductory exposition.

Then, we discuss NMF from a geometric point of view and devise an NMF algorithm that does not involve gradient descent or alternating least squares

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

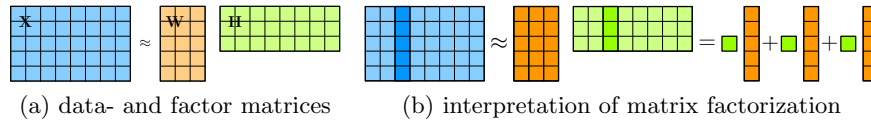


Fig. 1: Visualization of the idea of matrix factorization and its interpretation as representing data vectors in terms of linear combinations of a few latent vectors.

schemes. Rather, our approach is based on strikingly simple geometric properties that were already noted by Donoho and Stodden [7] and Chu and Lin [4] but, again to our best knowledge, have not yet been fully exploited to design NMF algorithms. In short, we present an approach towards computing NMF that does not explicitly solve constrained optimization problems but only relies on rather simple operations.

The three major benefits we see in this are: (a) our approach allows users to compute NMF even if they do not have access to specialized software for numerical optimization; (b) it allows for parallelization and therefore naturally scales to BIG DATA settings; (c) last but not least our approach hardly requires prior knowledge as to optimization theory and convex analysis and therefore provides an alternative, possibly more intuitive avenue towards teaching these materials to students.

2 Non-Negative Matrix Factorization

Applications of NMF naturally arise whenever we are dealing with the analysis of data that reflect counts, ranks, or physical measurements such as weights, heights, or circumferences which are non-negative by definition. In situations like these, the basic approach is as follows: Assume a set $\{\mathbf{x}_j\}_{j=1}^n$ of n non-negative data vectors $\mathbf{x}_j \in \mathbb{R}^m$ and gather them in an $m \times n$ data matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$.

Given such a non-negative data matrix, we write $\mathbf{X} \succeq \mathbf{0}$ to express that its entries $x_{ij} \geq 0$ for all i and j . The problem of computing a non-negative factorization of \mathbf{X} then consists of two basic tasks:

1. Fix an integer $k \ll \text{rank}(\mathbf{X}) \leq \min(m, n)$.
2. Determine two non-negative factor matrices \mathbf{W} and \mathbf{H} where \mathbf{W} is of size $m \times k$, \mathbf{H} is of size $k \times n$, and their product approximates \mathbf{X} . In other words, determine two non-negative, rank-reduced matrices \mathbf{W} and \mathbf{H} such that $\mathbf{X} \approx \mathbf{W}\mathbf{H}$. Mathematically, this can be cast as a constrained optimization problem

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{H}} E(k) &= \left\| \mathbf{X} - \mathbf{W}\mathbf{H} \right\|^2 & (1) \\ \text{subject to } & \mathbf{W} \succeq \mathbf{0} \\ & \mathbf{H} \succeq \mathbf{0} \end{aligned}$$

where $\|\cdot\|$ denotes the matrix Frobenius norm. Note that instead of minimizing a matrix norm to determine suitable factor matrices, we could also attempt to minimize (more) general divergence measures $D(\mathbf{X}||\mathbf{WH})$. Yet, w.r.t. actual computations this would not make much of a difference so that we confine our discussion to the more traditional norm-based approaches.

Now, assume, for the time being, that \mathbf{W} and \mathbf{H} have been computed already. Once they are available, it is easy to see that each column \mathbf{x}_j of \mathbf{X} can be reconstructed as

$$\mathbf{x}_j \approx \hat{\mathbf{x}}_j = \mathbf{W}\mathbf{h}_j = \sum_{i=1}^k \mathbf{w}_i h_{ij} \quad (2)$$

where \mathbf{h}_j denotes column j of \mathbf{H} and \mathbf{w}_i refers to the i th column of \mathbf{W} . Next, we briefly point out general benefits and applications of this representation of the given data.

2.1 General Use and Applications

Looking at (2), the following properties and corresponding applications of data matrix factorization quickly become apparent:

Latent component detection: Each data vector \mathbf{x}_j is approximated in terms of a linear combination of the k column vectors \mathbf{w}_i of matrix \mathbf{W} . Thus, in a slight abuse of terminology, \mathbf{W} is typically referred to as the matrix of “basis vectors”. Since each \mathbf{w}_i is an m -dimensional vector, any linear combination of the \mathbf{w}_i produces another m -dimensional vector. Yet, since the number k of basis vectors in \mathbf{W} is less than the dimension m of the embedding space, we see that the reconstructed data vectors $\hat{\mathbf{x}}_j$ reside in a k -dimensional subspace spanned by the \mathbf{w}_i . Hence, solving (1) for \mathbf{W} provides k latent factors \mathbf{w}_i each of which characterizes a different distinct aspect or tendency within the given data.

Dimensionality reduction: There is a one-to-one correspondence between the data vectors \mathbf{x}_j in \mathbf{X} and the columns \mathbf{h}_j of \mathbf{H} and we note that the entries h_{ij} of vector \mathbf{h}_j assume the role of coefficients in (2). Accordingly, the factor matrix \mathbf{H} is typically referred to as the coefficient matrix. We also note that while \mathbf{x}_j is an m -dimensional vector, the corresponding coefficient vector \mathbf{h}_j is only k -dimensional. In this sense, NMF implicitly maps m -dimensional data to k -dimensional representations.

Data compression: Storage requirements for the original data matrix \mathbf{X} are of the order of $O(mn)$. For the approximation $\mathbf{X} \approx \mathbf{WH}$, however, we would only have to store an $m \times k$ and a $k \times n$ matrix which would need space of the order of $O(k(m+n))$. Since typically $k \ll mn/(m+n)$ this allows for considerable savings.

All these practical benefits also apply to related methods such as the singular value decomposition (SVD) or independent component analysis to name but a few. In this sense, NMF is not at all special. However, while methods such as

the SVD are well appreciated for their statistical guarantees as to the quality of the resulting low-rank data representations, they are not necessarily faithful to the nature of the data. In other words, basis vectors resulting from other methods are usually not non-negative and therefore will explain non-negative data in terms of latent factors that may not have physical counterparts. It is for reasons like these that NMF has become popular.

2.2 General Properties and Characteristics

Looking at (1), we recognize a problem that is convex in either \mathbf{W} or \mathbf{H} but not in \mathbf{W} and \mathbf{H} jointly. In other words, NMF suffers from the fact that the objective function $E(k)$ usually has numerous local minima. Although a unique global minimum provably exists [20], there are no algorithms known today that were guaranteed to find it within reasonable time.

Indeed, (1) is an instance of a constrained Euclidean sum-of-squares problem and thus NP hard [1, 21]. Consequently, known NMF algorithms typically approach the problem using iterative procedures. Usually, both factor matrices are randomly initialized to non-negative values and then refined by means of alternating least squares or gradient descent schemes.

The former approach goes back to [14] and works like this: first, fixate \mathbf{W} and solve (1) for \mathbf{H} using non-negative least squares solvers. Then, given the updated coefficient matrix, solve (1) for \mathbf{W} and repeat both steps until convergence.

The latter idea was first considered in [11] and makes use of the fact that

$$E(k) = \|\mathbf{X} - \mathbf{W}\mathbf{H}\|^2 = \text{tr} \left[\mathbf{X}^T \mathbf{X} - 2\mathbf{X}^T \mathbf{W}\mathbf{H} + \mathbf{H}^T \mathbf{W}^T \mathbf{W}\mathbf{H} \right] \quad (3)$$

so that

$$\frac{\partial E}{\partial \mathbf{W}} = 2 \left[\mathbf{W}\mathbf{H}\mathbf{H}^T - \mathbf{X}\mathbf{H}^T \right] \quad \text{and} \quad \frac{\partial E}{\partial \mathbf{H}} = 2 \left[\mathbf{W}^T \mathbf{W}\mathbf{H} - \mathbf{W}^T \mathbf{X} \right]. \quad (4)$$

Updates for both factor matrices can thus be computed in another alternating fashion using

$$\mathbf{W} \leftarrow \mathbf{W} - \eta_W \frac{\partial E}{\partial \mathbf{W}} \quad \text{and} \quad \mathbf{H} \leftarrow \mathbf{H} - \eta_H \frac{\partial E}{\partial \mathbf{H}} \quad (5)$$

where η_W and η_H are step sizes which, if chosen cleverly, guarantee that any intermediate solutions for \mathbf{W} and \mathbf{H} remain non-negative [11].

As of this writing, numerous variations of these two ideas have been proposed which, for instance, involve projected- or sub-gradient methods [12, 15]. Further details and theoretical properties regarding such approaches can be found in [5].

We conclude our discussion of the properties of NMF by noting that solutions found through iteratively solving (1) critically depend on how \mathbf{W} and \mathbf{H} are initialized [3]. In fact, solutions found from considering (1) are usually not unique [10]. This can easily be seen as follows: Let $\mathbf{X} \approx \mathbf{W}\mathbf{H}$ and let \mathbf{D} be a scaling matrix, then $\mathbf{X} \approx \mathbf{W}\mathbf{D}\mathbf{D}^{-1}\mathbf{H} = \tilde{\mathbf{W}}\tilde{\mathbf{H}}$ which is to say that NMF “suffers” from indeterminate scales. Our discussion below will clarify this claim.

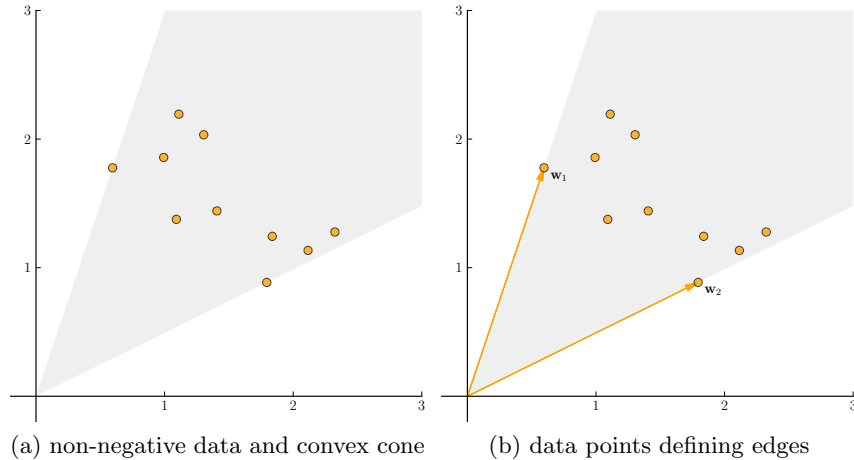


Fig. 2: Illustration of the fact that non-negative data reside in a polyhedral cone.

3 The Geometry of NFM

In the context of NMF, Donoho and Stodden [7] were the first to point to the fact that any set of non-negative vectors of arbitrary dimensionality resides within a convex cone which itself is embedded in the positive orthant of the corresponding vector space (see Fig. 2(a) for 2-dimensional example).

Since practical applications usually deal with finitely many data points, we note that any finite set of non-negative vectors $\mathbf{x}_j \in \mathbb{R}^m$ lies indeed within a *convex polyhedral cone*, i.e. within the convex hull of a set of halflines whose directions are defined by some of the given vectors. This is illustrated in Fig. 2(b) where the two vectors \mathbf{w}_1 and \mathbf{w}_2 that define the edges of the cone coincide with two of the data points.

These observations hint at NMF approaches where the estimation of \mathbf{W} can be *decoupled* from the computation of the coefficient matrix \mathbf{H} . If it was possible to identify those $p \leq n$ data points in \mathbf{X} that define the edges of the enclosing polyhedral cone, they could either be used to perfectly reconstruct the data or we could select $k \leq p$ of them that would allow for reasonably good approximations. These prototypes would form \mathbf{W} and the coefficient matrix \mathbf{H} could be computed subsequently. Moreover, as shown in [18], such a decoupling would enable parallel NMF: Once \mathbf{W} had been determined, the data matrix \mathbf{X} could be partitioned into r blocks $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_r]$ where $\mathbf{X}_i \in \mathbb{R}^{m \times n/r}$. For each block, we could then solve (1) for the corresponding \mathbf{H}_i which might be done on r cores simultaneously.

While the work in [18] approached the selection of suitable prototypes \mathbf{w}_i from \mathbf{X} by means of random projections, Chu and Lin [4] pointed out another interesting geometric property of NMF which we illustrate in Fig. 3. It shows that the cone that encloses the data in \mathbf{X} remains invariant under certain simple

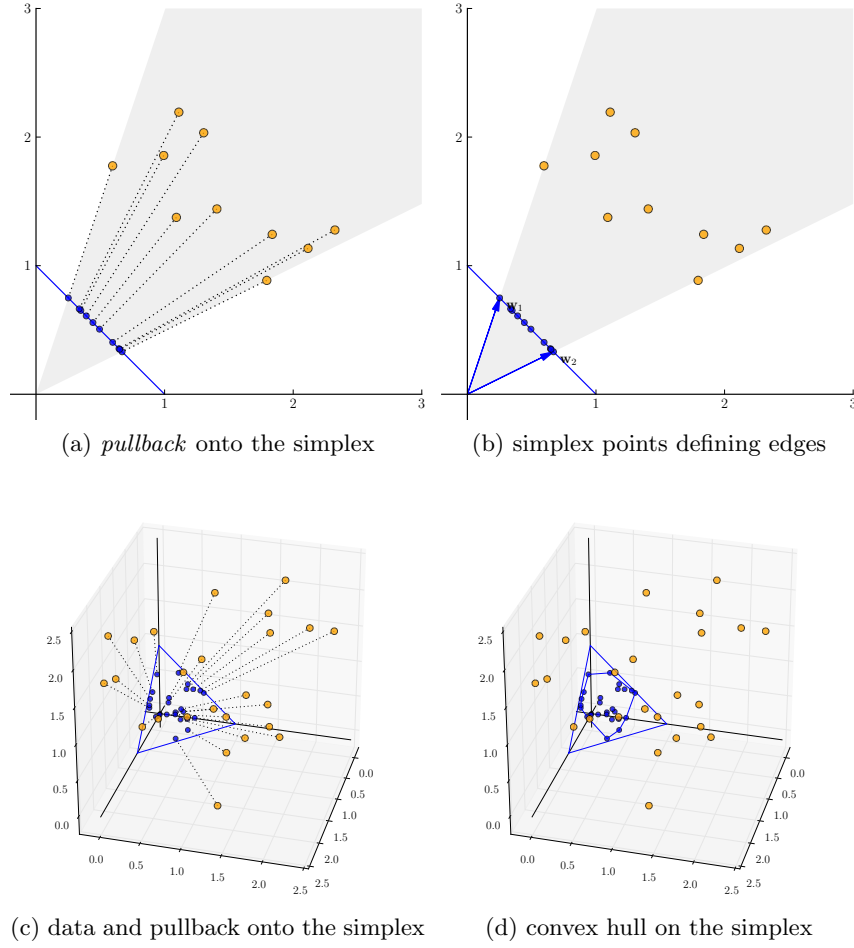


Fig. 3: Pulling non-negative data back onto the standard simplex leaves the geometry of the enclosing polyhedral cone intact.

transformations. In particular, the so called *pullback*

$$\mathbf{y}_j = \frac{\mathbf{x}_j}{\sum_i x_{ij}} \quad (6)$$

which maps each data point $\mathbf{x}_j \in \mathbb{R}^m$ to a point \mathbf{y}_j in the standard simplex Δ^{m-1} does not affect the halfplanes that define the cone.

Moreover, data points \mathbf{x}_j on the edges of the cone in \mathbb{R}^m will be mapped to vertices of the convex hull of the $\mathbf{y}_j \in \Delta^{m-1}$ (see Fig. 3). This observation is crucial, because it suggests that:

The problem of estimating a suitable basis matrix \mathbf{W} for NMF can be cast as a problem of *archetypal analysis* on the simplex.

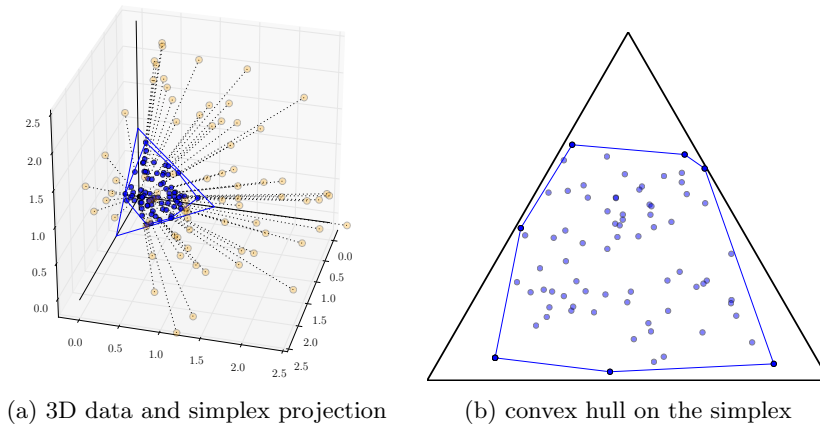


Fig. 4: Pullback to the simplex and data convex hull on the simplex.

Archetypal analysis is a latent factor model due to Cutler and Breiman [6] who proposed to represent data in terms of convex combinations of extremes, that is, in terms of convex combinations of points on the convex hull of a given set of data. Recently, it spawned considerable research because it was recognized that it allows for a decoupled and thus efficient computation of basis elements and coefficients [2, 8, 13]. Next, we apply what we just established and combine our geometric considerations with approaches to efficient archetypal analysis so as to devise an NMF algorithm that notably differs from the techniques above.

4 Yet Another NMF Algorithm

Due to its practical utility, research on NMF has produced vast literature. Yet, except for only a few contributions (most notably [4, 10]), most NMF algorithms to date vary the ideas in [11, 14]. Our approach in this section, however, does not involve constrained optimization. It is related to the work in [4, 10] which apply geometric criteria to find suitable basis vectors. We extend these ideas in that we consider basis selection heuristics recently developed for archetypal analysis and demonstrate that NMF coefficients, too, can be computed without constrained optimization.

Above, we saw that optimal NMF is an NP hard problem. We further saw that traditional algorithms attempt to determine matrices \mathbf{W} and \mathbf{H} simultaneously but that the geometry of non-negative data allows for a decoupled estimation of both matrices. While it is comparatively simple to determine coefficients once basis vectors are available, the difficulty lies in finding suitable basis vectors. We therefore first discuss estimating \mathbf{W} and then address the task of computing \mathbf{H} .

4.1 Computing Matrix W

In order to compute suitable basis vectors for a non-negative factorization of a given data set $\mathcal{X} = \{\mathbf{x}_j\}_{j=1}^n, \mathbf{x}_j \in \mathbb{R}^m$, we first transform the data using (6) and obtain a set of stochastic vectors $\mathcal{Y} = \{\mathbf{y}_j\}_{j=1}^n, \mathbf{y}_j \in \Delta^{m-1}$. Figure 4 illustrates this step by means of an examples of 3-dimensional data.

We note again that if we could determine the vertices $\mathbf{w}_1, \dots, \mathbf{w}_p$ of the convex hull of \mathcal{Y} where $p \leq n$, we could perfectly reconstruct the given data as

$$\mathbf{x}_j = \sum_{i=1}^p h_{ij} \mathbf{w}_i, \quad h_{ij} \geq 0 \quad \forall i. \quad (7)$$

However, in NMF we are interested in finding k basis vectors where k is usually chosen to be small. Yet, given the example in Fig. 4, we recognize that for higher dimensional data the convex hull of \mathcal{Y} generally consists of many vertices so that p likely exceeds k . We are thus dealing with two problems: how to determine the vertices of \mathcal{Y} and how to select k of them such that

$$\sum_{j=1}^n \left\| \mathbf{x}_j - \sum_{i=1}^k h_{ij} \mathbf{w}_i \right\|^2 \quad (8)$$

is as small as possible given that all the h_{ij} are non-negative?

These problems are indeed at the heart of recent work on archetypal analysis where it was shown that reasonable results can be obtained using the method of simplex volume maximization (SiVM) [17, 19] which answers both questions simultaneously. The idea is to select k points in \mathcal{Y} that enclose a volume that is as large as possible. Given n points, it is easy to show that the $k \ll n$ points that enclose the largest volume will indeed be vertices of \mathcal{Y} .

Following the approach in [17], we apply *distance geometry* and note that the volume of a set of k vertices $\mathcal{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_k\} \subseteq \mathcal{Y}$ is given by

$$V^2(\mathcal{W}) = \frac{-1^k}{2^{k-1}((k-1)!)^2} \det(\mathbf{A}) \quad (9)$$

where

$$\det(\mathbf{A}) = \begin{vmatrix} 0 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & d_{11}^2 & d_{12}^2 & \dots & d_{1k}^2 \\ 1 & d_{11}^2 & 0 & d_{22}^2 & \dots & d_{2k}^2 \\ 1 & d_{12}^2 & d_{22}^2 & 0 & \dots & d_{3k}^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & d_{1k}^2 & d_{2k}^2 & d_{3k}^2 & \dots & 0 \end{vmatrix}$$

is the Cayley-Menger Determinant whose elements indicate distance between the elements in \mathcal{W} and are simply given by

$$d_{rs}^2 = \|\mathbf{w}_r - \mathbf{w}_s\|^2. \quad (10)$$

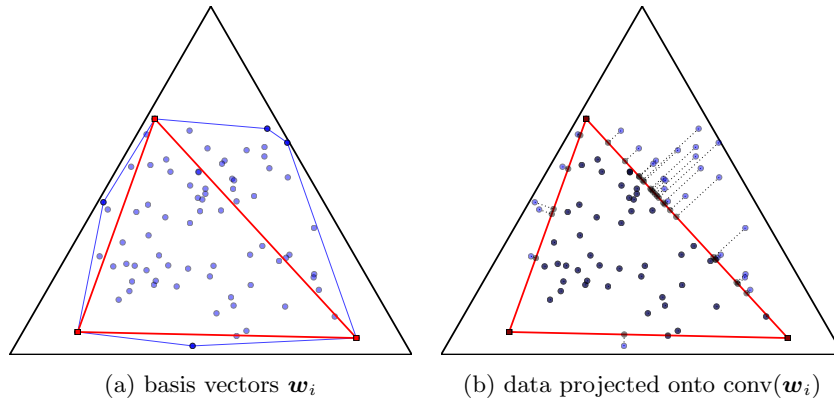


Fig. 5: $k = 3$ basis vectors found by SiVM through greedy stochastic hill climbing and projections of data points onto the corresponding convex hull $\text{conv}(\mathbf{w}_i)$.

Algorithm 1 SiVM through greedy stochastic hill climbing

```

randomly select  $\mathcal{W} \subset \mathcal{Y}$ 
for  $\mathbf{y}_j \in \mathcal{Y}$  do
  for  $\mathbf{w}_i \in \mathcal{W}$  do
    if  $V(\mathcal{W} \setminus \{\mathbf{w}_i\} \cup \{\mathbf{y}_j\}) > V(\mathcal{W})$  then
       $\mathcal{W} \leftarrow \mathcal{W} \setminus \{\mathbf{w}_i\} \cup \{\mathbf{y}_j\}$ 

```

We note again that NMF is an NP hard problem and that there is no free lunch. That is, even if we reduce the estimation of \mathbf{W} to the problem of selecting suitable vertices in \mathcal{Y} , we are still dealing with a subset selection problem of the order of $\binom{n}{k}$. Aiming at efficiency, we resort to a greedy stochastic hill climbing variant of SiVM that was proposed in [9]. It initializes \mathcal{W} by randomly selecting k points from \mathcal{Y} then iterates over the $\mathbf{y}_j \in \mathcal{Y}$ and tests if replacing any of the $\mathbf{w}_i \in \mathcal{W}$ by \mathbf{y}_j would lead to a larger volume. If so, the replacement is carried out and the search continues. Pseudocode of this procedure is shown in Algorithm 1 and Fig. 5(a) shows $k = 3$ basis vectors found in our example.

Concluding this subsection, we note that the basis vectors \mathbf{w}_i determined from the simplex projected data \mathbf{y}_j are all stochastic vectors whose entries are greater or equal than zero and sum to one. In contrast to conventional NMF approaches they are thus comparable in nature and do not suffer from ambiguous scales.

4.2 Computing Matrix H

Once a set $\mathcal{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_k\}$ of k basis vectors has been selected from the simplex projected data \mathbf{y}_j , every original data point \mathbf{x}_j that lies within the

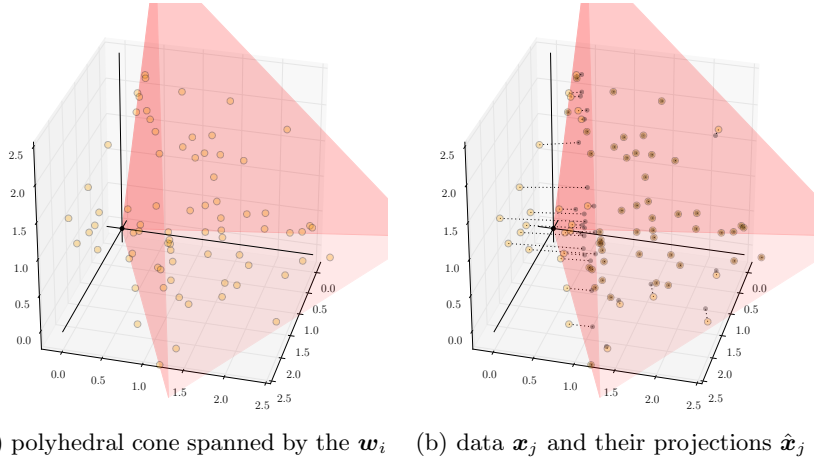


Fig. 6: Non-negative data and polyhedral cone spanned by $k = 3$ basis vectors found through SiVM. If each \mathbf{x}_j is projected to its closest point $\hat{\mathbf{x}}_j$ on this cone, it is easy to determine coefficients $h_{ij} \geq 0$ such that $\mathbf{x}_j \approx \hat{\mathbf{x}}_j = \sum_i h_{ij} \mathbf{w}_i$.

polyhedral cone spanned by the \mathbf{w}_i can be perfectly reconstructed as

$$\mathbf{x}_j = \sum_{i=1}^k h_{ij} \mathbf{w}_i, \quad h_{ij} \geq 0 \quad \forall i. \quad (11)$$

However, points outside that polyhedral cone cannot be expressed using non-negative coefficients. Typically, the best possible non-negative coefficients would therefore be determined using constrained least squares optimization. Here, we consider a different idea namely to project every \mathbf{x}_j to its closest point in the polyhedral cone of the \mathbf{w}_i and to determine coefficients for the projected point.

To achieve this, we first project the \mathbf{y}_j onto the convex hull of the \mathbf{w}_i in the simplex Δ^{m-1} and note that there are highly efficient computational geometry algorithms for this purpose [16, 22]. Figure 5(b) shows the corresponding result for our running example.

Let \mathbf{z}_j denote the closest point of \mathbf{y}_j in the convex hull of the \mathbf{w}_i . We then rescale the \mathbf{z}_j to unit length, i.e.

$$\mathbf{z}_j \leftarrow \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|} \quad (12)$$

and compute

$$\hat{\mathbf{x}}_j = \mathbf{z}_j \cdot (\mathbf{z}_j^T \mathbf{x}_j) \quad (13)$$

for all the original data vectors and thus obtain the point $\hat{\mathbf{x}}_j$ in the polyhedral cone of the \mathbf{w}_i that is closest to \mathbf{x}_j . The corresponding result in our example can be seen in Fig. 6 which shows the original data and their projections onto the polyhedral cone spanned by the $k = 3$ basis vectors found previously.

The $\hat{\mathbf{x}}_j$ are then gathered in a matrix $\hat{\mathbf{X}}$ and a unique coefficient matrix \mathbf{H} that, by nature of the $\hat{\mathbf{x}}_j$, will only contain non-negative entries is computed as

$$\mathbf{H} = (\mathbf{W}^T \mathbf{W})^{-1} \mathbf{W}^T \hat{\mathbf{X}} \quad (14)$$

so that we indeed obtain two factor matrices \mathbf{W} and \mathbf{H} for which $\mathbf{WH} \approx \mathbf{X}$.

5 Conclusion

In this paper, we first discussed traditional approaches to non-negative matrix factorization and pointed out some of the difficulties that arise in this context. We then assumed a geometric point of view on the problem and showed in a step by step construction that it is possible to compute NMF of a data matrix without having to resort to sophisticated methods from optimization theory.

We believe that there are several advantages to our approach. First of all, it is computationally simple and allows for parallelization. Second of all, it is intuitive and easy to visualize and thus provides alternative avenues for teaching this material to students. Third of all, it also creates new perspectives for NMF research. While traditional, optimization-based approaches to NMF are very well understood by now and most related recent publications are but mere variations of a common theme, the idea of matrix factorization as search for suitable basis vectors by means of geometric objectives such as maximum volumes raises new questions. For instance, in ongoing work we are currently exploring the role of *entropy* in NMF. Given the pullback onto the simplex, it is obvious to consider the entropy of the resulting stochastic vectors as a criterion for their selection as possible basis vectors. Indeed, points with lower entropy are situated closer to the simplex boundary and therefore seem appropriate candidates for basis vectors. Corresponding search algorithms are under development and we hope to report results soon.

References

1. Aloise, D., Deshapande, A., Hansen, P., Popat, P.: NP-Hardness of Euclidean Sum-of-Squares Clustering. *Machine Learning* 75(2), 245–248 (2009)
2. Bauckhage, C., Thureau, C.: Making Archetypal Analysis Practical. In: *Pattern Recognition*. LNCS, vol. 5748, pp. 272–281. Springer (2009)
3. Berry, M., Browne, M., Langville, A., Pauca, V., Plemmons, R.: Algorithms and Applications for Approximate Nonnegative Matrix Factorization. *Computational Statistics and Data Analysis* 52(1), 155–173 (2007)
4. Chu, M., Lin, M.: Low-Dimensional Polytope Approximation and Its Applications to Nonnegative Matrix Factorization. *SIAM J. on Scientific Computing* 30(3), 1131–1155 (2008)
5. Cichocki, A., Zdunek, R., Phan, A., Amari, S.: *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley (2009)
6. Cutler, A., Breiman, L.: Archetypal Analysis. *Technometrics* 36(4), 338–347 (1994)

7. Donoho, D., Stodden, V.: When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts? In: Proc. NIPS (2003)
8. Eugster, M., Leisch, F.: Weighted and Robust Archetypal Analysis. *Computational Statistics & Data Analysis* 55(3), 1215–1225 (2011)
9. Kersting, K., Bauckhage, C., Thureau, C., Wahabzada, M.: Matrix Factorization as Search. In: Proc. Eur. Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2012)
10. Klingenberg, B., Curry, J., Dougherty, A.: Non-negative Matrix Factorization: Ill-posedness and a Geometric Algorithm. *Pattern Recognition* 42(5), 918–928 (2008)
11. Lee, D., Seung, S.: Learning the Parts of Objects by Non-Negative Matrix Factorization. *Nature* 401(6755), 788–791 (1999)
12. Lin, C.J.: Projected Gradient Methods for Non-negative Matrix Factorization. *Neural Computation* 19(10), 2756–2779 (2007)
13. Morup, M., Hansen, L.: Archetypal Analysis for Machine Learning and Data Mining. *Neurocomputing* 80, 54–63 (2012)
14. Paatero, P., Tapper, U.: Positive Matrix Factorization: A Non-negative Factor Model with Optimal Utilization of Error Estimates of Data Values. *Environmetrics* 5(2), 11–126 (1994)
15. Recht, B., Re, C., Tropp, J., Bittorf, V.: Factoring Nonnegative Matrices with Linear Programs. In: Proc. NIPS (2012)
16. Sekitani, K., Yamamoto, Y.: A Recursive Algorithm for Finding the Minimum Norm Point in a Polytope and a Pair of Closest Points in Two Polytopes. *Mathematical Programming* 61(1), 233–249 (1993)
17. Thureau, C., Kersting, K., Bauckhage, C.: Yes We Can – Simplex Volume Maximization for Descriptive Web-Scale Matrix Factorization. In: Proc. Int. Conf. on Information and Knowledge Management. ACM (2010)
18. Thureau, C., Kersting, K., Wahabzada, M., Bauckhage, C.: Convex Non-negative Matrix Factorization for Massive Datasets. *Knowledge and Information Systems* 29(2), 457–478 (2011)
19. Thureau, C., Kersting, K., Wahabzada, M., Bauckhage, C.: Descriptive Matrix Factorization for Sustainability: Adopting the Principle of Opposites. *Data Mining and Knowledge Discovery* 24(2), 325–354 (2012)
20. Vasiloglou, N., Gray, A., Anderson, D.: Non-Negative Matrix Factorization, Convexity and Isometry. In: Proc. SIAM Int. Conf. on Data Mining (2009)
21. Vavasis, S.: On the Complexity of Nonnegative Matrix Factorization. *SIAM J. on Optimization* 20(3), 1364–1377 (2009)
22. Wolfe, P.: Finding the Nearest Point in a Polytope. *Mathematical Programming* 11(1), 128–149 (1976)

Learning Spatial Interest Regions from Videos to Inform Action Recognition in Still Images

A. Eweiwi¹, M.S. Cheema¹, and C. Bauckhage^{1,2}

¹B-IT, University of Bonn, Bonn, Germany

²Fraunhofer IAIS, Sankt Augustin, Germany

Abstract. Common approaches to human action recognition from images rely on local descriptors for classification. Typically, these descriptors are computed in the vicinity of key points which either result from running a key point detector or from dense or random sampling of pixel coordinates. Such key points are not a-priori related to human activities and thus of limited information with regard to action recognition. In this paper, we propose to identify action-specific key points in images using information available from videos. Our approach does not require manual segmentation or templates but applies non-negative matrix factorization to optical flow fields extracted from videos. The resulting basis flows are found to be indicative of action specific image regions and therefore allow for an informed sampling of key points. We also present a generative model that allows for characterizing joint distributions of regions of interest and a human actions. In practical experiments, we determine correspondences between regions of interest that were automatically learned from videos and manually annotated locations of human body parts available from independent benchmark image data sets. We observe high correlations between learned interest regions and body parts most relevant for different actions.

Keywords: optical flow, non-negative matrix factorization.

1 Introduction

Research on recognizing human activities from still images is motivated by promising applications in automatic indexing of very large image repositories and also contributes to problems in automatic scene description, context dependent object recognition, or human pose estimation [4, 13, 25, 28].

Currently, approaches to action recognition can be categorized into two main classes: (a) pose-based and (b) bags-of-features (BoF) methods. Stirred by the idea of *poselets* [3], a notion of part-based templates, pose-based approaches have recently been met with rekindled interest [21, 25]. However, the construction of

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Baecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>



Fig. 1. Examples of still images in which we can easily recognize human activities even if neither image shows all of the human body.

poselets requires cumbersome manual annotations which impede their use in BIG DATA settings. BoF approaches are known for their good performance in object recognition and have therefore been adapted to action recognition [5]. Yet, local image descriptors are typically computed in the vicinity of key points that result from low-level signal analysis or dense or random sampling and are therefore uninformative or independent of the activity depicted in an image.

Most physical activities of people show characteristic articulations and movements of different body parts. Yet, although activities are inherently dynamic, the human visual system easily infers human activities from still images that show posture or limb configurations. Consider, for instance, the images in Fig. 1 which we can interpret even without a full view of the human body. This raises the question if it is possible to automatically learn or identify action-specific, informative, regions of interest in still images without having to rely on exhaustive mining of low-level image descriptors or labor-intensive annotations?

In an attempt to answer this question, we propose an efficient approach towards automatically learning of action specific regions of interest in still images. Considering the fact that activities are temporal phenomena, we make use of information available from videos. Given videos that show human activities, we compute optical flow fields and consider the magnitudes of flow vectors in each frame. Given a collection of frame-wise flow magnitudes, we apply non-negative matrix factorization (NMF) and obtain basis flows. These basis flows are indicative of the position and configuration of different limbs or body parts whose motion characterizes certain activities. Viewed as images, the basis flows indicate action specific regions of interest and therefore allow for an informed sampling of key points for subsequent feature extraction. We also devise a generative probabilistic model that characterizes joint distributions of regions of interest and human actions. To evaluate our approach, we consider correspondences between regions of interest that were automatically learned from videos and manually annotated locations of human body parts that are available from independent data sets of still images. Our empirical results reveal a high correlation between extracted interest regions and those body parts that are most relevant for different actions.

2 Related Work

Approaches that rely on the idea of bags of visual words (BoWs) are popular because BoWs are known for their simplicity, robustness, and good performance in content-based image or video classification. Corresponding work treats an image as a collection of independent visual descriptors computed at key point locations. Computing key points is crucial within the BoW framework since it preselects image patches subsequent classification. Naturally, one would like to focus only on those patches that are most discriminative.

So far, BoW approaches [18, 22] based on key points detection [2, 12, 20, 23], though generally discriminative, do not regard task specific objectives in key point localization. Rather, key point locations are determined from low-level properties of the image signal. Moreover, corresponding approaches typically assume key points to be independent and therefore fail to explain characteristics of spatial layouts. The work in [15] therefore proposes a representation that encodes spatial relationships among key points. The authors of [11] employ data mining to build high-level compound features from noisy and over-complete sets of low-level features and the work [24] uses a triangular lattice of grouped point features to encode spatial layouts. Still, these approaches, too, center around low-level signal properties which do not necessarily provide an accurate account of the characteristics of an activity.

Sampling techniques such as random sampling have shown good performances, too. The authors of [7] empirically demonstrate that random sampling provides equal or better activity classifiers than sophisticated multi-scale interest point detectors; yet, their work also illustrates that the most important aspect of sampling is the number of sample points extracted. The authors of [27] claim that dense sampling outperforms all point detectors in realistic scenarios and. Yet, at the same time, recent work in [10] shows that state-of-the-art performance in action recognition can also be obtained from only a few randomly sampled key points. It therefore appears that the jury is still out on whether to use dense or random sampling and methods which mark a middle ground, namely informed sampling, seem to merit closer investigation. It is, however, obvious that the success of dense sampling is bought at the expense of memory- and runtime efficiency whereas random sampling methods do not provide statistical guarantees as to the adequacy for the task at hand.

Part-based approaches, too, are popular in research on action recognition and have been shown to successfully cope with the *PASCAL* challenge. The authors of [9] describe a deformable model which achieves good performance on benchmark data sets [5]. The work in [3] introduces exemplar-based pose representation, or *poselets*, for human detection. This term denotes a set of patches with similar pose configurations. The work in [21] utilizes poselets for identifying human poses and actions in still images and the authors of [25] propose an articulated part-based model for human pose estimation and detection which adapts a hierarchical (coarse-to-fine) representation. Despite their recent success, it is still questionable if these methods can make use of the favorable statistics of

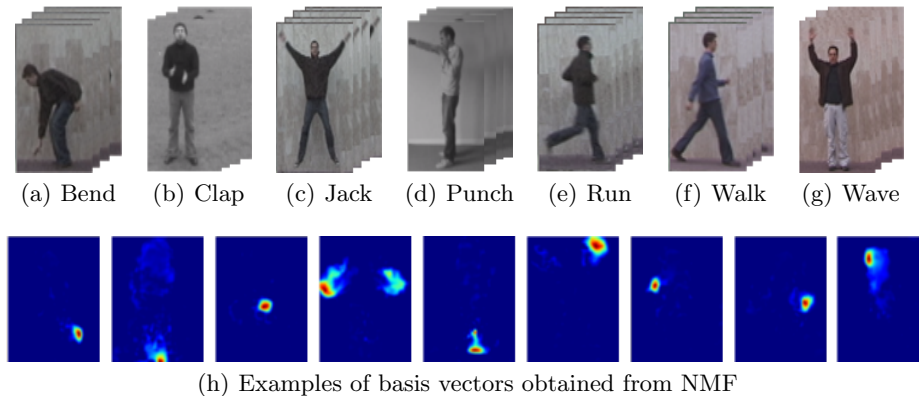


Fig. 2. (a–g) Examples of training videos from the Weizmann and KTH data sets; (h) examples of basis flows obtained from applying NMF to optical flow fields.

present day large scale data sets because the construction of suitable poselets requires extensive human intervention and manual labeling in the training phase.

The authors of [26] consider non-negative matrix factorization (NMF) for action recognition and apply it to learn pose- and background primitives. In [1], the authors estimate the human upper body pose through NMF and [16, 17] apply non-negative factor models to recognize activities from videos. The authors of [29] empirically evaluate human action recognition using pose- or appearance-based features and conclude that, even for rather coarse pose representations, pose-based features either match or outperform appearance-based features. However, they acknowledge that appearance-based features still represent an ideal resort for cases of considerable visual occlusion. Accordingly, it appears worthwhile to study methods that combine both approaches into a single framework.

Next, we discuss how the approach proposed in this paper indeed provides a method for the informed sampling of key points for appearance-based action recognition as well as an approach to learning descriptors of body poses.

3 Learning Action-specific Interest Regions from Videos

Our approach identifies discriminative regions in an image and subsequently learns the relative importance of those regions for different actions. In order to identify interesting spatial locations, we apply NMF to optical flow fields obtained from videos. Furthermore, we exploit NMF mixture coefficients to derive a generative probabilistic model that features joint distributions of regions of interest and human actions.

3.1 Learning NMF Bases

Given videos of different actions, we determine optical flow magnitudes at each pixel in a box of constant size surrounding a person visible in the video. Each frame can be transformed into an m dimensional non-negative vector \mathbf{v} . Let n_i represent the number of frames for an action $a_i \in \mathcal{A} = \{a_1, a_2, \dots, a_r\}$ and let $n = \sum_{i=1}^r n_i$. We build an $m \times n$ data matrix \mathbf{V} containing the flow magnitude vectors of all frames. Computing NMF yields k basis vectors, or *basis flows*, such that $\mathbf{V} \approx \mathbf{WH}$ where the columns of $\mathbf{W}_{m \times k}$ are non-negative basis elements and the columns of $\mathbf{H}_{k \times n}$ encode non-negative mixing coefficients.

In order to compute the factors \mathbf{W} and \mathbf{H} , we apply the algorithm according to Lee and Seung [19]. This method is known to yield sparse basis elements for it converges to vectors that lie in the facets of the simplicial cone spanned by the data (see the discussions in [6, 14]). Accordingly, we can expect the resulting basis flows to be sparse in the sense that most elements of a basis element \mathbf{w}_l will be (close to) zero and only a few entries will have noticeable values. Figure 2 (h) shows that this is indeed the case. It depicts pictorial representations of exemplary basis vectors \mathbf{w}_l resulting from NMF. Note that for each basis element only a few pixels are larger than zero; in each case, these pixels apparently form distinct, more or less compact patches in the image plane.

3.2 Learning the Action-specific Importance of Basis Flows

Different actions are characterized by articulation and movements of different body parts. The NMF basis vectors determined through factorization of frame-wise optical flow magnitudes appear to indicate image regions of importance for different activities. Here, we propose to learn the relative importance of different basis elements with respect to different actions. To this end, we consider the matrix \mathbf{H} since its entries encode linear mixing coefficients required to reconstruct the vectors in \mathbf{V} from the basis flows in \mathbf{W} . Consequently, the columns of \mathbf{H} encode the relevant importance of a basis for a given frame. Normalizing them to stochastic vectors allows us to estimate a joint probability distribution of actions and bases. The conditional probability of basis \mathbf{w}_l given an action a_i is determined as

$$p(\mathbf{w}_l | a_i) = \frac{\sum_f h_{lf}}{\sum_{j,f} h_{jf}} \quad (1)$$

where the summation index f indicates all columns \mathbf{v}_f in \mathbf{V} that show activity a_i and index j ranges from 1 to k .

Figure 3 plots the resulting distribution. Note that this probability distribution, i.e. the set of weights of a basis element w.r.t. an action, again is sparse. The distribution in equation (1) immediately allows us to determine how characteristic a certain basis flow is for an activity.

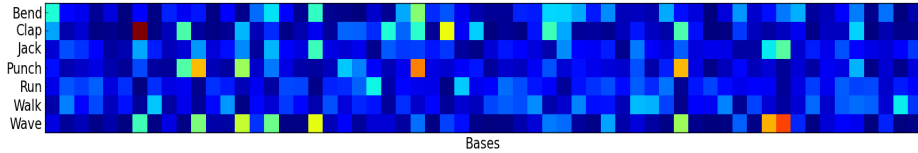


Fig. 3. Relative importance of bases w.r.t. different actions according to $p(\mathbf{w}_l|a_i)$. Note that actions flows can be approximated by a small number of basis vectors.

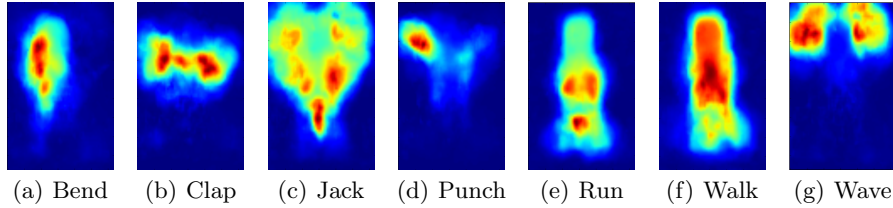


Fig. 4. Examples of action signatures resulting from equation (2).

The probability distribution $p(\mathbf{w}_l|a_i)$ in (1) also allows us to consider *action signatures* which we define to be the conditional expectations

$$\mathbf{s}_i = \sum_{l=1}^k p(\mathbf{w}_l|a_i) \mathbf{w}_l. \quad (2)$$

Computing and plotting action signatures s_i for different actions a_i , we find that characteristically different regions in the image plane are intensified for different actions. Figure 4 shows examples of action signatures which we obtained from basis flows extracted from the Weizmann¹ and KTH² data sets. Apparently, action signatures like these may serve two purposes. On the one hand, they provide us with a prior distribution for the sampling of interest points from still images showing people in order to compute action specific local features for activity classification. On the other hand, action signatures may be used as templates or filter masks for pose-based activity recognition.

3.3 Evaluation Methodology

To evaluate as to how far regions of interest extracted by our approach match the locations of human body parts in real images, we consider the manually annotated positions of limbs that are available in the H3D³ and VOC2011⁴ data sets. In particular, we determine the joint probability distribution of actions,

¹ www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html

² www.nada.kth.se/cvap/actions/

³ www.eecs.berkeley.edu/~lbourdev/h3d/

⁴ pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/

interest regions, and body parts. Given the locations of a body part b_j in an image of action a_i , we assume the following conditional independence model

$$p(b_j, \mathbf{w}_l, a_i) = p(b_j|a_i) p(\mathbf{w}_l|a_i) p(a_i). \quad (3)$$

Using (1) and taking the prior $p(a_i)$ to be uniform, allows for solving for $p(b_j|a_i)$ which can be understood to encode the relative importance of different body parts for different actions a_i .

4 Experimental Results

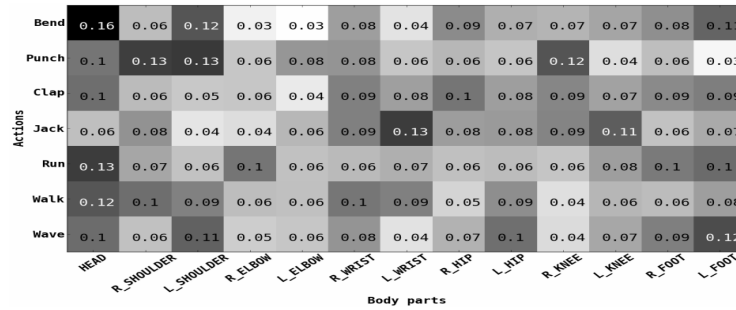
In order to learn action specific regions of interest, we considered the Weizmann and KTH data sets. As these video collections show little variations of background and view-point, they allow us to focus on estimating the importance of different body parts for different actions. In particular, we focused on the following actions *Bend*, *Clap*, *Jack*, *Punch*, *Run*, *Walk*, and *Wave*. We used the bounding boxes provided by [30] and resized them to size 88×64 . To determine optical flows, we considered the method due to Farneback [8]. Finally, all of the results reported below were obtained using 200 basis flows \mathbf{w}_i .

To evaluate the suitability of the resulting interest regions for still image based action recognition, we considered limb or joint annotations available in the H3D and VOC2011 data sets. We used 240 annotated images and determined the joint distribution of actions, interest regions, and body parts. For each of the selected action classes, we considered the location of 13 body parts or joints including, for example, head, feet, knees, hips, shoulders, elbows, and hands.

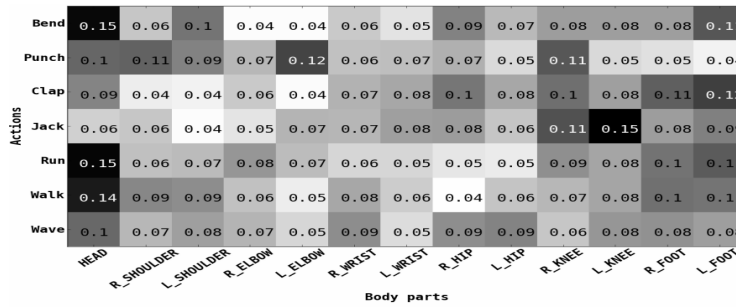
We compared our interest regions to key points extracted by the popular Harris [12] and SIFT [20] key point detectors. In each case, we selected key points with the highest response in every image, assigned them to their nearest annotated body part, and normalized the resulting histogram. For each action, we obtained a stochastic vector by iterating over all images of that action thus representing the conditional distribution $p(b_j|a_i)$ discussed above.

Figure 5 compares results due to our approach of extracting interesting regions from video data to the ones obtained from using Harris and SIFT key points. It shows the relative importance of different body parts for different actions. In case of Harris and SIFT key points, head and feet dominate other limbs regardless of the action (Fig. 5 (a) and (b)). Furthermore, the probabilities for other body parts are almost uniform and do not convincingly relate to the different actions. For example, body parts naturally characterizing *Clap*, i.e. elbows, and hands, achieved rather low scores.

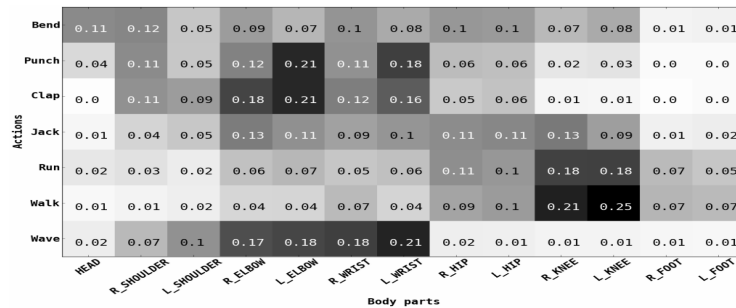
On the other hand, our approach exhibits logically coherent relationships between body parts and actions (Fig. 5(c)). Compare, for instance, the varying importance of different body parts for *clapping* and *running*. Clearly the lower body parts are dominant for the action of running while the arms are of higher importance for the action of clapping. From the perspective of body parts observe that, for instance, the head is less relevant for actions such as *Clap* or *Run* as compared to *Bend*. Figure 6 visualizes these results using stick figures where



(a) Importance of body parts using spatial distribution of Harris corners



(b) Importance of body parts using spatial distribution of SIFT key points



(c) Importance of body parts using our approach

Fig. 5. Conditional probabilities of body parts w.r.t actions. Our approach (c) exhibits logically coherent relationships between body parts and actions as compared to appearance based sampling using Harris corners (a) and SIFT interest points (b).

the size of plotted body parts correspond to their relevance for an activity. In general these results suggest that the regions of interest which we obtain from factorizing flow fields are well correlated with the locations of action specific body parts available from independent sets of manually annotated images.

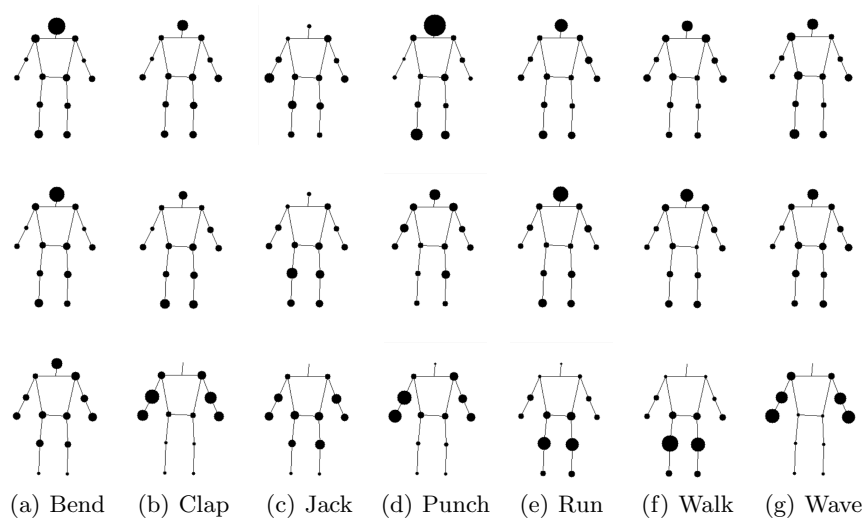


Fig. 6. Stick figures depicting the relevance of different body parts for different actions. Important key points computed using the Harris detector (first row) and SIFT detector (second row) hardly correlate to action-specific body parts; interest regions from our approach correlate better (third row).

5 Conclusion and Future Work

We presented an approach to the automatic detection of regions of interest for human action recognition in still images. Since human activities are inherently dynamic in nature, we proposed to learn interest regions from optical flow fields extracted from video sequences of human actions. Using non-negative matrix factorization, we obtained sets of basis flows which were found to be indicative of the location of different limbs or joints in different activities. Our approach fundamentally differs from existing pre-processing approaches for action recognition in still images. First, although we consider rather low-level properties of videos of activities, the characteristics of optical flow enable us to identify locations of body parts whose articulation define an action. Consequently, unlike common bag-of-features approaches, our approach facilitates informed sampling of key points in the image plane. Second, the proposed concept of action signatures provides probabilistic templates for pose-based recognition. Compared to common approaches based on distributed pose representations, our approach does not require meticulous manual annotation of images or frames and thus offers more scalability and convenience for large data sets. Also, compared to conventional part based approaches, our approach does not assume an underlying elastic model of body but provides priors even for cluttered or occluded images. This paper therefore established a baseline for video-based feature selection towards action recognition in still images.

The logical next step for future work is, of course, to build activity classifiers based on information available from action specific regions of interest. To this end, we currently consider standard descriptors(e.g. HOG, SIFT, SURF) which are computed at locations determined according to the probabilities encoded in action signatures.

References

1. Agarwal, A., Triggs, B.: A local basis representation for estimating human pose from cluttered images. In: ACCV (2006)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. *CVIU* 110(3), 346–359 (2008)
3. Bourdev, L., Malik, J.: Poselets: body part detectors trained using 3d human pose annotations. In: ICCV (2009)
4. Cheema, M., Eweiwi, A., Thureau, C., Bauckhage, C.: Action recognition by learning discriminative key poses. In: ICCV Workshop on Performance Evaluation of Recognition of Human Actions (2011)
5. Deltaire, V., Laptev, I., Sivic, J.: Recognizing human actions in still images: A study of bag-of-features and part-based representations. In: BMVC (2010)
6. Donoho, D., Stodden, V.: When Does Non-negative Matrix Factorization Give a Correct Decomposition into Parts? In: NIPS (2004)
7. E.Nowak, Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: ECCV (2006)
8. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: SCIA (2003)
9. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. *TPAMI* 32, 1627 – 1645 (2010)
10. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.S.: Hough forests for object detection, tracking, and action recognition. *TPAMI* 33, 2188–2202 (2011)
11. Gilbert, A., Illingworth, J., Bowden, R.: Action recognition using mined hierarchical compound features. *TPAMI* 33, 883 – 897 (2011)
12. Harris, C., Stephens, M.: A combined corner and edge detection. In: In Alvey Vision Conference (1988)
13. Johnson, S., Everingham, M.: Learning effective human pose estimation from inaccurate annotation. In: CVPR (2011)
14. Klingenberg, B., Curry, J., Dougherty, A.: Non-negative matrix factorization: Ill-posedness and a geometric algorithm. *PR* 42(5), 918–928 (2008)
15. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: CVPR (2010)
16. Krausz, B., Bauckhage, C.: Action recognition in videos using nonnegative tensor factorization. In: ICPR (2010)
17. Krausz, B., Bauckhage, C.: Loveparade 2010: Automatic video analysis of a crowd disaster. *CVIU* 116(3), 307–319 (2012)
18. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
19. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755), 788–799 (1999)
20. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* 60(2), 91–110 (2004)

21. Maji, S., Bourdev, L., Malik, J.: Action recognition from a distributed representation of pose and appearance. In: CVPR (2011)
22. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: action recognition through the motion analysis of tracked features. In: ICCV Workshop on Video-Oriented Object and Event Classification (2009)
23. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. IJCV 37, 151–172 (2000)
24. Song, Y., Goncalves, L., Perona, P.: Unsupervised learning of human motion. TPAMI 25, 814 – 827 (2003)
25. Sun, M., Savarese, S.: Articulated part-based model for joint object detection and pose estimation. In: ICCV (2011)
26. Thureau, C., Hlavac, V.: Pose primitive based human action recognition in videos or still images. In: CVPR (2008)
27. Wang, H., Ullah, M.M., Klaeser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2009)
28. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. CVIU 115, 224–241 (2011)
29. Yao, A., Gall, J., Fanelli, G., Van Gool, L.: Does human action recognition benefit from pose estimation? In: BMVC (2011)
30. Yao, A., Gall, J., Van Gool, L.: A hough transform-based voting framework for action recognition. In: CVPR (2010)

Demonstration von thematischen Frames im TopicExplorer-System

Extended Abstract

Alexander Hinneburg¹, Frank Rosner¹, Stefan Peßler² und Christian Oberländer²

¹Informatik, Martin-Luther-Universität Halle-Wittenberg

²Japanologie, Martin-Luther-Universität Halle-Wittenberg

hinneburg@informatik.uni-halle.de

frank.rosner@student.uni-halle.de

stefan.peessler@japanologie.uni-halle.de

christian.oberlaender@japanologie.uni-halle.de

Themenmodelle bieten sich an, die Inhalte großer Dokumentensammlungen zu erforschen. Thematische Wortlisten präsentieren typische Inhalte. Diese Themen werden automatisch gelernt, ohne das Dokumente manuell annotiert werden müssen. Während des Lernens eines Themenmodells werden die Wörter der Dokumente Themen zugeordnet. Dabei werden zwei gegenläufige Ziele verfolgt: erstens, einem Thema sollen so wenig wie möglich verschiedene Wörter zugeordnet werden und zweitens, ein Dokument soll so wenig wie möglich verschiedene Themen enthalten [2]. Die unüberwachten Lernalgorithmen finden Kompromisslösungen für diese Aufgabenstellung, welche im Fall von Variationsinferenz zu lokalen Extrema der freien Energiefunktion des Modells und im Fall von Gibbs-Samplern zu wahrscheinlichen Zuständen einer Markov-Kette korrespondieren. In keinem Fall garantieren die Algorithmen, dass die berechneten Themen gut durch Menschen interpretierbar sind.

Es ist state-of-the-art die Themen, welche mathematisch gesehen Wahrscheinlichkeitsverteilungen über Wörtern sind, durch die wahrscheinlichsten Wörtern zu repräsentieren. Die Interpretation dieser Wortlisten kann jedoch eine schwierige Aufgabe für den Anwender sein. Eine erfolgreiche Interpretation hängt vom Hintergrundwissen der Person und der Vertrautheit mit dem genutzten Vokabular ab. Zwei wesentliche Probleme können die Interpretation eines Thema beeinträchtigen. Erstens, thematische Wortlisten können komplett aus Substantiven bestehen, deren Beziehungen zueinander mehrdeutig sein können. Ein Beispiel ist ein Thema, das durch eine Liste von Ländernamen repräsentiert wird. Trotz dessen, dass alle Länder in einer eng umgrenzten Region liegen können, gibt es immer noch mehrere verschiedene Interpretationen, die zu einer solchen Liste passen würden. Deshalb ist sie nicht gut interpretierbar. Ein zweiter Grund kann darin liegen, dass die präsentierten Wörter dem Anwender als unzusammenhängend erscheinen. Dies kann an Wörtern liegen, die der Anwender nicht kennt.

Es ist eine offene Frage, wie durch Themenmodelle berechnete Themen so repräsentiert werden können, dass sie klar und eindeutig durch Menschen interpretiert werden

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

können. Jüngste Forschungen zur Messung von Kohärenz von Themen zeigen, dass sie besser durch Menschen interpretiert werden können, wenn Paare von Wörtern des Themas oft in Dokumenten nahe beieinander stehen [3]. Deshalb kann eine Themenrepräsentation leichter interpretierbar sein, die Paare von wahrscheinlichen Wörtern zeigt, die oft in Dokumenten nahe beieinander stehen. Dieser Ansatz löst jedoch nicht das Problem von Substantivlisten. Eine Schlüsselbeobachtung ist, dass Verben in Themenverteilungen oft weniger wahrscheinlich als Substantive sind, weil sie flexibler in verschiedenen Kontexten gebraucht werden können. Deshalb tauchen Verben in nach Themenwahrscheinlichkeit sortierten Wortlisten weiter hinten auf und müssen somit gesondert behandelt werden.

Wort-Kombinationen aus je einem Verb und einem Substantiv können als Basiseinheiten angesehen werden um Inhalte zu transportieren. Minski nannte in den ersten Forschungen zu künstlicher Intelligenz solche Einheiten Frames [1,4]. Deshalb stellt das Auftreten von einem Verb in der Nähe eines Substantivs in einem Dokument eine notwendige Bedingung für das Vorhandensein eines Frames dar. Ein thematischer Frame kann vorhanden sein, wenn ein Themenmodell ein Verb und ein Substantiv, die in einem Dokument nahe zusammenstehen, dem selben Thema zuweist. Unsere Demonstration zeigt anhand mehrerer Beispiele, dass Themen durch die Repräsentation mittels thematischer Frames interpretiert werden können, deren Inhalte allein durch das Zeigen von Wortlisten unklar bleiben würde. Die Entwicklung von Evaluationsmethoden für diese Aufgabe ist jedoch Gegenstand weiterer Forschungen.

Unsere Implementation von thematischen Frames ist in das TopicExplorer System (<http://topicexplorer.informatik.uni-halle.de/>) eingebettet. Wir demonstrieren die thematischen Frames mit Hilfe von verschiedenen Dokumentsammlungen in unterschiedlichen Sprachen. Die Sammlungen müssen allgemein bekannt sein, damit die Themen leicht und ohne Fachwissen zugänglich sind. Deshalb haben wir zu Demonstrationszwecken einen Teil der englischen Wikipedia sowie eine Sammlung deutscher Märchen als Dokumentsammlungen ausgewählt. Weiterhin zeigen wir als echte Anwendung des TopicExplorers und der thematischen Frames die Unterstützung von sozialwissenschaftlicher Forschung bei der Analyse von japanischen Blogs, welche die Auswirkungen der Fukushima-Katastrophe von 2011 und die soziale Verantwortung diskutieren. [Die englische Version des Artikels wird zur CIKM 2014 erscheinen.]

Danksagung

Wir danken Mattes Angelus, Benjamin Schandera und Gert Böhmer für ihre wertvollen Programmierbeiträge zur Code-Basis des TopicExplorer. Weiterhin danken wir der Klaus Tschira Stiftung für die finanzielle Unterstützung des Projektes.

Literatur

1. Allan, K.: Natural language semantics. Wiley (2001)
2. Blei, D.: Topic modeling and digital humanities. *Journal of Digital Humanities* 2(1) (2012)
3. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: Proceedings of the European Chapter of the Association for Computational Linguistics (2014)
4. Minsky, M.: Frame-system theory. In: Johnson-Laird, P.N., Wason, P.C. (eds.) *Thinking*. pp. 355–377. Cambridge: Cambridge University Press (1977)

Using Semantic Data Mining for Classification Improvement and Knowledge Extraction

Fernando Benites and Elena Sapozhnikova

University of Konstanz, 78464 Konstanz, Germany.

Abstract. The objective of this position paper is to show that the integration of semantic data mining into the DAMIART data mining system can help further improve classification performance and knowledge extraction. DAMIART performs multi-label classification in the presence of multiple class ontologies, hierarchy extraction from multi-labels and concept relation by association rule mining. Whereas DAMIART combines knowledge from multiple data sources and multiple class ontologies, the proposed extension should also explore available ontologies over attributes. This will allow the system to produce not only more accurate classification results but also improve their interpretability and overcome such problems as data sparseness.

Keywords: Semantic Data Mining, Linked Open Data, Ontology

1 Introduction

Data Mining is defined as the process of discovering implicit, novel, potentially useful and understandable patterns or relationships in large volumes of data [8]. In this context, conventional data mining algorithms treat the data simply as numbers lacking any semantic information and process them independently from the particular domain. Data preprocessing as well as interpretation of the obtained results are though domain-dependent tasks, which are usually solved by human experts possessing required domain knowledge. However, such knowledge can be very useful at any other stage of the data mining process, e.g. for choosing suitable data and proper mining techniques or for the effective pruning of the hypothesis space. So, it has been early realized that the incorporation of available domain knowledge is one of the most important problems in data mining [9]. Now, its importance is growing even more because the data are becoming more and more complex, and a manual approach to obtaining domain knowledge is not sufficiently efficient. With more interconnected data, more possible interpretations can be generated by data mining algorithms, overwhelming any human expert.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

The new field of Semantic Data Mining, which has emerged in the past few years, has suggested a possible solution to this problem: Domain knowledge can be derived from semantic data (data which include semantic information, e.g. ontologies or annotated data collections) and directly incorporated in the data mining process. The term Semantic Data Mining was first introduced by [15] in order to designate a data mining approach where domain ontologies are used as background knowledge for data mining (Fig. 1). It includes methods for systematic incorporation of domain knowledge in an intelligent data mining environment [12].

Alternatively, d’Amato et al. proposed the term Ontology Mining for the same research area, reflecting the importance of the role ontologies play in knowledge representation [4]. A domain ontology can be viewed as a model that contains the structural and conceptual information about the domain. It typically consists of all important concepts of the domain, their specific properties, the relationships between the concepts, and possibly additional restrictions on the domain. A common example may be an ontology for the tourism domain containing concepts such as accommodation, attractions and transport (where, for example, “hotel” and “youth hostel” are subconcepts of “accommodation”). Due to the rapid growth of the number of ontologies becoming available on the Web in a wide range of domains, semantic data mining has great potential in many application areas such as biology, sociology, and finance.

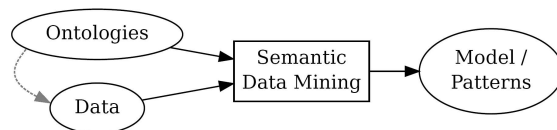


Fig. 1. Semantic Data Mining.

The main advantage of ontology-based systems is their ability of sharing knowledge among people as well as among computer systems. This was the motivation behind the development of the semantic Web [7] when companies and large institutions intended to automatically exchange data over the Internet. Nowadays, semantic Web ontologies have been established as a key technology for intelligent knowledge processing. This has resulted in a paradigm shift within the data mining community: instead of mining the large volumes of numerical data supported by scarce domain knowledge, the new challenge is to mine the abundance of knowledge encoded in domain ontologies, constrained by the heuristics computed from the data [11]. Recently developed semantic Web technologies, such as RDF (Resource Description Framework) and OWL (Ontology Web Language), enable domain knowledge to be captured automatically with minimum manual effort. The first attempts to utilize Linked Open Data (LOD) in data mining process have been shown to be successful in many application areas, including user recommendation systems [18], medical domain [13], Web search [17] and cross lingual text classification [14].

Another challenge of modern data mining is the connecting of different data sources. As a result of increasing data complexity, there is often a variety of interrelated data sources, which can all be used to describe the same problem. Classical data mining utilizing just one data set at a time is insufficient in such a case. It is especially important in biological applications as, for example, in functional genomics where a single data source can often reveal only a certain perspective of the underlying complex biological mechanism. By integrating evidence from multiple data sources, it is possible to obtain more accurate predictions of unknown gene functions. Generally, combining information from the ontologies providing different insights into a problem domain can be very helpful and can lead to the discovery of new knowledge. So, the integration of available evidence from multiple data sources may significantly decrease human efforts in creating useful knowledge representations. This was the goal of the project “DAMIART – Data Mining of heterogeneous data with an Adaptive-Resonance-Theory-based neural network” which intended to solve the latter problem by integrating available data sources and class ontologies.

This paper starts by reporting on the data mining system developed in the project DAMIART. Then we propose a natural extension to this system which should exploit LOD for performance improvement and knowledge extraction. We also discuss a set of objectives this extension should deal with. The Conclusion and Future Work section closes the paper.

2 DAMIART System

The DAMIART project combined multiple data source and multiple class ontology approaches into a single data mining system performing multi-label classification by a neuro-fuzzy classifier. It should lead to the improvement of classification performance and result interpretation because of using complementary domain knowledge extracted from different data sources. The most important tasks of the developed system are hierarchy extraction from multi-labels [3] and concept relation [1], both solved by association analysis. Concept relation implies that relations found between the classes of multiple class ontologies can assist experts in extracting new knowledge from data. For example, if a film can be classified either by its genre into a genre ontology or by the producing company in an ontology of producers, one can find a possible interesting connection between a certain genre and a producing company, specializing in this genre. This potentially useful information can be utilized in many ways, for example, to narrow the huge search space for data mining algorithms or to better interpret the results presented to the user. It was shown that the system was able to discover valuable relationships between class ontologies [2]. Additionally, fuzzy rules extracted from the trained classifier can be used for the plausibility check of discovered association rules. When experts subsequently interact with the system (see Fig. 2), it should be possible to reveal conflicts in the classification rules and to correct them. Finding relations between concepts in our system

is instance-based, which means that they are determined by data only and may change accordingly when the data change.

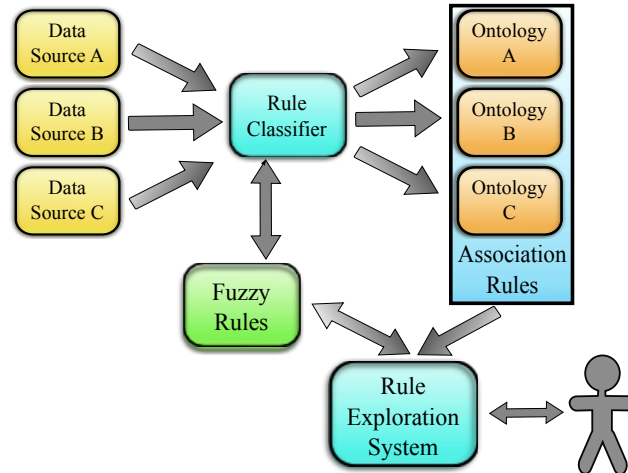


Fig. 2. DAMIART System.

3 Motivation and Objectives

In the focus of the project DAMIART were only multiple class ontologies providing additional information about relationships between the labels found in a training set. Considering a large number of ontologies available also for data attributes, such restriction to class ontologies seems to be inappropriate. To cope with the challenges discussed in Introduction, the DAMIART system can be naturally extended to utilize not only class ontologies, but also ontologies available for data attributes (see Fig. 3). In the above example of the film classification, the data contain short film descriptions, which are then transformed into feature vectors by standard text mining methods. The use of an ontology like Wordnet [20] enables interesting connections between some subsets of words and a certain genre of films, such as e.g. thriller or comedy, to be extracted. We therefore propose to extend the DAMIART system by integrating existing ontological knowledge about attributes. The analysis of state-of-the-art approaches revealed that the proposed extension can be used to solve at least the following tasks:

1. to enrich training data with additional features derived from LOD;
2. to perform feature selection effectively;
3. to further improve classification performance;
4. to enhance the interpretation of fuzzy rules extracted from the classifier;
5. to facilitate understanding of obtained classification results.

It has been already shown in different applications (e.g. [16]) that significant improvement of classification performance can be achieved by the data enrichment

through large Web ontologies like DBpedia [6]. It is important to note that the methods can be diverse: one can either directly incorporate additional features in a dataset or exploit high-level knowledge in order to avoid overfitting by replacing specialized features with more general concepts, e.g. names of certain streets can be replaced with the concept “Street”. Obviously, the use of additional information facilitates the feature selection [10]. It has also been shown in [19] how LOD can be successfully applied to enhancing the interpretation of data mining results.

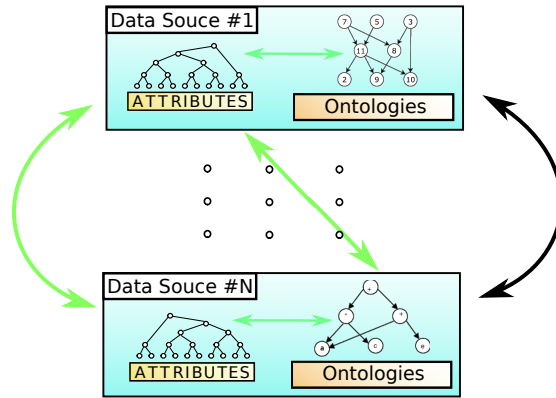


Fig. 3. Extension of the DAMIART System.

4 Conclusion and Future Work

In this paper we have proposed the semantic data mining extension to the DAMIART system. Its implementation will be the subject of future work. Among the benefits expected from the extension is the reduction of data sparseness if a dataset has only few features: Including new features from LOD helps solve this problem. However, the danger of overfitting arises if a dataset already has a lot of features. In this case it is important to select the features that are sufficiently general to represent more specific features of the training data. For this purpose ontological structures of LOD are very useful. It is also expected that the system will be able to produce more accurate results. Additionally, we expect an improvement of interpretability and understandability of the classification results due to better representation of the fuzzy rules extracted from the trained classifier. Moreover, possible new knowledge found by combining different data sources could be used to further update the ontologies, generating a feedback cycle in the data mining process similarly to [5]. The system will have many potential applications such as politics (analysis of the election results), medicine (patient-report analysis), genetics (functional gene classification), and machine translation. An additional point of the future work is evaluation of the proposed extension in one or several application fields.

References

1. Benites, F., Sapozhnikova, E.: Learning different concept hierarchies and the relations between them from classified data. In: Intel. Data Analysis for Real-Life Applications: Theory and Practice, pp. 18–34. IGI Global, Hershey (2012)
2. Benites, F., Simon, S., Sapozhnikova, E.: Mining rare associations between biological ontologies. PLOS ONE 9(1), e84475 (2014)
3. Brucker, F., Benites, F., Sapozhnikova, E.P.: Multi-label classification and extracting predicted class hierarchies. Pattern Recognition 44, 724–738 (2011)
4. d’Amato, C., Fanizzi, N., Esposito, F.: Inductive learning for the semantic web: What does it buy? Semantic Web 1, 53–59 (2010)
5. d’Aquin, M., Kronberger, G., Surez-Figueroa, M.: Combining data mining and ontology engineering to enrich ontologies and linked data. In: Workshop: Knowledge Discovery and Data Mining Meets Linked Open Data (2012)
6. DBpedia, <http://dbpedia.org>
7. Domingue, J., Fensel, D., Hendler, J.A. (eds.): Handbook of Semantic Web Technologies. Springer, Heidelberg (2011)
8. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: An overview. In: LECT NOTES ARTIF INT, pp. 1–34. LNCS (1996)
9. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Magazine 17, 37–54 (1996)
10. Jeong, Y., Myaeng, S.H.: Feature selection using a semantic hierarchy for event recognition and type classification. In: Sixth Int. Joint Conf. on Natural Language Processing, pp. 136–144. Asian Federation of Natural Language Processing, Nagoya, Japan (October 2013)
11. Lavrač, N., Vavpetič, A., Soldatova, L., Trajkovski, I., Novak, P.K.: Using ontologies in semantic data mining with SEGS and g-SEGS. In: 14th Int. Conf. on Discovery science, pp. 165–178. DS’11, Springer, Heidelberg (2011)
12. Liu, H.: Towards semantic data mining. In: 9th Int. Semantic Web Conf. (2010)
13. Moss, L., Sleeman, D.H., Sim, M., Booth, M., Daniel, M., Donaldson, L., Gilhooly, C.J., Hughes, M., Kinsella, J.: Ontology-driven hypothesis generation to explain anomalous patient responses to treatment. Knowl.-Based Syst. 23, 309–315 (2010)
14. Ni, X., Sun, J.T., Hu, J., Chen, Z.: Cross lingual text classification by mining multi-lingual topics from wikipedia. In: WSDM ’11 fourth ACM international conference on Web search and data mining, pp. 375–384. ACM, New York (2011)
15. Novak, P.K., Vavpetič, A., Trajkovski, I., Lavrač, N.: Towards semantic data mining with g-SEGS. In: 13th International Multiconference Information Society (IS 2010), pp. 173–176 (2010)
16. Paulheim, H.: Exploiting linked open data as background knowledge in data mining. In: Int. Workshop on Data Mining on Linked Data, with Linked Data Mining Challenge at ECMLPKDD 2013, pp. 1–10 (2013)
17. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: 17th International Conference on World Wide Web, pp. 91–100. ACM, New York (2008)
18. Singhal, A., Kasturi, R., Sivakumar, V., Srivastava, J.: Leveraging web intelligence for finding interesting research datasets. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology 1, 321–328 (2013)
19. Tiddi, I.: Explaining data patterns using background knowledge from linked data. In: ISWC-DC, pp. 56–63 (2013)
20. WordNet, <http://wordnet.princeton.edu/>

Discovering Periodic Patterns in System Logs^{*}

Marcin Zimniak¹, Janusz R. Getta², and Wolfgang Benn¹

¹ Faculty of Computer Science, TU Chemnitz, Germany
{marcin.zimniak, benn}@cs.tu-chemnitz.de

² School of Computer Science and Software Engineering,
University of Wollongong, Australia
jrg@uow.edu.au

Abstract. Historical information stored in the software system logs, audit trails, traces of user applications, etc. can be analysed to discover the patterns in periodic variations of levels and structures of the past workloads. These patterns allow for the estimation of intensities and structures of the future workloads. The correctly anticipated future workloads are used to improve performance of software systems through appropriate allocation of computing resources and through restructuring of associated system support. This work defines a concept of *periodic pattern* and presents the algorithms that find the periodic patterns in the traces of elementary and complex operations on data recorded in the system logs.

1 Introduction

The typical approaches to performance tuning of software systems either find the software components that have significant impact on performance or find a group of software components whose simultaneous processing contributes to the performance bottlenecks [1]. Optimization of software components restructures associated system support. It relocates data containers to the faster storage devices, runs processes on the faster processors, adds more resources and computational power, increases the priorities of performance critical processes, etc. Optimization through elimination of bottlenecks restructures the functionality of software components involved in the collisions during their simultaneous processing.

Automated performance tuning of software systems [2] implements an “observer” module that automatically changes a level of system support or resolves the collisions whenever it is necessary. An important factor is the ability of an “observer” to anticipate the low workload times when re-balancing of system support or elimination of bottlenecks can be done. An important question is

^{*} Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

whether it is possible to predict the future characteristics of workloads from information about the past behaviour of a software system. A promising approach is to utilize the periodic repetitions of data processing caused by the “repetitive nature” of real world processes. For example, students enrolling courses at the beginning of each session, accountants processing financial data at the end of financial year, footballers playing games every week, etc. trigger the repetitions of the same data processing cycles. An objective of this work is to provide efficient algorithms for discovery of periodic patterns in the traces of elementary and complex operations on data recorded in the system logs. We assume an abstract model of computations where a system log is represented by a sequence of nested n-ary operations on data later on called a *workload trace*. A workload trace is sequence of groups of operations computed in the same periods of time. A workload trace is “reduced” by elimination of all nested operations that provide the arguments for only one higher level operation. Then, the processing patterns of “children” operations are the same as the patterns of “parent” operations.

The paper is organized in the following way. The next section refers to the previous works in the related research areas. Section 3 defines the basic concepts and a model of workload trace. A concept of periodic pattern in a system workload is defined in Section 4 and discovering of elementary periodic patterns is explained in Section 5. Section 6 concludes the paper.

2 Related work

Data mining techniques that inspired the works on periodic patterns came from the works on mining association rules [3] and later on from mining frequent episodes [4] and its extensions on mining complex events.

The problem seems to be very similar to a typical periodicity mining in time series [5], where analysis is performed on the long sequences of elementary data items discretized into a number of ranges and associated with the timestamps. In our case, the input data is a sequence of complex data processing statements, e.g. SQL statements and due to its internal structure cannot be treated in the same way as analysis of elementary data elements in time series or genetic sequences.

The recent approaches, which addressed full periodicity, partial periodicity, perfect and imperfect periodicity [6], and asynchronous periodicity [7] are all based on fixed size and adjacent time units and fixed length of discovered patterns.

Our problem is also similar to a problem of mining cyclic association rules [8] where an objective is to find the periodic executions of the largest sets of items that have enough support.

Invocation of operation on data along the various points in time can be easily described by temporal predicates within a formal scope of temporal deductive database systems [9]. The reviews of data mining techniques based on analysis of ordered set of operations on data performed by the user applications are available in [10], [11]. The model of periodicity considered in this paper is consistent with the model proposed in [12].

3 Basic concepts

The operations processed by the *software components* c_1, \dots, c_n are recorded in an *operation trace*. A *trace* of a software component c_i is a finite sequence of pairs $\langle p_{i_1}:t_{i_1}, \dots, p_{i_n}:t_{i_n} \rangle$ where each t_{i_j} is a timestamp when an operation p_{i_j} has been processed.

A *system trace* A is a sequence of interleaved trails of software components processed in a certain period of time. For example, a sequence $\langle p_{i_1}:t_{i_1}, p_{j_1}:t_{j_1}, p_{i_2}:t_{i_2}, p_{j_2}:t_{j_2} \rangle$ is a sample system trace from the processing of software components c_i , and c_j .

Let $\langle t_{start}, t_{end} \rangle$ be a period of time over which a system trace is recorded. A *time unit* is a pair $\langle t, \tau \rangle$ where t is a start point of a unit and τ is a length of the unit. A nonempty sequence U of n disjoint time units $\langle t^{(i)}, \tau^{(i)} \rangle$ $i = 1, \dots, n$ over $\langle t_{start}, t_{end} \rangle$ is any sequence of time units that satisfies the following properties: $t_{start} \leq t^{(1)}$ and $t^{(i)} + \tau^{(i)} \leq t^{(i+1)}$ and $t^{(n)} + \tau^{(n)} \leq t_{end}$.

A multiset M is defined as a pair $\langle S, f \rangle$ where S is a set of values and $f : S \rightarrow N^+$ is a function that determines multiplicity of each element in S and N^+ is a set of positive integers. In the rest of this paper we shall denote a multiset $\langle \{T_1, \dots, T_m\}, f \rangle$ where $f(T_i) = k_i$ for $i = 1, \dots, m$ as $(T_1^{k_1} \dots T_m^{k_m})$. We shall denote an empty multiset $\langle \emptyset, f \rangle$ as \emptyset and we shall abbreviate a multiset (T^k) to T^k and T^1 to T .

A *workload trace* of an operation T is a sequence W_T of $|U|$ multisets of operations such that $W_T[i] = \langle \{T\}, f_i \rangle$ and $f_i(T) = |T.timestamp(i)| \forall i = 1, \dots, |U|$ i.e. $f_i(T)$ is equal to the total number of times an operation T was processed in the i -th time unit $U[i]$.

Let \mathbf{T} be a set of all operations obtained from a system trace A . A *workload trace* of A is denoted by W_A and $W_A[i] = \biguplus_{T \in \mathbf{T}} W_T[i], \forall i = 1, \dots, |U|$, i.e. it is a sum of workload traces of all operations processed in U .

4 Periodic patterns

A *periodic pattern* is a tuple $\langle \mathcal{T}, U, b, p, e \rangle$ where \mathcal{T} is a nonempty sequence of multisets of operations, U is a sequence of disjoint time units, $b \geq 1$ is a number of time unit in U where the repetitions of \mathcal{T} start, $p \geq 1$ is the total number of time units after which processing of \mathcal{T} is repeated in every processing cycle, $e > b$ is a number of time unit in U where the processing of \mathcal{T} is performed for the last time. A sequence of multisets \mathcal{T} may contain one or more empty multisets. The positional parameters b, p , and e of a periodic pattern must satisfy a property $(e - b) \bmod p = 0$. A value $c = \frac{e-b}{p} + 1$ is called as the *total number of cycles* in the periodic pattern.

Let \mathcal{T}_{ext} be a sequence of multisets of operations obtained from \mathcal{T} and extended on the right with $e - b$ empty multisets. Then, a *workload trace of a periodic pattern* $\langle \mathcal{T}, U, b, p, e \rangle$ with the total number of cycles $c = \frac{e-b}{p} + 1$ is a sequence $W_{\mathcal{T}}$ of $e - b + |\mathcal{T}|$ multisets of operations such that $W_{\mathcal{T}}[i] =$

$\mathcal{T}_{ext}[g(i)] \uplus \mathcal{T}_{ext}[g(i - p)] \uplus \mathcal{T}_{ext}[g(i - 2 * p)] \uplus \dots \mathcal{T}_{ext}[g(i - (c - 1) * p)]$ for $i = 1, \dots, e - b + |\mathcal{T}|$ where a function g is computed such that **if** $x > 0$ **then** $g(x) = x$ **else** $g(x) = x + e - b + |\mathcal{T}|$.

For example, $\langle \emptyset TV, U, 1, 1, 3 \rangle$ is a periodic pattern where processing of a sequence of operations $\emptyset TV$ starts in the time units 1, 2, and 3 and its workload trace is a sequence of multisets $\emptyset T(VT)(VT)V$. The periodic pattern has 3 cycles.

Let $|W_{\mathcal{T}}|$ be the total number of elements in $W_{\mathcal{T}}$. Let v be the total number of elements in $W_{\mathcal{T}}$ such that $W_{\mathcal{T}}[i] \subseteq W_A[b + i - 1]$ for $i = 1, \dots, e - b + |\mathcal{T}|$. Then, we say that a periodic pattern $\langle \mathcal{T}, U, b, p, e \rangle$ is *valid in a system trace A with a support* $0 < \sigma \leq 1$ if $W_{\mathcal{T}}[1] \subseteq W_A[b]$ and $W_{\mathcal{T}}[e - b + |\mathcal{T}|] \subseteq W_A[e + |\mathcal{T}|]$ and $\sigma \leq v/|W_{\mathcal{T}}|$.

For example, a periodic pattern $\langle (T^2V)\emptyset W, U, 2, 3, 8 \rangle$ has a workload trace $(T^2V)\emptyset W(T^2V)\emptyset W(T^2V)\emptyset W$. The pattern is valid in a system trace A with support $\sigma = 1$ if every element of its workload trace is included in a workload trace W_A from position 2 to position 10.

A periodic pattern $\langle \mathcal{T}, U, b, p, e \rangle$ such that $\mathcal{T} = T^{i_1} \dots T^{i_n}$ where $i_k, n \geq 1$ and $p \geq |\mathcal{T}|$ is called as a *homogeneous periodic pattern*. For example, a periodic pattern $\langle T^7 T^2 T^{11}, U, 10, 4, 22 \rangle$ is a homogeneous periodic pattern, which has four cycles.

5 Discovering periodic patterns

A method for discovering periodic patterns in database audit trails proposed in [12] iterates over the dimensions of syntax trees of SQL statements retrieved from an audit trail and the dimensions of positional parameters b , p , and e . The algorithm finds only homogeneous periodic patterns such that $\mathcal{T} = T^k$ and its computational complexity is approximately $O(k * n^3)$ where $0 < k < 1/8$ and n is the total number of time units of an audit trail.

An approach to mining periodic patterns proposed in this paper is based on two algorithms. The first algorithm finds in a workload trace W_A the longest subsequence $W_{\mathcal{T}}$ of nonempty multisets, which is included in the largest number of times in the trace W_A . Then, $W_{\mathcal{T}}$ is passed to the second algorithm to generate the candidate periodic patterns and to return the candidate pattern with the highest support to the first algorithm. Next, the first algorithm saves the periodic pattern, it removes from W_A a workload trace of the pattern, and it repeats itself until it is possible to find a new $W_{\mathcal{T}}$ which has at least two separate subsequences.

Algorithm 1

Let \mathbf{T} be a set of all operations included in a workload trace W_A . The following algorithm iteratively performs the following steps over the operations $T \in \mathbf{T}$. At the beginning a set of periodic patterns \mathcal{P} is empty.

- (1) We transform $W_{\mathcal{T}}$ into a sequence of numbers (words with length equal to $|U|$), $\langle f_i(T) \rangle$, such that $f_i(T) \in N_0, \forall i = 1, \dots, |U|$, $W'_{\mathcal{T}} := W_{\mathcal{T}}$.

- (2) If there are no empty multisets in W'_T then we transform W'_T as follows, $W'_T := W'_T - (\text{workload trace of } \langle T, 1, 1, |U| \rangle)$ the smallest number of $k_{i_{min}}$ times such that in the result of transformation we obtain at least one empty multiset, i.e. there is at least one zero in a sequence (word) $\langle f'_i(T) \rangle$. We add a periodic pattern to a set \mathcal{P} , $\mathcal{P} := \mathcal{P} \cup \langle T^{k_{i_{min}}}, 1, 1, |U| \rangle$.
- (3) We find in W'_T all longest sub-sequences $\{W_i(W'_T)\}$ in a sense of inclusion of multisets over their components and such that there exists the largest number of the same sub-sequences whose length is equal to $\max \| \langle f'_{i_j}(T), \dots, f'_{i_k}(T) \rangle \|$.
- (4) For each individual $W_i(W'_T)$ we apply **Algorithm 2** described below and from all periodic patterns found we pick a pattern with the largest value of support σ .
- (5) If from a step (4) we get pp_{hom} then $\mathcal{P} := \mathcal{P} \cup pp_{hom}$ and $W'_T := W'_T - (\text{workload trace of } pp_{hom})$ and we return to step (3). If a step (4) returns no solutions then we progress to the next step.
- (6) It is possible to remove from any valid periodic pattern any leading or trailing sequence of T and still get a periodic pattern valid in the same workload trace. We search W'_T for shorter homogeneous periodic patterns pp_{hom}' . We insert each pp_{hom}' found into \mathcal{P} and $W'_T := W'_T - (\text{workload trace of } pp_{hom}')$. We return to step (3).

Algorithm 2

An input to the second algorithm is a workload trace W_T , a given sequence of multisets of operations \mathcal{T} , the locations of the first (f) and the last (t) instances of \mathcal{T} in W_T . The parameters of a candidate periodic pattern in W_T must satisfy the following linear Diophantine equation:

$$c * |\mathcal{T}| + (c - 1) * d = e - b + |\mathcal{T}| \quad (1)$$

where d is a distance between the instances of \mathcal{T} in the pattern and c is the total number of cycles in the pattern. To solve the equation we assume that $b = f$, $e = t$. Let d_{min} (d_{max}) be the shortest (the longest) distance between any two locations where \mathcal{T} is included in W_T . The algorithm consists of the following steps.

- (1) We make a set of candidate periodic patterns P empty.
- (2) We iterate over the values of $d = d_{min}, d_{min} + 1, \dots, d_{max}$.
- (2.1) For a given value of d we find the following values of c and r :

$$c = \frac{e - b}{|\mathcal{T}| + d} + 1 \quad (2)$$

$$r = (e - b) \text{ mod } (|\mathcal{T}| + d) \quad (3)$$

- (2.2) Let $p = |\mathcal{T}| + d$. We create the periodic patterns $\langle \mathcal{T}, b, p, b + p * (c - 1) \rangle$, $\langle \mathcal{T}, b + 1, p, b + p * (c - 1) + 1 \rangle$, \dots , $\langle \mathcal{T}, b + r, p, b + p * (c - 1) + r \rangle$. We append the periodic patterns found to a set of candidate periodic patterns P . If available we pick the next value of d and we return to step (2.1).
- (3) A set of candidate homogeneous periodic patterns P is returned to the first algorithm.

6 Summary and conclusions

Discovering the complex periodic patterns in the system logs is a difficult and time consuming task. This work defines a concept of *periodic pattern* and shows how to find the periodic patterns in the the system logs. An approach described here shows that it is easier to find the simple periodic patterns and later on to compose them into the complex ones instead of directly searching for all complex patterns. The discovered periodic patterns can be used to model future workload after the old applications are replaced with the new ones or the new applications are added to a system. It is also easier to reconcile the new audit trails with the collections of periodic patterns discovered from the previous system logs than to integrate the complete logs.

References

1. Osterhage, W.: Computer Performance Optimization. Springer-Verlag (2013)
2. Bruno, N., ed.: Automated Physical Database Design and Tuning. CRC Press Taylor and Francis Group (2011)
3. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of The 1993 ACM SIGMOD Intl. Conf. on Management of Data. (1993) 207–216
4. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery* **1** (1997) 259–289
5. Rasheed, F., Alshalalfa, M., Alhajj, R.: Efficient periodicity mining in time series databases using suffix trees. *IEEE Transactions on Knowledge and Data Engineering* **23**(1) (2011) 79–94
6. Huang, K.Y., Chang, C.H.: SMCA: A general model for mining asynchronous periodic patterns in temporal databases. *IEEE Transactions on Knowledge and Data Engineering* **17**(6) (2005) 774–785
7. Yang, J., Wang, W., Yu, P.S.: Mining asynchronous periodic patterns in time series data. *IEEE Trans. on Knowl. and Data Eng.* **15**(3) (March 2003) 613–628
8. Özden, B., Ramaswamy, S., Silberschatz, A.: Cyclic association rules. In: Proceedings of the Fourteenth International Conference on Data Engineering. (1998) 412–421
9. Baudinet, M., Chomicki, J., Wolper, P.: Temporal deductive databases (1992)
10. Laxman, S., Sastry, P.S.: A survey of temporal data mining. *Sadhana, Academy Proceedings in Engineering Sciences* **31**(2) (2006) 173–198
11. Roddick, J.F., Society, I.C., Spiliopoulou, M., Society, I.C.: A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering* **14** (2002) 750–767
12. Zimniak, M., Getta, J., Benn, W.: Deriving composite periodic patterns from database audit trails. In: The 6th Asian Conference on Intelligent Information and Database Systems. (2014) 310–321

On the Predictability of Talk Attendance at Academic Conferences

(Extended Abstract*)

Christoph Scholz, Jens Illig, Martin Atzmueller, and Gerd Stumme

University of Kassel, ITeG Research Center
Knowledge and Data Engineering Group
Wilhelmshöher Allee 73, 34121 Kassel, Germany

{scholz, illig, atzmueller, stumme}@cs.uni-kassel.de

1 Introduction

Academic conferences facilitate scientific exchange, collaboration and innovation, e. g., fostered by social contacts and interesting talks. A major task for conference attendees is the selection of talks relevant to their research. Conference guidance systems such as Conference Navigator [10] and CONFERATOR [2, 3], support this with the possibility of creating a personalized schedule. Picking talks manually, however, may become complex due to the large amount of available talks at a conference. In such contexts, recommendation components of conference guidance systems can support their users by presenting suggestions of talks which the system determined as most interesting for the respective user. Such recommendations influence the user’s decision e. g., due to recommended talks which were otherwise not considered.

In this paper, we focus on the predictability of real talk attendances, i. e., we try to find models imitating the actual decision process without recommendation influence. We study and discuss the predictability of such talk attendances using real-world face-to-face contact data and user interest models extracted from the users’ previous publications. Specifically, for our evaluation we use real-world talk attendance data which was collected using the CONFERATOR system that applies RFID technology developed by the Sociopatterns consortium. Given such RFID data and collected content information of scientific papers, we derive a set of social interaction networks [1]. Based on these, we investigate the potential of social contact information and content-similarity for predicting real-world talk attendance decisions. In particular, we analyze the potential of combining different information sources for improving the overall prediction quality. We find that contact and similarity networks achieve comparable results, and that combining these networks helps to a limited extent to improve the prediction quality.

* This extended abstract summarizes the paper [8]: Scholz, C., Illig, J., Atzmueller, M., Stumme, G.: On the Predictability of Talk Attendance at Academic Conferences. In: Proc. Hypertext. ACM Press, New York, NY, USA (2014). An extended version can be found in [9]. *Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes.* In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

2 Experiments and Results

For capturing social interaction networks of face-to-face proximity, participants of the 22nd ACM Conference on Hypertext and Hypermedia 2011 (HT 2011) were offered to wear active RFID tags; these detect other active RFID tags within a range of up to 1.5 meters. Table 1 provides a summary on the characteristics of the collected dataset, with diameter (d), the average aggregated contact-duration (AACD) and the average path length (APL). For more details, cf. [5–7]. For analysis, we focused on the 14 parallel talks at HT 2011; we observed 359 visited talks from 53 conference participants. We also considered the content of all papers. For each conference participant, we further crawled all papers listed in DBLP since 2006, for a total of 707 papers.

| | HT 2011 |
|---------------|---------|
| $ V $ | 68 |
| $ E $ | 698 |
| $Avg.Deg.(G)$ | 20.53 |
| $APL(G)$ | 1.76 |
| $d(G)$ | 4 |
| AACD | 529 |

Table 1. Collected dataset at HT 2011.

In our experiments, we studied the influence of face-to-face contacts and user interests on the talk attendance. We showed, that the probability of two participants attending the same talk is nearly random, if there exists no contact before that talk; conversely, that probability is significantly increased if there exists a contact in the break before the talk. We also analyzed the influence of user interests based on the contents of the crawled papers: Prediction based on user-interest alone achieves better results than prediction based solely on face-to-face contact data. Furthermore, we showed that a combination of different networks helps to further improve the prediction accuracy. Also, the combination of all information belonging to one session, i. e., merging the presenter nodes, significantly improves prediction accuracy. For future work, we aim to integrate and exploit information from further social interaction networks, cf. [1] and to consider description-oriented approaches, e. g., [4], for further improving the predictions.

References

1. Atzmueller, M.: Data Mining on Social Interaction Networks. *JDMDH* 1 (2014)
2. Atzmueller, M., Becker, M., Kibanov, M., Scholz, C., Doerfel, S., Hotho, A., Macek, B.E., Mitzlaff, F., Mueller, J., Stumme, G.: Ubicon and its Applications for Ubiquitous Social Computing. *New Review of Hypermedia and Multimedia* 20(1), 53–77 (2014)
3. Atzmueller, M., Benz, D., Doerfel, S., Hotho, A., Jäschke, R., Macek, B.E., Mitzlaff, F., Scholz, C., Stumme, G.: Enhancing Social Interactions at Conferences. *it+ti* 3, 1–6 (2011)
4. Atzmueller, M., Lemmerich, F.: VIKAMINE - Open-Source Subgroup Discovery, Pattern Mining, and Analytics. In: *Proc. ECML/PKDD*. Springer, Heidelberg, Germany (2012)
5. Macek, B.E., Scholz, C., Atzmueller, M., Stumme, G.: Anatomy of a Conference. In: *Proc. Hypertext*. pp. 245–254. ACM Press, New York, NY, USA (2012)
6. Scholz, C., Atzmueller, M., Barrat, A., Cattuto, C., Stumme, G.: New Insights and Methods For Predicting Face-To-Face Contacts. In: *Proc. ICWSM* (2013)
7. Scholz, C., Atzmueller, M., Stumme, G.: Predictability of Evolving Contacts and Triadic Closure in Human Face-to-Face Proximity Networks. *SNAM* 4(217) (2014)
8. Scholz, C., Illig, J., Atzmueller, M., Stumme, G.: On the Predictability of Talk Attendance at Academic Conferences. In: *Proc. Hypertext*. ACM Press, New York, NY, USA (2014)
9. Scholz, C., Illig, J., Atzmueller, M., Stumme, G.: On the Predictability of Talk Attendance at Academic Conferences. *CoRR* abs/1407.0613 (2014)
10. Wongchokprasitti, C., Brusilovsky, P., Para, D.: Conference Navigator 2.0: Community-Based Recommendation for Academic Conferences. In: *Proc. Workshop SRS, IUT'10* (2010)

Identifying and Analyzing Researchers on Twitter

Asmelash Teka Hadgu and Robert Jäschke

L3S Research Center, Appelstraße 4, 30167 Hannover, Germany
teka@L3S.de, jaeschke@L3S.de

Abstract Twitter is a communication platform, a social network, and a system for resource sharing. For scientists, it offers an opportunity to connect with other researchers, announce calls for papers and the like, communicate and discuss – basically: stay up-to-date. However, the exponential growth of information in society does not exclude social media like Twitter: an abundant number of users court on one’s attention which leads to the question of how (young) researchers can focus on the essential users and tweets?

The classical approach in science to filter information is peer review: only information that is considered to be novel, sound, and significant by experts in the respective field is published. Currently, such a process is at most implemented manually: researchers can subscribe individually to other researcher’s feeds by following them. However, there is no ‘directory’ of scientists on Twitter and finding feeds of experts in a specific discipline or area of interest is cumbersome.

Furthermore, the trend to consider visibility of scientific articles in the social web as a possible (and immediate) alternative or complement to citation counts (with services like Altmetric¹ that provide counts for how often a scientific article has been mentioned on Twitter and other social networks) necessitates the need for peer-review-like mechanisms for the social web. Simple approaches purely based on the popularity of users, tweets, or URLs do not work as a tool for scientists to discover relevant research(ers), since popularity on the social web is fundamentally a matter of the crowd of non-scientists. Articles that are popularized by the media – often independent of their scientific significance – get superior attention compared to other, more important works. Consider the Ig Nobel Prizes,² whose winning (scientific) publications get quite some attention on the social web, e.g., the URL³ of the winner of the 2012 physics prize has been mentioned in more than 230 tweets.⁴ Enabling

Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org/>

¹ <http://www.altmetric.com/>

² <http://www.improbable.com/ig/>

³ <http://prl.aps.org/abstract/PRL/v108/i7/e078101>

⁴ <http://topsy.com/trackback?url=http%3A%2F%2Fprl.aps.org%2Fabstract%2FPRL%2Fv108%2Fi7%2Fe078101>

users (and in particular researchers) to access the scientists' perspective in the social web and considering only tweets from physicists would provide a different and likely better picture.

Existing Twitter directories like Wefollow⁵ rely on users' initiative to register and reveal their interests. This clearly limits the set of available profiles, since professionals have limited time and there is no immediate benefit for registration. Therefore, providing an automatically curated directory of scientists would simplify expert finding and the provision of topic-relevant feeds authored by peers. This approach requires to first identify scientists on Twitter and then classify their discipline, topics of interest, and expertise. Since only little is known about scientists on Twitter, such an endeavor should be accompanied by further steps to understand how Twitter is used by them.

In this work we present an approach for the identification and classification of scientists on Twitter together with an empirical analysis of researchers from computer science found on Twitter. We take a pragmatic approach on which users we regard as 'scientists': users being interested in the topics of the target discipline and having similar, Twitter-based features like users that have published scientific papers. We start with a list of seeds that are highly-relevant for the discipline of interest and use it to build and augment a set of candidate users that are likely scientists. For a subset of the candidates that we can match to ground-truth data from a digital library, we build a model for the classification of scientists. We can show that the model is very accurate and use it to classify all of our candidates. Both sets of users (matched and classified) allow us to perform an empirical analysis of scientists on Twitter.

The main contributions of this work are

- a complete framework for discipline-specific researcher classification on Twitter using a small set of seeds only,
- an automatic approach for the generation of ground-truth data by combining different data sources,
- an empirical analysis of computer scientists that are using Twitter, and
- the provision of the used datasets.⁶

The results were published in A.T. Hadgu and R. Jäschke: Identifying and analyzing researchers on Twitter. In *Proceedings of the 2014 ACM Conference on Web Science, WebSci '14*, pages 23–30, New York, NY, USA, 2014. ACM. DOI:10.1145/2615569.2615676

⁵ <http://wefollow.com/>

⁶ <https://github.com/L3S/twitter-researcher>

Evaluation des Konfigurationsraumes von Kohärenzmaßen für Themenmodelle

Extended Abstract

Michael Röder^{1,3}, Andreas Both³ und Alexander Hinneburg²

¹ Universität Leipzig

² Martin-Luther-Universität Halle-Wittenberg

³ R&D, Unister GmbH, Leipzig

{michael.roeder|andreas.both}@unister.de,
hinneburg@informatik.uni-halle.de

Eine Menge von Aussagen oder Fakten wird als kohärent angesehen, wenn sie sich gegenseitig unterstützen. Deshalb kann eine kohärente Faktenmenge gut in einem Kontext interpretiert werden, der alle oder die meisten Fakten umfasst. Ein Beispiel für eine solche Faktenmenge ist “das Spiel ist eine Mannschaftssportart”, “das Spiel wird mit einem Ball gespielt”, “das Spiel erfordert große physische Anstrengungen”, die z.B. im Kontext von Fußball einen Sinn ergibt. Eine offene Forschungsfrage ist, wie die Kohärenz einer Faktenmenge quantifiziert werden kann [2]. In Arbeiten aus dem Bereich der Wissenschaftsphilosophie wurden Maße vorgeschlagen, die als Funktionen von Verbund- und Randwahrscheinlichkeiten formalisiert wurden, welche den Fakten zugeordnet sind. Bovens und Hartmann [2] diskutieren viele Beispiele, die zu einer Menge von notwendigen Bedingungen führen, die ein solches Maß erfüllen soll. Die Arbeiten in diesem Bereich beschäftigen sich vor allem mit verschiedenen Schemata, die das Zusammenhängen und zueinander Passen der einzelnen Fakten einer größeren Faktenmenge abschätzen. Beispiele für solche Schemata sind (1) vergleiche jeden einzelnen Fakt mit dem Rest aller verbleibenden Fakten, (2) vergleiche alle Paare von Fakten miteinander und (3) vergleiche alle disjunkten Teilmengen der Fakten miteinander. Diese theoretischen Arbeiten aus dem Bereich der Wissenschaftsphilosophie – siehe [4] für einen Überblick – sind in der Informatik weitgehend unbekannt.

Das Interesse an Kohärenzmaßen entstand im Bereich Text Mining, weil unüberwachte Lernmethoden, wie z.B. Themenmodelle, keine Garantie dafür geben, dass ihre Ausgabe interpretierbar ist. Themenmodelle lernen unüberwacht Themen, die üblicherweise als Menge von wichtigen Wörtern repräsentiert werden. Dies ist eine attraktive Methode, um unstrukturierte Textdaten mit einer Struktur zu versehen. In der grundlegenden Arbeit von Newman et al. [7] werden Kohärenzmaße vorgeschlagen, die bewerten, wie verständlich durch Wortmengen repräsentierte Themen sind. Die vorgeschlagenen Maße behandeln Wörter als Fakten und nutzen das Schema, das paarweise alle Wörter vergleicht. Für die Evaluationen in [7] werden durch Menschen erstellte Themen-Rankings verwendet. Die Auswertungen zeigten, dass Maße, die auf Statistiken über das gemeinsame Auftreten von Wörtern beruhen, stärker mit men-

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

schlichen Bewertungen korrelieren als andere Maße, die auf WordNet und ähnlichen semantischen Ressourcen beruhen. Anschließend empirische Arbeiten zu Themenkohärenzmaßen [6,9,5] schlugen eine Vielzahl von weiteren Maßen vor, die auf Wortstatistiken basieren. Diese Maße unterscheiden sich in mehreren Details wie der Definition, Normalisierung und Zusammenfassung von Wortstatistiken sowie den Referenzkorpora zur Erstellung der Statistiken. Weiterhin wurde kürzlich in [1] eine neue Methode basierend auf Kontextvektoren vorgeschlagen. Die Beiträge zur Kohärenz aus Wissenschaftsphilosophie und Text Mining sind komplementär. Während in wissenschaftsphilosophischen Beiträgen Schemata zum Vergleichen von Fakten vorgeschlagen werden, entwickeln die Text-Mining-Beiträge Methoden zum Schätzen und Zusammenfassen von Wortwahrscheinlichkeiten. Es fehlt jedoch eine systematische Evaluation der Methoden aus beiden Bereichen und deren noch unerforschten Kombinationen.

Menschliche Themen-Rankings dienen als Goldstandard für die Evaluation von Kohärenz, die jedoch aufwendig zu erstellen sind. Unsere empirische Evaluation nutzt alle drei öffentlich verfügbaren Quellen solcher Rankings: (1) die Daten von Chang et al. [3], die von Lau et al. [5] für Kohärenzevaluation vorbereitet wurden, (2) Aletras und Stevenson [1] und (3) Rosner et al. [8]. Die Beiträge dieser Arbeit sind: erstens, wir schlagen einen vereinheitlichenden Rahmen vor, der einen Konfigurationsraum aufspannt, der alle bekannten Kohärenzmaße und die Kombinationen der einzelnen Ideen der Ansätze enthält. Zweitens, der Konfigurationsraum wird systematisch durchsucht und alle Kohärenzmaße, bekannte und bisher nicht bekannte, werden auf den verfügbaren Benchmark-Daten evaluiert. Die Ergebnisse zeigen, dass eine bisher nicht bekannte Kombination von Ideen bisheriger Ansätze deutlich stärker mit menschlichen Bewertungen korreliert. Abschließend diskutieren wir Anwendungen von Kohärenzmaßen, die über Themenmodelle hinausgehen.

Literatur

1. Aletras, N., Stevenson, M.: Evaluating topic coherence using distributional semantics. In: Proc. of the 10th Int. Conf. on Computational Semantics (IWCS'13). pp. 13–22. (2013)
2. Bovens, L., Hartmann, S.: Bayesian Epistemology. Oxford University Press (2003)
3. Chang, J., Gerrish, S., Wang, C., Boyd-graber, J.L., Blei, D.M.: Reading tea leaves: How humans interpret topic models. In: Advances in Neural Information Processing Systems 22, pp. 288–296. (2009)
4. Douven, I., Meijs, W.: Measuring coherence. *Synthese* 156(3), 405–425 (2007)
5. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: Proc. of the European Chapter of the Association for Computational Linguistics (2014)
6. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. pp. 262–272. (2011)
7. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: Human Language Technologies: The 2010 Annual Conf. of the North American Chapter of the Association for Computational Linguistics. pp. 100–108. (2010)
8. Rosner, F., Hinneburg, A., Röder, M., Nettle, M., Both, A.: Evaluating topic coherence measures. CoRR abs/1403.6397 (2014), <http://arxiv.org/abs/1403.6397>
9. Stevens, K., Kegelmeyer, P., Andrzejewski, D., Buttler, D.: Exploring topic coherence over many models and many topics. In: Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 952–961. (2012)

Contextual Entity Resolution Approach for Genealogical Data

Hossein Rahmani¹, Bijan Ranjbar-Sahraei¹, Gerhard Weiss¹, and Karl Tuyls²

¹Maastricht University, PO Box 616, Maastricht 6200 MD, The Netherlands
{h.rahmani,b.ranjbarsahraei,gerhard.weiss}@maastrichtuniversity.nl

²University of Liverpool, Ashton Building, Liverpool L69 3BX, United Kingdom
k.tuyls@liverpool.ac.uk

Abstract. Due to huge amount of inaccurate information and different types of ambiguity in the available digitized genealogical data, applying Entity Resolution techniques for determining the records referring to the same entity should be considered as the first and still very important step in analysis of this type of data. Traditional methods, use a standard string similarity measure to calculate the similarity among references, neglecting the contextual information available for each reference, and then introduce the most similar pairs as matches. In this paper, first, we introduce a novel blocking strategy to reduce the number of potential candidate pairs. Second, we propose a contextual similarity measure which not only considers the string similarity among references but also contextual information available for them. Third, we evaluate our proposed method extensively from different perspectives and among many discussed patterns, the “early child death” pattern discovered to be prominent.

Keywords: Entity Resolution, Contextual Similarity, Genealogy

1 Problem Definition

The work discussed in this paper has been developed as part of a larger project, the MISS¹ (Mining Social Structures from Genealogical Data) Project, which is funded by the NWO (Netherlands Organisation for Scientific Research) Association. MISS project uses the historical certificates of BHIC² (Brabants Historical Information Center) to unravel the genealogical connections and also mine the social structures from a prosopographical [8] point of view.

There are three important certificate types used in this project, namely “Birth”, “Death” and “Marriage”. Section 2 describes these certificate types in

Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

¹ <http://swarmlab.unimaas.nl/catch/>

² <http://www.bhic.nl>

detail. The digitized version of these certificates, however, is not at all flawless; many names are duplicate, have several alternative spellings, or even contain mistakes. The main challenge in this project is to find an approach which can resolve when two references refer to same entity in spite of errors and inconsistencies in the input certificates. We model this project as a system which takes large number of error-prone and inconsistent certificates as input and as an output, generates the graph of individuals in which each node represents an entity and each edge shows the family relationship among two connected entities. Two main goals of this project are 1) Detecting and eliminating duplicate references referring to same entity (Known in literature in many different ways such as Record Linkage [12, 18], the Merge/Purge problem [6], Duplicate Detection [10, 16], Hardening Soft Databases [1], Reference Matching [9] and Entity Resolution [5, 4]), and 2) Re-constructing family relationships among individuals.

2 Input Data

We consider three certificate types “Birth”, “Death” and “Marriage” as input of our system. Table 1 shows the considered features for each certificate type. As shown in Table 1 Birth certificates include 3 individual references (i.e., child, father and mother). The Death certificates include 4 individual references (i.e., deceased, father, mother and relative of deceased). Finally, the Marriage certificates include 6 references (i.e., groom, bride and father and mother of each).

Table 1: Considered features for each certificate type.

| | |
|----------------------|---|
| Birth Certificate | FIRSTNAME, LASTNAME, GENDER, BIRTHDATE, BIRTHPLACE, FATHERFIRSTNAME, FATHERLASTNAME, MOTHERFIRSTNAME, MOTHERLASTNAME |
| Death Certificate | FIRSTNAME, LASTNAME, GENDER, BIRTHDATE, BIRTHPLACE, DEATHDATE, DEATHPLACE, FATHERFIRSTNAME, FATHERLASTNAME, MOTHERFIRSTNAME, MOTHERLASTNAME RELATIVEFIRSTNAME, RELATIVELASTNAME, RELATIONTYPE |
| Marriage Certificate | GROOMFIRSTNAME, GROOMLASTNAME, GROOMAGE, BRIDEFIRSTNAME, BRIDELASTNAME, BRIDEAGE, GROOMFATHERFIRSTNAME, GROOMFATHERLASTNAME, GROOMMOTHERFIRSTNAME, GROOMMOTHERLASTNAME, BRIDEFATHERFIRSTNAME, BRIDEFATHERLASTNAME, BRIDEMOTHERFIRSTNAME, BRIDEMOTHERLASTNAME |

This database consists of 5,300,000 individual references extracted from 1,170,000 certificates (details are provided in Table 2). Considering the non-mentioned parents or relatives in some certificates, 500,000 references do not have any name (i.e., `first_name = null` and `last_name = null`). Therefore, we have 4,800,000 informative references. Among these references we have 170,000

distinct first names and 100,000 distinct last names. The dates mentioned in different certificates span a period of time between 1810 and 1920. The certificates are registered in 200 different municipalities.

Table 2: Statistical information of input data.

| | |
|--------------------------------|-----------|
| Number of Birth Certificate | 110,000 |
| Number of Marriage Certificate | 350,000 |
| Number of Death Certificate | 710,000 |
| Number of Extracted References | 5,270,000 |

3 Proposed Blocking Strategy

The most direct way of finding duplicates among the references is to apply pairwise comparison among the references and then consider the reference pairs with highest similarity values as a same entity. The computational order of this process is $O(n^2)$ which makes it infeasible in our project with roughly 5,000,000 references. In order to avoid having to compare all pairs of references, we propose a new blocking strategy to split all references into different blocking partitions. This process reduces the search space and diminishes the number of potential candidate pairs.

As a blocking strategy, the previous methods use the standard string encoding systems such as Soundex [7], metaphone [14] and double-metaphone [15]. Soundex, indexes the names based on their pronunciation in English. The main goal in this algorithm is that the letter with similar pronunciations be encoded with same characters so that spelling errors can be resolved. Metaphone is an extension of the Soundex code. Compared to Soundex, this code takes into account more information about variations and inconsistencies in English spelling and pronunciation. Afterwards, Double Metaphone was proposed which takes into account spelling peculiarities of a number of other languages. This indexing algorithm generates up to two codes for each word, that can improve some of the limitations of the original Metaphone for dealing with foreign languages.

In this section, we investigate the dataset of Dutch names [13] in order to first, evaluate the effectiveness of standard string encoding systems, second, extract the informative features for blocking strategy and third, propose a blocking strategy which considers both typing error and conventional name variations in Dutch names. The following subsections discuss in details the three mentioned steps.

3.1 Dutch Name Dataset

There are different writing variations for each (Dutch) name. For example, “Ghendrik”, “Haendrik”, “Handrikus”, “Hanri” and “Hedrik” are all referring to the same entity “Hendrik”. The reason behind this name variations could be of typing error or some historical/geographical issues. In this section, we use the dataset of Meertens Institute [13] to find the relationships among different

variations of Dutch names with their standard format. In total, the Meertens database contains 44,000 distinct first names (18,000 and 26,000 for male and females, respectively) and 120,000 distinct last names. The main attributes of the dataset are *name*, *standard name* and *popularity*. In this dataset, in average each standard first name has about 16 name alternatives while the standard last names have about 8 alternatives in average. However, there exists some standard names with very high number of alternatives.

3.2 Extracting Informative Features

So far, we have analyzed the following features from the the Meertens dataset.

- F1:** [Boolean feature] If first 2 letters of name and standard name are equal.
- F2:** [Boolean feature] If first 3 letters of name and standard name are equal.
- F3:** [Boolean feature] If last 2 letters of name and standard name are equal.
- F4:** [Boolean feature] If last 3 letters of name and standard name are equal.
- F5:** [Boolean feature] If size of name and standard name are equal.
- F6:** [Integer feature] Absolute difference of name length and standard length.
- F7:** [Integer feature] Number of longest first equal chars.
- F8:** [Integer feature] Number of longest last equal chars.
- F9:** [Boolean feature] If *soundex* code of name and standard name is equal.
- F10:** [Boolean feature] If *metaphone* code of name and standard name is equal.
- F11:** [Boolean feature] If *double-metaphone* code of name and standard name is equal.
- F12:** [Integer feature] Longest common chars between name and its standard name.

Table 3 calculates the *min*, *mean*, *s.t.d.* and *max* for each feature for the *male first name*, *female first name* and *last name* datasets.

Table 3 provides detailed information about the 12 very basic and important features of Dutch names. Among all the features, **F1** is a very discriminative feature as it is true in more than 70% of the cases (i.e., the first two letters of a name and its standard name are equal in more than 70% of the cases). Among the phonetic-based string similarity measures (**F9**, **F10** and **F11**) Soundex code has the highest score of being identical between name and its standard form in about 50% of the cases. However, the absolute difference of name length and its standard form length **F6** has a maximum of 15, which means some name lengths can deviate very much from length of its standard form.

In next subsection, we use the most discriminative features discussed in this section to build a blocking key for partitioning the references based on their similarities.

3.3 Blocking Key Generation

Following the conclusions discussed in Section 3.2, we propose a new blocking key strategy which considers both name variations and spelling error.

Table 3: Feature analysis of Dutch names. Features **F6**, **F7**, **F8** and **F12** are continuous features and the rest of features are all binary.

| Feature | first name (male) | | | | first name (female) | | | | last name | | | |
|---------|-------------------|------|--------|-----|---------------------|------|--------|-----|-----------|------|--------|-----|
| | min | mean | s.t.d. | max | min | mean | s.t.d. | max | min | mean | s.t.d. | max |
| F1 | 0 | 0.71 | 0.46 | 1 | 0 | 0.70 | 0.46 | 1 | 0 | 0.79 | 0.40 | 1 |
| F2 | 0 | 0.52 | 0.49 | 1 | 0 | 0.50 | 0.49 | 1 | 0 | 0.60 | 0.48 | 1 |
| F3 | 0 | 0.36 | 0.49 | 1 | 0 | 0.42 | 0.49 | 1 | 0 | 0.54 | 0.50 | 1 |
| F4 | 0 | 0.27 | 0.44 | 1 | 0 | 0.30 | 0.45 | 1 | 0 | 0.45 | 0.49 | 1 |
| F5 | 0 | 0.35 | 0.48 | 1 | 0 | 0.34 | 0.48 | 1 | 0 | 0.43 | 0.5 | 1 |
| F6 | 0 | 1.15 | 1.31 | 15 | 0 | 1.10 | 1.31 | 13 | 0 | 0.77 | 0.88 | 10 |
| F7 | 0 | 2.90 | 2.07 | 13 | 0 | 2.90 | 2.06 | 11 | 0 | 3.57 | 2.41 | 16 |
| F8 | 0 | 1.57 | 2.07 | 13 | 0 | 1.77 | 2.06 | 11 | 0 | 2.59 | 2.65 | 16 |
| F9 | 0 | 0.50 | 0.5 | 1 | 0 | 0.47 | 0.5 | 1 | 0 | 0.58 | 0.49 | 1 |
| F10 | 0 | 0.31 | 0.47 | 1 | 0 | 0.29 | 0.46 | 1 | 0 | 0.42 | 0.49 | 1 |
| F11 | 0 | 0.39 | 0.49 | 1 | 0 | 0.37 | 0.49 | 1 | 0 | 0.49 | 0.49 | 1 |
| F12 | 0 | 3.90 | 1.85 | 14 | 0 | 3.98 | 1.85 | 12 | 0 | 4.82 | 2.1 | 17 |

$$\begin{aligned}
\text{BLOCKING_KEY}(r_i) = & \text{GENDER}(r_i) \\
& + \text{FIRSTNAME}(r_i)[: 3] + \text{FIRSTNAME}(r_i)[- 2 :] \\
& + \text{LASTNAME}(r_i)[: 3] + \text{LASTNAME}(r_i)[- 2 :] \\
& + \text{soundex}(\text{FIRSTNAME}(r_i)) + \text{soundex}(\text{LASTNAME}(r_i)) \quad (1)
\end{aligned}$$

where in Formula 1, `STRING[:i]` and `STRING[-i:]` refers to the first i and last i characters of the `STRING`, respectively.

For each reference in our dataset, we build its blocking key and then we assume all the references with similar blocking key in the same block. This process builds blocks with different sizes (=member count). As the size of one block increases our confidence for that block decreases. Formula 2 calculates the Confidence value for block b_i .

$$\text{Conf}(b_i) = \frac{N}{\text{size}(b_i)} - 1 \quad (2)$$

In this formula, N is the number of all references and `size(b_i)` returns the number of references belong to block b_i . In the most extreme case, all references belong to one block and the confidence of that block becomes 0 (i.e., $\text{Conf}(b_0) = N/N - 1 = 0$).

In the next section, we propose a contextual similarity measure to compare all reference pairs belonging to similar blocks.

4 Contextual Similarity Measure

After partitioning all the references into different blocks, now we could compare all the reference pairs belonging to the same block. The existing similarity

measures such as Levenstinen, Jaro-Winkler, etc. [17, 11, 2] simply calculate the string similarity between reference’s First and Last Names neglecting the contextual information available for each reference.

As discussed in Section 2, the references that are used in this paper are extracted from three different type of certificates: “Birth”, “Death” and “Marriage”. Each of these certificate types contains the information of a group of Subfigs. 2(a) and 2(b) show that which should be considered when we compare two references. For example, imagine two arbitrary references r_i and r_j where both belong to the same block b_k (i.e., $r_i \in b_k$ and $r_j \in b_k$). If their partners both belong to another block b_L (i.e., $\text{partner}(r_i) \in b_L$ and $\text{partner}(r_j) \in b_L$) then the probability that r_i and r_j referring to same entity should be increased, comparing to the case that their partners belong to different blocks. To consider the contextual information hidden in each certificate, we use Formula 3 to extract the Block Context of each certificate c_i .

$$BC(c_i) = \bigcup_{r_j \in c_i, r_j \in b_k} b_k \quad (3)$$

$BC(c_i)$ simply includes the block ids of all reference members of certificate c_i . We propose a following similarity measure to consider both String similarity among references in addition to their certificate contextual information.

$$\text{Similarity}(r_i, r_j) = \text{Sim}_{NC}(r_i, r_j) + \text{Sim}_{BC}(r_i, r_j) \quad (4)$$

In Formula 4, $\text{Sim}_{NC}(r_i, r_j)$ and $\text{Sim}_{BC}(r_i, r_j)$ calculates the “No Context” and “Blocking Context” similarity values between two references r_i and r_j , respectively. In, Formula 5, we use Jaro-Winkler algorithm [17] to calculate the string similarity between FirstName and LastName of two references r_i and r_j .

$$\begin{aligned} \text{Sim}_{NC}(r_i, r_j) = & \frac{1}{2} \left[\text{JaroWinkler}(\text{FIRSTNAME}(r_i), \text{FIRSTNAME}(r_j)) \right] \\ & + \frac{1}{2} \left[\text{JaroWinkler}(\text{LASTNAME}(r_i), \text{LASTNAME}(r_j)) \right] \quad (5) \end{aligned}$$

If two references r_i and r_j belong to two certificates c_i and c_j respectively, then we use Formula 6 to calculate the contextual-based similarity between them.

$$\text{Sim}_{BC}(r_i, r_j) = \frac{\sum_{b_k \in \{BC(c_i) \cap BC(c_j)\}} \text{Conf}(b_k)}{\sum_{b_k \in \{BC(c_i) \cup BC(c_j)\}} \text{Conf}(b_k)} \quad (6)$$

Formula 6 checks if the other references in the same certificates belonging to the similar blocks or not.

5 Empirical Studies

In this section, first, our proposed blocking strategy is applied on the genealogical dataset introduced in Section 2. Then, the contextual similarity measure

proposed in Section 4 is used to extract the links between certificates. Finally, by means of examples and manual evaluation, the final results are evaluated by domain experts.

5.1 Results of Proposed Blocking Technique

We applied our proposed blocking strategy to the BHIC dataset, which contains about 5,000,000 references. As a result, 690,000 blocks are constructed with different sizes ranging from size 1 to 3,845, where each of the blocks of size one contains just 1 reference and the block with largest size contains 3,845 references; the block key of this largest block is “female.Mar_Jan_ia_en.M600_J525” which turns out to be the most pattern among Dutch references reported between 1890 and 1920. The average block size is 7 and the standard deviation is 29. This shows that not many blocks of very large size exist. Fig. 1 shows the block size distribution by focusing on the blocks with size 2 to 50.

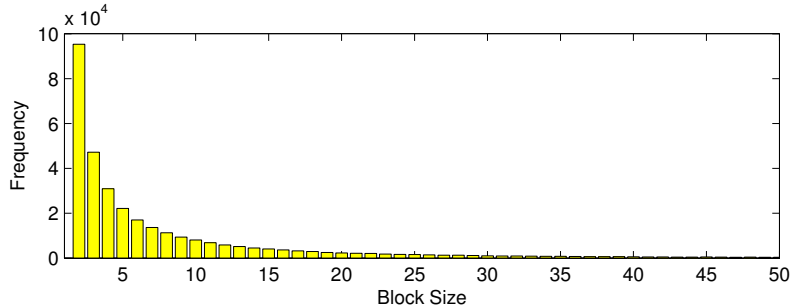


Fig. 1: Distribution of block sizes. The average block size is 7 and the standard deviation is 29. Blocks with size 1 are excluded from the figure as they contain just 1 reference and are not informative in matching.

Given that originally about 25×10^{12} pairwise comparisons (i.e., $5,000,000 \times 5,000,000$) were required for accomplishing the task of traditional entity resolution, based on the partitions introduced by the proposed blocking strategy, the average search space is now reduced by 3.5×10^{-5} (i.e., $\frac{690,000 \times 7^2}{25 \times 10^{12}}$).

5.2 Result of Contextual Similarity Measures

In this subsection, we report the results of applying the proposed contextual similarity measure on matching candidates from all blocks b_i of size less than or equal to 100 (i.e., $size(b_i) \leq 100$). This generates in total 40,489,999 matching pairs with similarity scores less than 2.0, where 14% of matching candidates (i.e., 5,703,687 pairs of references) have a score larger than 1.2. The score distribution of these matching candidates are shown in Fig. 2.

Subfigs. 2(a) and 2(b) show that many of the matching candidates have a matching score equal to 2.0. We consider these matches as perfect matches (identical certificates) which can refer to either record duplicates or correct matches where the same group of references are mentioned in both certificates.

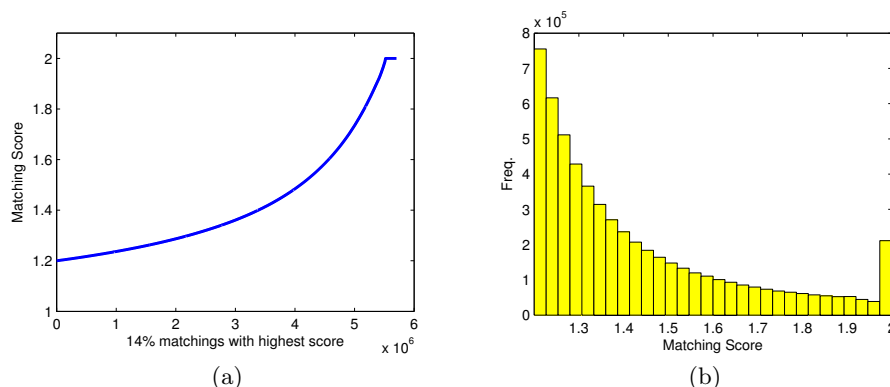


Fig. 2: The contextual similarity measure of 14% matching candidates with highest score (a) the matching candidates are sorted based on the matching score. 185,000 of the candidates have a perfect match with exact matching score 2.0. (b) distribution of the matching candidates with a score higher than 1.2.

Table 4 provides a categorization of all the certificate-pairs that contain the 185,000 perfect matches with score = 2.0 (i.e., about 90,000 certificate pairs). Table 4 shows that more than 50,000 matches refer to connections between death certificates. By further exploration of the results we realized that about 28% of these matches refer to record duplicates³. Besides 78% of the matches refer to the certificates with an average of 2.1 year difference in issue time. One major reason for these matches can be early death of child in families of 19th century which results in using the same name for next child which might end with death of next child as well.

By exploring the 28,000 matches between death and birth certificates, again 25% of these matches can be because of the early death of the birth as both certificates are issued in the same place in the same year. However, 75% of the matches refer to the matches of birth and death of an entity with an average age of 28.4 years. The reason for having such a low average age is that in such death certificates, no relative name is mentioned for the deceased person (i.e., most probably the deceased has been single), which is the case for young references. For details about other matching types refer to Table 4.

5.3 Result of Contextual Matching

In this subsection, we discuss the results of the proposed contextual matching technique by presenting some examples of the revealed matches between different references, and also the results of a manual check of over 300 instances of data will be provided.

³ record duplicates in our genealogical dataset refer to the cases that an event is issued by two authorities or due to data storage inconsistencies the record is stored more than one time with minor differences in location name, archive index, etc.

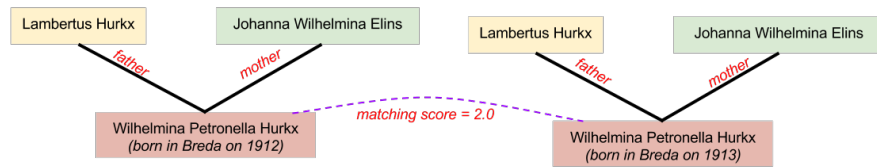
Table 4: Categorisation of matches with score 2.0 where the number of certificates seen for each type of matching and the average difference in date of certificate issue is provided.

| Matching Type | Avg. Date Diff. | Freq. |
|---|-----------------|--------|
| Death Certificate (<i>to</i>) Death Certificate <ul style="list-style-type: none"> • 28% due to record duplicate • 72% due to early child death and using the name for next child, or other reasons. | 2.1 years | 51,293 |
| Death Certificate (<i>to</i>) Birth Certificate <ul style="list-style-type: none"> • 25% due the early child birth • 75% due to matching between birth and death of an entity, or others | 9.8 years | 28,401 |
| Birth Certificate (<i>to</i>) Birth Certificate <ul style="list-style-type: none"> • 100% due to early child death and using the name for next child, or others | 3.5 years | 8,679 |
| Marriage Certificate (<i>to</i>) Marriage Certificate <ul style="list-style-type: none"> • 100% due the record duplicate | 0 years | 1642 |
| Marriage Certificate (<i>to</i>) Death Certificate <ul style="list-style-type: none"> • 100% due to matching between a death and marriage certificate of an entity when the parents of deceased partner are not mentioned in the marriage certificate | 26 years | 99 |

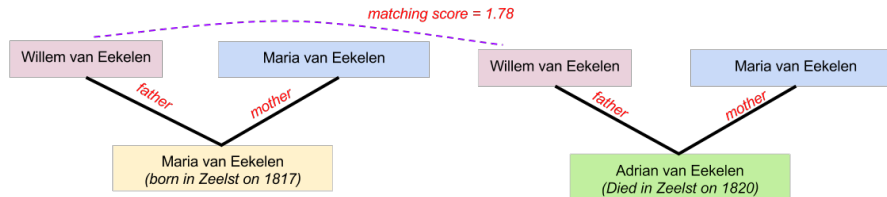
By a careful study on discovered matchings with highest score, different patterns can be introduced, where for each pattern specific certificate roles are matched with each other. For instance, one common pattern can be a match between two born references due to similarities between their names and their parents' names. Subfig. 3(a) illustrates an example where the match between two born references is considered as a perfect match with score = 2.0. In this example, both certificates are issued in the same place with one year difference in time of issue. Therefore, this case might refer to an early death of a born child, where her name is used for the the next born child in the following year.

Another common pattern in discovered matches is the high contextual similarity between parents of a birth or death certificate and parents of another birth or death certificate. Subfig. 3(b) shows an example of this pattern where a father in a birth certificate is matched with father in a death certificate which is issued three years later in the same place. The matching is not perfect (score = 1.78) as the children have different first names (this can refer to two siblings).

In order to evaluate the quality of the revealed matches between references, we chose 324 matches randomly (from matches with score higher than 1.3) and for each match a domain expert, familiar with genealogical data, used the provided evidence (names of references, family relations, place and date of issue, blocking key and blocks confidence) to evaluate the match by choosing either a



(a) Matching between two birth certificates (score = 2.0)



(b) Matching between a birth certificate and a death certificate (score = 1.78)

Fig.3: Examples of matched pairs with different scores (a) A perfect match between two birth certificates, which are issued in the same place with one year difference. This can be due to early death of the first child, and using her name for next born child. (b) Father in a birth certificate is matched to the father in death certificate of another child 3 years later. (As can be seen the born child in left certificate has an identical name with mother)

True Positive or a *False Positive* category⁴. This evaluation approach is similar to the approach described in [3]. Fig. 4 depicts the distribution of true positive and false positive matches for manual evaluation. In this figure, no evidence of false positive matches with a score higher than 1.7 can be seen while for lower scores many false positive matches are seen.

6 Conclusions and Future Work

The reliability of any data analysis method strongly depends on the quality of the input data. Considering the Genealogical data, with huge amount of inaccurate information and different types of ambiguities, applying Entity Resolution techniques for cleaning and integrating the references extracted from different historical certificates should be taken into account as the first step toward any data analysis approach. Traditional methods, use a standard string similarity measure to calculate the similarity among references, neglecting the contextual

⁴ Please note that due to missing data, typing errors, redundancies and lack of extra evidence confirming that two references point the same real entity is impossible in many of the cases. Therefore, in evaluations of this paper, we stick to the evidence at hand and assume that a reasonable similarity between two references, similar family members, and feasible date and similar places can suggest a true positive match, otherwise it will be considered as a false positive match.

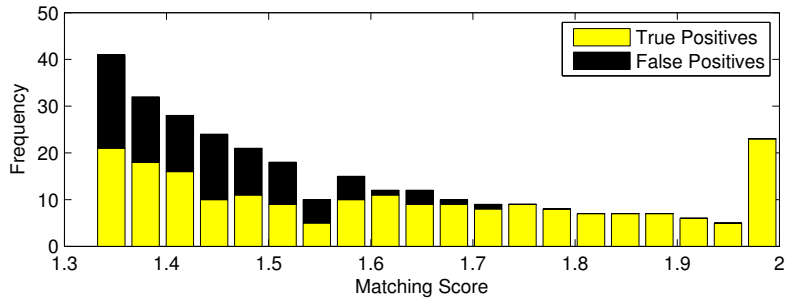


Fig. 4: Distribution of True/False positive matches using a manual evaluation over 300 matching candidates. The results show 70% True positives and 30% False positives. The false positives are detected for scores lower than or equal to 1.7.

information available for each reference, and then introduce the most similar pairs as matches. In order to avoid having to compare all pairs of references, we investigated the dataset of dutch name [13], we selected the most discriminative features and finally, we proposed a blocking strategy to split all references into different blocking partitions. As a result of applying our proposed blocking strategy, the search space is reduced by 3.5×10^{-5} . To compare all the references belong to the similar blocks, we proposed a contextual similarity measure which not only considers the string similarity among references but also contextual information available for them. According to considered genealogical certificates, we defined context of each reference as its first level family relationships (i.e., partner, father, mother etc) and accordingly, we increased the probability of reference matches if they share a common context. We evaluated our proposed contextual similarity measure from different perspectives and among many discussed patterns, the “early child death” pattern discovered to be prominent. In this pattern, child dies in early years and the family uses the same name for the next born baby.

Regarding future research induced by our work, we see three particularly important directions for refinement and extension of our approach. First, further exploration of possibilities for extensive validation of the achieved results. This is challenging because we typically do not have grounded truth against which the results can be directly compared. We have already discussed and validated our results in Sections 5.2 and 5.3 by human domain experts; however, it would be very useful and considerably more efficient to have a way of (at least partially) evaluating the results automatically or at least semi-automatically by simulating domain-expert behavior. Second, investigation of Random Walk to take into account a wider range of contextual information such as second-level family members. And third, when the graph of entities is built, the study of common characteristics of specific groups of entities in order to unravel previously unknown information and connections within the groups [8].

Acknowledgments. This research has been supported under the NWO CATCH program in the MISS project (project no. 640.005.003). The authors are grateful to the BHIC center for the support in data gathering and direction.

References

1. William W. Cohen, Henry A. Kautz, and David A. McAllester. Hardening soft information sources. In Raghu Ramakrishnan, Salvatore J. Stolfo, Roberto J. Bayardo, and Ismail Parsa, editors, *KDD*, pages 255–259. ACM, 2000.
2. Julia Efremova, Bijan Ranjbar-Sahraei, and Toon Calders. A hybrid disambiguation measure for inaccurate cultural heritage data. In *the 8th Workshop on LaT-eCH*, pages 47–55, 2014.
3. Julia Efremova, Bijan Ranjbar-Sahraei, Frans A. Oliehoek, Toon Calders, and Karl Tuyls. An interactive, web-based tool for genealogical entity resolution. In *25th Benelux Conference on Artificial Intelligence*, pages 376–377, 2013.
4. Julia Efremova, Bijan Ranjbar-Sahraei, Frans A. Oliehoek, Toon Calders, and Karl Tuyls. A baseline method for genealogical entity resolution. In *Workshop on Population Reconstruction*, 2014.
5. Lise Getoor and Ashwin Machanavajjhala. Entity resolution for big data. In *Proceedings of the 19th ACM SIGKDD*, pages 1527–1527. ACM, 2013.
6. Mauricio A. Hernández and Salvatore J. Stolfo. The merge/purge problem for large databases. *SIGMOD Rec.*, 24(2):127–138, May 1995.
7. Donald E. Knuth. *The art of computer programming 1: Fundamental algorithms 2: Seminumerical algorithms 3: Sorting and searching*, 1968.
8. Verboven Koenraad, Carlier Myriam, and Dumolyn Jan. A short manual to the art of prosopography. In Keats-Rohan K.S.B., editor, *Prosopography Approaches and Applications. A Handbook*, pages 35–69. Unit for Prosopographical Research (Linacre College), Oxford, 2007.
9. Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD*, pages 169–178. ACM, 2000.
10. Alvaro E. Monge and Charles Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *DMKD*, pages 0–, 1997.
11. Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.
12. Howard B. Newcombe, James M. Kennedy, S.J. Axford, and A.P. James. Automatic Linkage of Vital Records. *Science*, 130(3381):954–959, October 1959.
13. Meertens Institute Databases of Names. <http://www.meertens.knaw.nl/cms/en/collections/databases>. Accessed 2014-06-27.
14. Lawrence Philips. Hanging on the metaphone. *Computer Language*, 7(12), 1990.
15. Lawrence Philips. The double metaphone search algorithm. *C/C++ users journal*, 18(6):38–43, 2000.
16. Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the Eighth ACM SIGKDD, KDD '02*, pages 269–278, New York, NY, USA, 2002. ACM.
17. William E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. 1990.
18. William E. Winkler. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*, 1999.

ROCsearch — An ROC-guided Search Strategy for Subgroup Discovery

Marvin Meeng¹, Wouter Duivesteijn², and Arno Knobbe¹

¹ LIACS, Leiden University, {m.meeng,a.j.knobbe}@liacs.leidenuniv.nl

² Fakultät für Informatik, LS VIII, Technische Universität Dortmund,
wouter.duivesteijn@tu-dortmund.de

Subgroup Discovery (SD) aims to find coherent, easy-to-interpret subsets of the dataset at hand, where something exceptional is going on. Since the resulting subgroups are defined in terms of conditions on attributes of the dataset, this data mining task is ideally suited to be used by non-expert analysts. The typical SD approach uses a heuristic beam search, involving parameters that strongly influence the outcome. Unfortunately, these parameters are often hard to set properly for someone who is not a data mining expert; correct settings depend on properties of the dataset, and on the resulting search landscape. To remove this potential obstacle for casual SD users, we introduce ROCSEARCH [1], a new ROC-based beam search variant for Subgroup Discovery.

On each search level of the beam search, ROCSEARCH analyzes the intermediate results in ROC space to automatically determine a sensible search width for the next search level. Thus, beam search parameter setting is taken out of the domain expert’s hands, lowering the threshold for using Subgroup Discovery. Also, ROCSEARCH automatically adapts its search behavior to the properties and resulting search landscape of the dataset at hand. Aside from these advantages, we also show that ROCSEARCH is an order of magnitude more efficient than traditional beam search, while its results are equivalent and on large datasets even better than traditional beam search results.

Acknowledgment This research is supported in part by the Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 “Providing Information by Resource-Constrained Analysis”, project C1.

References

1. M. Meeng, W. Duivesteijn, A. Knobbe, ROCsearch – An ROC-guided Search Strategy for Subgroup Discovery, Proc. SDM, pp. 704–712, 2014.

Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

SHrimp: Descriptive Patterns in a Tree

Sibylle Hess, Nico Piatkowski, and Katharina Morik

TU Dortmund University, 44227 Dortmund, Germany
sibylle.hess@tu-dortmund.de
<http://www-ai.cs.tu-dortmund.de>

Abstract. The appliance of the minimum description length (MDL) principle to the field of theory mining enables a precise description of main characteristics of a dataset in comparison to the numerous and hardly understandable output of the popular frequent pattern mining algorithms. The loss function that determines the quality of a pattern selection with respect to the MDL principle is however difficult to analyze and the selection is computed heuristically for all known algorithms. With SHRIMP, the attempt to create a data structure that reflects the influences of the pattern selection to the database and that enables a faster computation of the quality of the selection is initiated.

Keywords: Pattern Mining, MDL, Pattern Selection, Itemsets.

1 Introduction

The identification of interesting subsets and pattern mining in general is a fundamental concept when it comes to compute characteristics of large databases. Patterns shall reflect the inherent structure of the dataset, particular interesting or at least reoccurring parts of it. As described by Mannila and Toivonen [7] the theory of the dataset, represented by the subsets that satisfy a given predicate of interest, is required. The most common practice is the frequent pattern mining [1], where the relevance of a pattern is identified with its frequency. The monotonic property of frequent sets, that all subsets of a frequent pattern are also frequent, results however in an output with highly redundant patterns. In addition, the threshold that denotes the minimal frequency to be fulfilled, is hard to set. A high threshold often results in a small set of patterns that reveals nothing but common knowledge and a lower one lets the number of issued patterns literally explode. These enormous amounts of patterns are hard to understand and the connections or correlations between them are not given explicitly.

Another approach has been introduced by Siebes et al. with KRIMP [12] where the most interesting patterns are selected according to the *minimum description length* (MDL) principle [3]. The best set of patterns is identified as the one that

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

compresses the database best. This strategy reduces the number of returned patterns drastically, e.g. from billions to less than 700 itemsets for the mushroom database, and has a variety of applications [10,11,5]. KRIMP has as input a preferable large set of frequent patterns, the code word candidates, a set that is likely to be very much larger than the input dataset. KRIMP tries to find the best selection of patterns from this candidate set with respect to the encoding of the database in a greedy procedure. Every candidate pattern is regarded once and the compression size with the considered pattern is computed. Since the consequences for the encoding of the database when a single candidate is added to the set of code words is difficult to derive, the whole database has to be traversed for each of the candidates to identify affected parts of the database such that their encoding can be recomputed.

The algorithm SLIM [9] encounters this problem by a candidate pattern generation that follows directly from the current selection of code words. In every iteration, candidates are generated and sorted by their estimated compression gain and the first pattern that enhances the compression size indeed is accepted. The estimation of the compression size requires however an identification of the affected parts of the database and for some candidates the actual compression size has to be computed as well. These operations are performed for most of the time and the combinatorial possibilities for the candidate generation are numerous, thus, an indexing structure that supports these operations and that gives insight into the consequences of integrating a pattern into the encoding, is desirable.

With SHRIMP we introduce a tree structure that facilitates a fast identification of the interesting parts of a dataset and enables a direct determination of the consequences that result from a change of the current pattern selection. In addition, the tree structure reflects the dependencies of mined patterns completely and can be used to get fundamental as well as more profound views of the characteristics of the given dataset.

We proceed as follows: In section 2 we give a theoretical introduction to the principles of KRIMP and the algorithmic procedure. We proceed with an explanation of the tree structure of SHRIMP and its application concerning the candidate selection in section 3. Runtime comparisons for some well studied datasets are shown and discussed in section 4 and we conclude in section 5.

2 Preliminaries

Let $\mathcal{P}(X)$ denote the power set of X . Given a set of items \mathcal{I} we define a database $\mathcal{D} \subseteq \mathcal{P}(\mathcal{I})$ as a set of transactions $t \subseteq \mathcal{I}$. For a given minimum support $minsup \in [0, 1]$ a set $X \in \mathcal{I}$ is called frequent if

$$sup(X) = \frac{|\{t \in \mathcal{D} | X \subseteq t\}|}{|\mathcal{D}|} \geq minsup.$$

The value $sup(X)$ is called the support of an itemset X . We define \mathcal{F} as the set of all frequent patterns, the regarded candidates.

2.1 The MDL principle

MDL has been introduced by Rissanen et al. [8] as an applicable version of the Kolmogorov complexity [6]. Given a set of models \mathcal{M} , the best model is identified as the one that minimizes the compression size

$$L(\mathcal{D}, M) = L(\mathcal{D}|M) + L(M),$$

whereby $L(\mathcal{D}|M)$ denotes the compression size of the database in bits, assuming that model M is used for the encoding and $L(M)$ is the description size in bits of the model M itself.

2.2 Encoding the Database

We define a coding set $CS \subseteq \mathcal{P}(\mathcal{I})$ as a set of patterns that contains at least all singleton itemsets $\{\{x\}|x \in \mathcal{I}\}$. Let $code : CS \rightarrow \{0, 1\}^*$ be a mapping from patterns in the coding set to a finite, unique and prefix-free code. A *code table* is denoted by a set of pairs

$$CT \subseteq \{(X, code(X))|X \in CS\}.$$

Code tables represent the compressing models in KRIMP and can be interpreted as dictionaries for code words. Once the *code* function is determined, the coding set induces a code table. The problem can thus be formalized as the task of finding the best compressing coding set of a database. The explicit *code* function is introduced in section 2.3.

The function $cover : \mathcal{P}(\mathcal{I}) \rightarrow \mathcal{P}(CS)$ selects the patterns of a coding set, and with that the code words, that encode a specified transaction or itemset in general. With respect to a given transaction t , the set $cover(t)$ is called *cover set* and the items $x \in X \in cover(t)$ are called *covered by X*.

2.3 Computing the Compression Size

The codes are created such that the more frequently used patterns get the shorter codes. The existence of such a code is guaranteed by Theorem 5.4.1 in Cover & Thomas [2], that states for a given distribution P over a finite set \mathcal{X} , that there exists an optimal prefix-free code such that the length of the code $L(code(X))$ for $X \in \mathcal{X}$ is given by

$$L(code(X)) = -\log(P(X)).$$

Codes that satisfy this property are e.g the Shannon-Fano or Huffman code. The probability distribution over all itemsets in the code table is defined as follows. Let CT be a code table and $X \in CS$ an itemset of the respective coding set. The probability of X to be used for the encoding in a given database is defined as

$$P(X) = \frac{usage_{CT}(X)}{\sum_{Y \in CT} usage_{CT}(Y)},$$

where $usage_{CT}(X)$ denotes the number of transactions that use X for their encoding. So, the usage of an itemset $X \in CT$ is equal to the frequency of the code $code(X)$ in the encoded database. Now, the size of a database \mathcal{D} compressed by a code table CT can be computed as

$$\begin{aligned} L(\mathcal{D}|CT) &= \sum_{t \in \mathcal{D}} \sum_{X \in cover(t)} L(code_{CT}(X)) \\ &= - \sum_{X \in CS} usage_{CT}(X) \cdot \log(P(X)). \end{aligned}$$

The latter formulation as a summation over the coding set allows a faster and more incremental way of computation. Since the code table contains assumably much less sets than transactions in the database exist, this sum is computed with low expenses if the usage function is computed capably.

A code table CT is represented by means of the *standard code table* that provides codes for singleton itemsets. The items in the coding set of CT are represented by their codes from the standard code table ST , such that the description length of a code table is calculated as

$$\begin{aligned} L(CT) &= \sum_{X \in CS} \left(L(code_{CT}(X)) + \sum_{x \in X} L(code_{ST}(x)) \right) \\ &= - \sum_{X \in CS} \left(\log(P(X)) + \sum_{x \in X} \log(sup(\{x\})) \right). \end{aligned}$$

Applying the MDL principle, the total compression size is calculated as the sum of the description sizes $L(\mathcal{D}|CT) + L(CT)$. We observe, that the compression size depends mainly on the usage function. Thus, an understanding of usage dependencies is likely to be a crucial point for all KRIMP-related algorithms.

2.4 The Algorithm Krimp

KRIMP (Alg. 1) has as input a database \mathcal{D} and the frequent patterns of a preferably low minimum support \mathcal{F} . The frequent patterns are sorted in *standard candidate order* that is first decreasing on support, second decreasing on cardinality and at last lexicographically (line 2). The code table is initialized to the standard code table (line 3) and each candidate pattern is regarded in the specified order. A candidate is added to the code table if this reduces the compression size (lines 4-9). Since the compression size decreases monotonically in the number of regarded candidates, the best compression in this procedure is achieved if the minimum support is set to $\frac{1}{|\mathcal{D}|}$. This results however in an extremely large candidate set due to the pattern explosion. A speed-up of the usage calculation that is carried out for each of the frequent item sets would thus accelerate the whole process significantly.

Algorithm 1 Krimp [12].

```
1: procedure KRIMP( $\mathcal{D}, \mathcal{F}$ )
2:    $\mathcal{F} \leftarrow \text{sort}(\mathcal{F})$  ▷ in standard candidate order
3:    $CT \leftarrow \text{STANDARDCODETABLE}(\mathcal{D})$ 
4:   for  $fp \in \mathcal{F} \setminus \mathcal{I}$  do
5:      $CT_c \leftarrow CT \cup fp$ 
6:     if  $L(\mathcal{D}, CT_c) < L(\mathcal{D}, CT)$  then
7:        $CT \leftarrow CT_c$ 
8:     end if
9:   end for
10: end procedure
```

Computing the Usage. The usage calculation relies on the method STANDARDCOVER (Alg. 2). It is invoked for every transaction that contains the currently regarded candidate pattern. Assuming that the code table is sorted by a total order, the algorithm traverses the code table and selects the first pattern that is contained in the specified transaction (line 2). The used order for this procedure is called *standard cover order* that is first decreasing on cardinality, second decreasing on support and at last lexicographically. If the transaction is covered completely by elements of the code table, the algorithm stops (line 3), otherwise the procedure is called recursively for the uncovered part of the transaction (line 6). The identification of those transactions that contain a specified

Algorithm 2 Standard Cover [12].

```
1: procedure STANDARDCOVER( $t, CT$ )
2:    $X^* \leftarrow \min\{X \mid X \subseteq t \wedge X \in CT\}$ 
3:   if  $t \setminus X^* \leftarrow \emptyset$  then
4:     return  $\{X^*\}$ 
5:   else
6:     return  $\{X^*\} \cup \text{STANDARDCOVER}(t \setminus X^*)$ 
7:   end if
8: end procedure
```

candidate pattern is not trivial, the implementation of transactions as bit vectors improves the process significantly, but the question arises if there exists some representation of the database that enables an identification of affected parts without a scan of the whole database.

3 SHrimp

With SHRIMP we present a tree structure, called *SH-tree*, that reflects the encoded database and enables a faster computation of the usage function. The tree is similar to to the *FP-tree* introduced by Han et al. [4], except that nodes

contain sets of items and not only single items. Each branch from the root to a leaf represents a transaction and provides the information about all possible and the current encoding for a given coding set. More precisely, the nodes fulfill the following properties.

Definition 1 (SH-tree). *Given a total order on itemsets \preceq , a SH-tree is a tree structure with the following properties*

1. *The root of the tree is labeled as null.*
2. *Each node $n \neq \text{null}$ is described by a pattern ($n.\text{pattern}$), a set of inactive items ($n.\text{inact}$), a counter ($n.\text{freq}$) and the pointers to its children ($n.\text{children}$) and the parent node ($n.\text{parent}$). $n.\text{freq}$ denotes the number of transactions represented by the branch from the root to that node.*
3. *For a node n and the parent node $n_p = n.\text{parent} \neq \text{null}$ it holds that*

$$n_p.\text{pattern} \preceq n.\text{pattern}$$

The meaning of the field *inact* requires a more detailed explanation.

Definition 2. *Let n be a node with $|n.\text{pattern}| \geq 2$ and let $\text{anc}(n)$ denote the ancestor nodes of n . For an item $x \in n.\text{pattern}$ it holds that*

$$x \in n.\text{inact} \Leftrightarrow \exists n_a \in \text{anc}(n) : n_a.\text{inact} = \emptyset \wedge x \in n_a.\text{pattern}.$$

Items of a node n occurring in the set $n.\text{inact}$ are called inactive, otherwise active. Accordingly, nodes that have inactive items are called inactive, otherwise active.

Inactive nodes denote patterns that would be used for a transaction if some of their items were not already encoded by another pattern. The application of these nodes may reduce the complexity of the tree, because singletons that occur in inactive nodes must not be displayed explicitly, but it may also enlarge the complexity due to the reflection of redundant information. Inactive nodes are however integrated into the tree because they define the consequences of a pattern selection change and enable thereby a fast computation of usage impacts if a certain pattern is removed or added. To get now a fundamental understanding of the algorithm, we examine an example of a tree first.

Example 1 (SH-tree). Let \mathcal{D} be the sample database displayed in Table 1, \preceq the standard cover order and the coding set be given by the patterns $\{b, d, e\} \preceq \{a, c\} \preceq \{a, g\}$ besides of the singleton itemsets. The resulting cover sets of \mathcal{D} computed by the standard cover function are displayed in the right column of Table 1. The corresponding tree representation of the database is shown in Fig. 1. We can see that each transaction is represented by a branch, every leaf is marked by the number of the equivalent transaction. Inactive nodes and items are greyed out.

| TID | transaction | cover set |
|-----|--------------------|----------------------------|
| 1 | a, b, d, e, f, g | $\{b, d, e\}\{a, g\}\{f\}$ |
| 2 | a, b, c, d, e, g | $\{b, d, e\}\{a, c\}\{g\}$ |
| 3 | a, c, e, f | $\{a, c\}\{e\}\{f\}$ |
| 4 | a, b, c, d, e, f | $\{b, d, e\}\{a, c\}\{f\}$ |
| 5 | a, c, e, g, f | $\{a, c\}\{e\}\{g\}\{f\}$ |
| 6 | a, b, d, e, g | $\{b, d, e\}\{a, g\}$ |

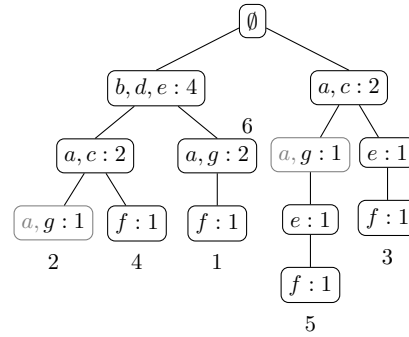


Table 1: A sample transactional database and the resp. cover sets. **Fig. 1:** The tree representation of the covered database.

The question arises how this structure can be utilized to compute the usage function after a pattern is integrated. For this occasion we further examine our running example.

Example 2 (Usage Computation). We imagine that the regarded candidate is the pattern $\{c, e, f\}$, and $\{b, d, e\} \preceq \{c, e, f\} \preceq \{a, c\} \preceq \{a, g\}$. The computation of emerging usage differences starts with an identification of branches that use the designated pattern for their encoding. The algorithm examines the smallest child of the root node, i.e. the node with the pattern $\{b, d, e\}$ first. Since b and e would be encoded by this child furthermore, the candidate pattern is not used in this sub tree. The algorithm proceeds with the right sub tree, finds that transactions 5 and 3 would use the candidate and stores the corresponding leaves with the singleton $\{e\}$. In these branches, the effects of the encoding by the candidate are calculated, i.e. $\{a, c\}$ is not used anymore, but the node with $\{a, g\}$ becomes active again. If the resulting usage function gives a better compression size, the pattern is integrated into the tree. The consequent tree is displayed in Fig. 2.

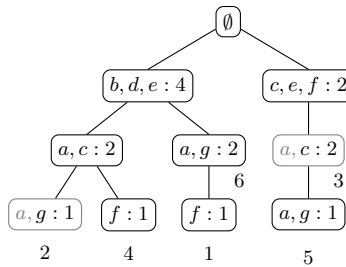


Fig. 2: The SH-tree of Example 1 when the pattern $\{c, e, f\}$ is inserted.

3.1 The Algorithm SHrimp

The procedure of SHRIMP (Alg. 3) is similar to that of KRIMP. The input set of frequent patterns is sorted (line 2) and the tree is initialized (line 3). Since the initial code table is the standard code table that contains only singleton patterns, the initial tree equals the FP-tree. For every candidate pattern, the transactions that would use the candidate are computed and stored by the corresponding leaves called *fpLeaves* (line 6) and the resulting usage function is computed (line 7). If the inclusion of the pattern improves the compression, the pattern is integrated into the tree (line 10).

Algorithm 3 SHrimp.

```

1: procedure SHRIMP( $\mathcal{D}, \mathcal{F}$ )
2:   sort( $\mathcal{F}$ ) ▷ in Standard Candidate Order
3:   tree  $\leftarrow$  initTree( $\mathcal{D}$ )
4:   usage  $\leftarrow$   $\{(item, item.frequency) \mid item \in \mathcal{D}\}$ 
5:   for  $fp \in \mathcal{F} \setminus \mathcal{I}$  do
6:     fpLeaves  $\leftarrow$  USINGTRANSACTIONS( $fp, tree$ )
7:     usagec  $\leftarrow$  USAGEINCLUDING( $fp, fpLeaves, tree$ )
8:     if  $L(usage_c) < L(usage)$  then
9:       usage  $\leftarrow$  usagec
10:    INSERT( $fp, fpLeaves, tree$ )
11:   end if
12: end for
13: end procedure

```

The method USINGTRANSACTIONS($fp, tree$) (line 6) identifies the using transactions in a depth-first like search, exploiting the order of the tree as described in Example 2. For each of the returned leaves, the method DIFFUSAGE (Alg. 4) is called. This procedure considers a branch as an ordered sequence of ancestor nodes of the specified leaf that are succeeding to the candidate pattern fp with regard to the standard cover order (line 4). The set *bounded* (line 5) collects all items that are covered now and the set *freed* (line 6) those that are not covered anymore if fp is inserted. By means of these sets, it is checked for every node of the branch whether it would change its status of activity and the usage function is adapted accordingly (line 7-15).

The method INSERT($fp, tree$) in Alg. 3 (line 10) creates the nodes of the candidate pattern and alters the tree accordingly. The integration of a node induces bounds and branches. Singletons that are covered by the inserted node are removed and a branch might appear more bound as in Example 2. Generally speaking, the concerning branch is divided into these transactions that contain the candidate pattern and the remaining ones. For this reason, the branch of transactions containing the specified pattern has to be split from the original one. The method MODIFICATIONS($fp, fpLeaves$) (Alg. 5) computes thereby the arising structural changes that come with the insertion of fp . This algorithm

Algorithm 4 Computing the resulting usage differences concerning the inclusion of the pattern fp .

```

1: procedure USAGEINCLUDING( $fp, fpLeaves, tree$ )
2:    $usage_c \leftarrow usage(tree)$ 
3:   for  $fpLeaf \in fpLeaves$  do
4:      $branch \leftarrow \{n \in anc(leaf) | fp \prec n\}$        $\triangleright$  sorted in Standard Cover Order
5:      $bounded \leftarrow fp$ 
6:      $freed \leftarrow \emptyset$ 
7:     for  $n \in branch$  do
8:       if  $\emptyset = n.inact \wedge (bounded \cap n.pattern \neq \emptyset)$  then
9:          $usage_c(n.pattern) \leftarrow usage_c(n.pattern) - u$ 
10:         $freed \leftarrow freed \cup n.pattern$ 
11:       else if  $(\emptyset \neq n.inact \subseteq freed) \wedge (bounded \cap n.pattern = \emptyset)$  then
12:          $usage_c(n.pattern) \leftarrow usage_c(n.pattern) + u$ 
13:          $bounded \leftarrow bounded \cup n.pattern$ 
14:       end if
15:     end for
16:   end for
17:   return  $usage_c$ 
18: end procedure

```

traverses the tree from the specified leaves up to the first node that is preceding to fp , the prospective parent node (lines 4-10). For each of the regarded nodes, the modifications of their fields in the branch with the integrated pattern are computed (line 6,7). Accordingly to the gained information, the nodes are created and the tree altered. Inactive nodes are integrated by the same procedure.

4 Experiments

The experiments are invoked as *RapidMiner*¹ processes, based on the JAVA programming language. The implemented operator Krimp is modelled after the original C++ implementation², i.e. transactions are represented as bit vectors, the code words are stored as a list of lists, that enables an immediate access to the insertion point of a candidate pattern, etc. The frequent patterns are mined by the available FP-Growth operator from RapidMiner. Due to storage limitations, the minimum support for the mined candidates differs in dependence of the dataset, which was mainly dependent on the performance of the FP-Growth algorithm. Table 2 shows the basic characteristics of the used datasets and the parameters for the candidate generation. The datasets were taken from the FIMI repository³ and a collection of prepared UCI datasets⁴.

¹ <http://rapidminer.com>

² <http://www.patternsthatmatter.org/implementations/#krimp>

³ <http://fimi.ua.ac.be/data>

⁴ <https://dtai.cs.kuleuven.be/CP4IM/datasets>

Algorithm 5 Computes the modifications of the tree resulting from the integration of the pattern fp .

```

1: procedure MODIFICATIONS( $fp, fpLeaves$ )
2:    $fpParents \leftarrow \emptyset$ 
3:   for  $fpLeaf \in fpLeaves$  do
4:      $n \leftarrow fpLeaf$ 
5:     while  $fp \prec n$  do
6:        $mod(n.freq) \leftarrow mod(n.freq) + fpLeaf.freq$ 
7:        $mod(n.parent.children) \leftarrow mod(n.parent.children) \cup n$ 
8:        $n \leftarrow n.parent$ 
9:     end while
10:     $fpParents \leftarrow fpParents \cup \{n\}$ 
11:  end for
12: end procedure
13: return  $mod$ 

```

The runtime results, comparing KRIMP and SHRIMP, are displayed in Figure 3. We can see that SHRIMP has an exceptionally better performance on the FIMI datasets (Mushroom, Chess and Connect). For these datasets, the minimum support was very hard to set, because a small decrease in the threshold resulted in an enormous growth of frequent patterns that caused the pattern generation process to crash due to an available memory capacity of about 100GiB. Regarding the Soybean dataset, the results are assimiabile well. For about 20% of the first regarded candidates, SHRIMP is slightly faster, but the trend reverses with time. For the rather small Tic-tac-toe and the Tumor dataset, KRIMP is however eminently faster than SHRIMP. It might be noted, that the amount of time spent on the insertion of patterns into the tree is negligible small in comparison to the time needed for the usage calculation of all candidates. Further insights into node dependencies and the usage calculation in general might thus improve the algorithm thoroughly.

| Dataset | $ \mathcal{D} $ | $ \mathcal{I} $ | density | $minsup$ | $ \mathcal{F} $ |
|---------------|-----------------|-----------------|---------|----------|-----------------|
| Tic-tac-toe | 958 | 27 | 33% | 0 | 250985 |
| Soybean | 630 | 50 | 32% | 0.01 | 2613499 |
| Primary-tumor | 336 | 31 | 48% | 0.01 | 1290968 |
| Mushroom | 8124 | 119 | 18% | 0.1 | 574431 |
| Chess(k-k) | 3196 | 75 | 50% | 0.6 | 254944 |
| Connect | 67557 | 129 | 33% | 0.9 | 27127 |

Table 2: Basic characteristics of examined datasets.

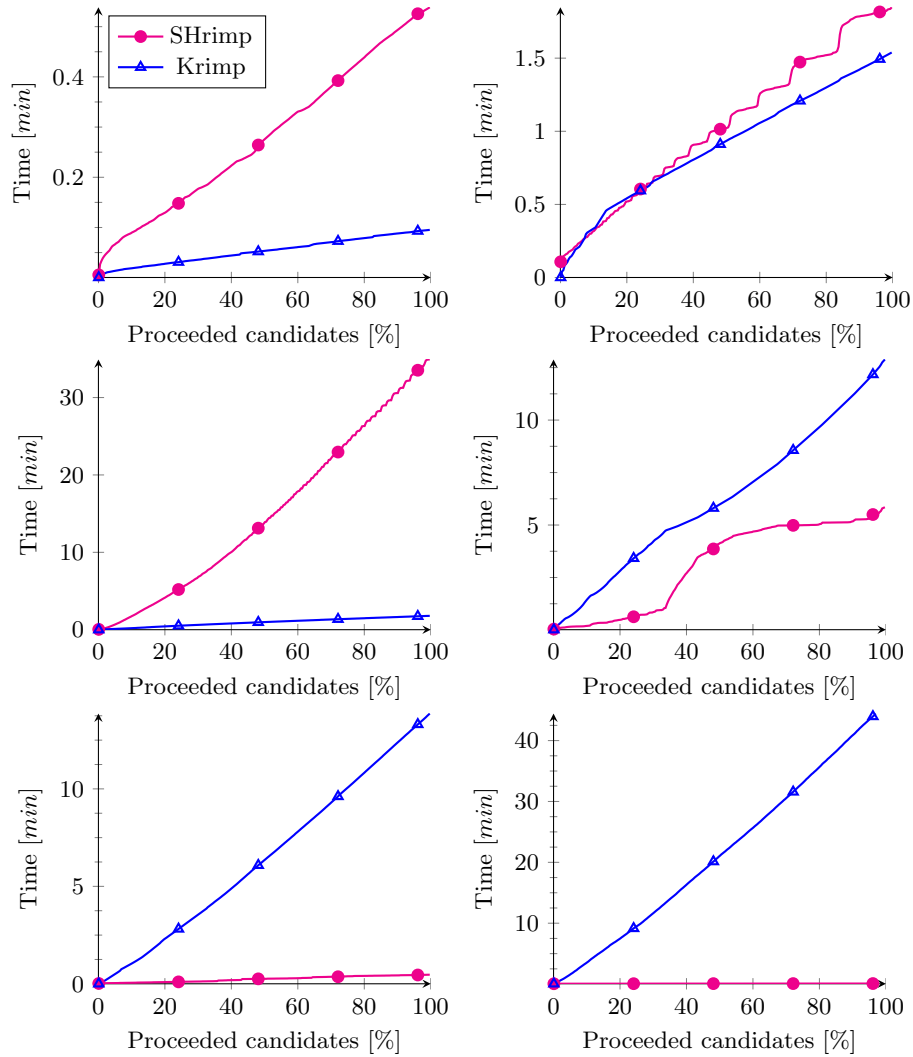


Fig. 3: Runtime for Krimp (blue triangle marks) and SHrimp (red square marks) in relation to the percentage of examined patterns for the Tic-tac-toe (left above), Soybean (right above), Primary-tumor (left middle), Mushroom (right middle), Chess (left below) and Connect (right below) dataset.

5 Conclusion

A first attempt to create a data structure that reflects the possibilities of summarization due to the inherent structure of the dataset has been substantiated with the development of SHRIMP. A prototype implementation facilitates a much faster computation of influences on the compression size when the coding set changes, for at least some well known datasets. This data structure might be applied to other MDL approaches, e.g. SLIM, as well. The structure offers the possibility of a human readable understanding of the main characteristics of the dataset. We can think of an output that reveals only the treetop, e.g. all nodes up to a certain level, where the composition of the most prioritized codes in relation to the standard cover order is displayed.

Further exploitations of the order of the code table, respectively the nodes, and a less redundant representation of codes that are subsets of other codes are likely to improve the results and are worth to be explored.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. *SIGMOD Rec.* 22(2), 207–216 (1993)
2. Cover, T., Thomas, J.: *Elements of information theory*. Wiley-Interscience (2006)
3. Grünwald, P.: *Minimum Description Length Principle*. MIT press, Cambridge, MA (2007)
4. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. *SIGMOD Rec.* 29(2), 1–12 (2000)
5. van Leeuwen, M., Bonchi, F., Sigurbjörnsson, B., Siebes, A.: Compressing tags to find interesting media groups. In: *CIKM*. pp. 1147–1156. ACM (2009)
6. Li, P.V.M.: *An Introduction to Kolmogorov Complexity and Its Applications*. Springer (1997)
7. Mannila, H., Toivonen, H.: *Levelwise search and borders of theories in knowledge discovery* (1997)
8. Rissanen, J.: Modeling by shortest data description. *Automatica* 14, 465–471 (1978)
9. Smets, K., Vreeken, J.: Slim: Directly mining descriptive patterns. In: *SDM*. pp. 236–247. SIAM / Omnipress (2012)
10. Vreeken, J., van Leeuwen, M., Siebes, A.: Characterising the difference. In: *KDD*. pp. 765–774. ACM (2007)
11. Vreeken, J., van Leeuwen, M., Siebes, A.: Preserving privacy through data generation. In: *ICDM*. pp. 685–690. IEEE Computer Society (2007)
12. Vreeken, J., van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. *Data Min. Knowl. Discov.* 23, 169–214 (2011)

Novel Criteria to Measure Performance of Time Series Segmentation Techniques

André Gensler, Bernhard Sick

Intelligent Embedded Systems
University of Kassel
Kassel, Germany
Email: {gensler, bsick}@uni-kassel.de

Abstract. An important task in signal processing and temporal data mining is time series segmentation. In order to perform tasks such as time series classification, anomaly detection in time series, motif detection, or time series forecasting, segmentation is often a pre-requisite. However, there has not been much research on evaluation of time series segmentation techniques. The quality of segmentation techniques is mostly measured indirectly using the least-squares error that an approximation algorithm makes when reconstructing the segments of a time series given by segmentation. In this article, we propose a novel evaluation paradigm, measuring the occurrence of segmentation points directly. The measures we introduce help to determine and compare the quality of segmentation algorithms better, especially in areas such as finding perceptually important points (PIP) and other user-specified points.

1 Introduction and State of the Art

An important task in signal processing and temporal data mining is *time series segmentation*, the division of a time series in a sequence of segments. In order to perform tasks such as time series classification, anomaly detection in time series, motif detection, or time series forecasting, segmentation is often a pre-requisite. Depending on the application, segmentation can have arbitrary goals which can basically be divided in two sub-categories.

Segmentation for time series reconstruction and representation

The first category is segmentation for time series reconstruction and representation purposes. This category often uses algorithms which evaluate the approximation error in some form and often aim at representing a time series by a series of linear approximations. The existing algorithms can be categorized in

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

one of the two categories off-line or on-line segmentation [13]. Off-line algorithms have a global view on the data, they therefore know the development of the data points, and, in theory, can achieve better results than on-line techniques. Popular examples for off-line techniques include the Top-Down or the Bottom-Up Segmentation [14], or the k -Segmentation [4], which performs a perfect segmentation of a time series given k segments and an objective function (error function) in very high computing time. On-line techniques only have a local view on the data and have to decide about executing a segmentation without knowing the future development of data points. For many use cases, such as applications with harsh timing constraints, real-time applications, or the processing of large amounts of data, only on-line techniques are applicable. In this realm, the Sliding Window and Bottom-Up (SWAB) algorithm (and variants of it) is widely used [1, 14, 20]. Surveys comparing several time series segmentation techniques can be found in [8, 9, 14, 15]. It can be observed, that most segmentation algorithms evaluate the quality of a segmentation of a time series by the least-squares reconstruction error that an approximation algorithm makes when approximating the segments of a time series [7, 10, 14, 16, 17].

Segmentation at characteristic points of the time series

The second category contains algorithms which aim at performing a segmentation when the characteristics of the time series change in a certain way. This category contains applications, such as segmentation for higher efficiency, indexing long time series, or finding perceptually important points (PIP) [6] and other user-specified points. An algorithm of this category is for example the Sliding Window algorithm, which can deliver reasonably good results [10] very fast using systems of orthogonal polynomials [12]. Various error criteria can be used in this approach to determine the segmentation points, e.g., the approximation error or combinations of polynomial coefficients, such as slope or curvature.

However, there has been little research in the field of evaluation of segmentation in this realm. The frequently used reconstruction error measure is not optimal, as it usually declines with an increasing number of segments (though more segments usually are not necessarily related to a good segmentation) and highly depends on the approximation algorithm and its parameters. Furthermore, it also is an indirect measure, as it only rates the quality of the reconstruction rather than the segmentation points itself. To the best of our knowledge, there does not exist a measure that determines the quality of a segmentation with respect to points the user actually *wants* the algorithm to segment at. Therefore, we introduce a scheme for the evaluation which aims at quantifying the segmentation results with respect to user-defined segmentation points (i.e., labeled points).

The remainder of this article is organized as follows: In Section 2, we introduce a novel interpretation of evaluation of segmentation by treating the segmentation result as a classification problem. We then discuss measures which make sense in order to quantify the segmentation results. Section 3 discusses

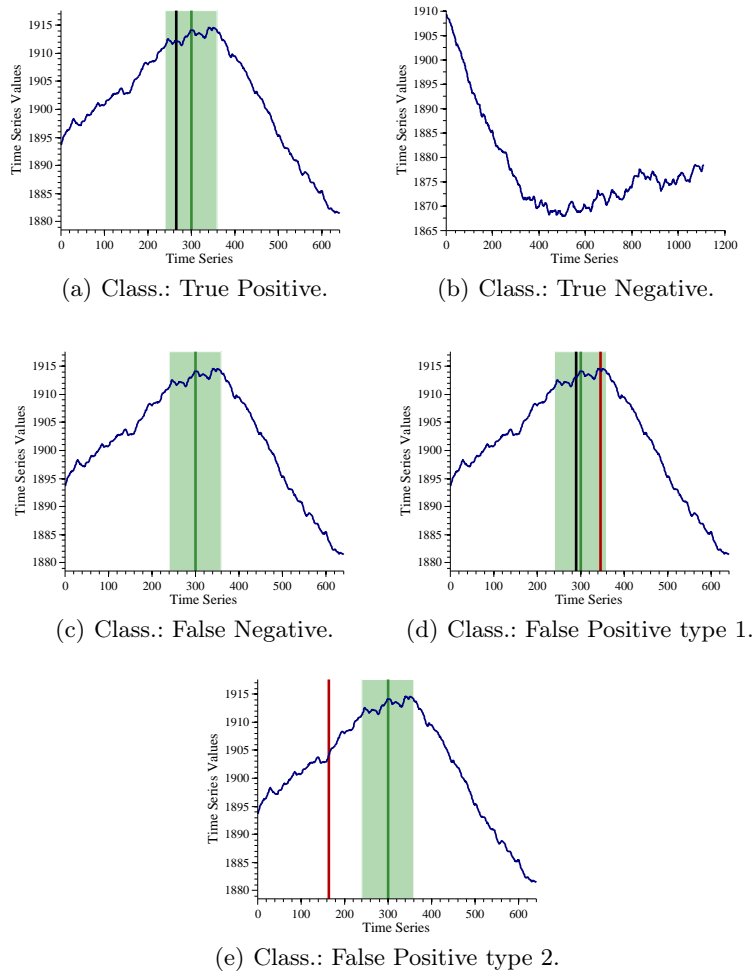


Fig. 1. Cases in the classification of segmentation points. The green vertical lines represents the segmentation zone (SZ) center while the green areas show the valid SZ area. The black and red lines represent segmentation points determined by a segmentation algorithm. While the black lines represent valid segmentation points (true positives), the red lines are false positives. The goal of a segmentation is to hit an SZ close to the SZ center exactly one time while not producing any segmentation point outside an SZ.

further measures not directly related to the classification interpretation, but nevertheless describe properties of segmentation results. In Section 4, we aim at giving further insight in how these measures work by conducting some experiments and evaluating our measures for the experimental results. Section 5 wraps up our findings.

2 Classification Related Measures

The segmentation of time series can be seen as a classification problem: In each time step, the algorithm has to decide whether to perform a segmentation (S^+) or not (S^-). Consequently, by comparing the performed segmentation to a user-specified target one, a standard metric such as a confusion matrix (see Table 1) can be applied. The algorithm will return a vector of predicted labels \mathbf{p} , e.g., $\mathbf{p} = \{S^-, S^-, S^-, S^+, \dots, S^-\}$ for each evaluated time series with length N . For the evaluation of a performed time series segmentation, the criteria to determine the elements of the confusion matrix (shown in Table 1) differ from those of a standard classification task.

| | | Ground Truth | | Total |
|------------|----------|--------------|----------|-------|
| | | Positive | Negative | |
| Prediction | Positive | TP | FP | TP+FP |
| | Negative | FN | TN | FN+TN |
| Total | | TP+FN | FP+TN | |

Table 1. A standard confusion matrix. Here, we propose an interpretation of the confusion matrix (normally used for standard classification tasks) for segmentation problems.

In a naïve approach, every point in time is given a ground truth label to form a vector $\mathbf{t} \in \{S^+, S^-\}^N$ with only a very small amount of target segmentation points S^+ . For the sake of simplicity, our time series here is assumed to consist of equidistant data points in the time domain (though this is not required for the evaluation). By comparing \mathbf{p} and \mathbf{t} pairwise, the confusion matrix can be formed. This approach, though, does not account for temporal adjacency. For most applications, a segmentation $\mathbf{p}(n) = S^+$ at point in time n of the time series would be considered as a good-enough hit if the target segmentation point was located in the immediate neighborhood $\mathbf{t}(n \pm \epsilon) = S^+$, ($\epsilon \in \mathbb{N}^+$, ϵ is a tolerated deviation). But, in our naïve approach, such a result would lead to both a false positive (FP) and false negative (FN) result as they do not match exactly. Therefore, the evaluation metric has to be modified to incorporate temporal neighborhood in a small area around a target segmentation point as a valid segmentation. Additionally, the evaluation result depends not only on a single segmentation decision in time, but on the result in conjunction with the predicted labels in the temporal neighborhood, i.e., while one segmentation at the right location is desirable, multiple segmentation points at the same location have to be penalized.

Depending on the nature of the time series and the desired application, every segmentation task has its own requirements regarding its temporal accuracy of the segmentation, e.g., for some tasks, a too early segmentation may be unproblematic while a late segmentation must not be allowed. For an evaluation it

Algorithm 1 Calc. of Average Segmentation Count (*ASC*)

```
procedure CALCULATEASC(Segmentation Points, All SZ)
  Create counter variable  $v$  for every SZ
  for each Segmentation Point of Segmentation Points do
    if Segmentation Point is inside SZ then
      Increment variable  $v$  of this SZ by 1
    end if
  end for
  return Sum of all  $v$  divided by total number of SZ
end procedure
```

therefore makes sense to model each segment to be determined with an earliest and latest point which will still be considered as inside an allowed *segmentation zone* (*SZ*). Fig. 1(a) visualizes a sample segmentation zone design, the green vertical line represents the target segmentation point while the green area shows the size of the SZ. The black and red lines represent segmentation points determined by an arbitrary algorithm. In the shown case, one SZ is assigned exactly one determined segmentation point, therefore it is treated as a true positive (TP). Another simple case is shown in Fig. 1(b). If the algorithm does not set a segmentation point in an area where none is expected, the sample is treated as a true negative (TN). If a SZ is not detected, i.e., no segmentation point is associated with it, it is treated as a false negative (FN, type II error), (see Fig. 1(c)). False positives (FP, type I error) can be created in two situations: (1) An SZ is assigned more than one segmentation point, each segmentation point more than one is then treated as a FP, Fig. 1(d). (2) A segmentation point is found in an area where no SZ exists, Fig. 1(e).

Due to the fact that in most cases there will be a lot more elements where there is no segmentation (S^-) than elements where there is a segmentation (S^+), we can assume a heavily imbalanced dataset regarding the class distribution, invalidating basic measures such as accuracy defined by

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \quad (1)$$

as it does not consider the distribution of the classes at all. A well-known measure to describe classification performance is the Receiver-Operating-Characteristic (ROC) curve, describing the development of the false-positive rate (FPR)

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (2)$$

and the true-positive rate (TPR)

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (3)$$

This measure is also valid for imbalanced datasets, though only a small area of the total ROC curve is covered. The FPR remains problematic as it will adopt only small values.

Algorithm 2 Calc. of Absolute Segmentation Distance (*ASD*)

```
procedure CALCULATEASD(Segmentation Points, All SZ)
  Initialize variable ASD with 0
  for each Segmentation Point of Segmentation Points do
    if Segmentation Point is inside SZ then
      Add Dist. between Segmentation Point and this SZ center to ASD
    end if
  end for
  return ASD divided by number of found Segmentation Points
end procedure
```

Other prominent measures describing the confusion matrix in one single value beside accuracy are the Area Under (ROC) Curve (*AUC*) and F_n -score measures such as the F_1 score (both evaluated in [19]) or the Matthews correlation coefficient (*MCC*). The F_1 score calculated by

$$F_1 = \frac{2TP}{(2TP + FP + FN)} \quad (4)$$

describes the harmonic mean of precision and sensitivity, which in turn means that the amount of TN is not taken into account. The *MCC* published in [18] calculates a correlation of a two-class classification prediction and is regarded as a balanced measure which can even be used if the class sizes are very different [3]. The *MCC* takes into account *all* elements of the confusion matrix and is calculated by

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \quad (5)$$

It adopts values from -1 to $+1$, where $+1$ represents perfect correlation, 0 a result not better than random guess and -1 absolute disagreement between prediction and ground truth. In contrast to the F_1 score, *MCC* also incorporates the TN, thus representing the overall structure of the confusion matrix in more detail. In general, it is regarded as a good measure for unbalanced classification problems [3].

To determine the overall quality of a segmentation result, the F_1 score and the *MCC* seem to be the most promising measures, they have different advantages and disadvantages, though they behave similar in general [2]. The *MCC* does not just account for (in)correct predictions, but measures correlation, which means that it takes into account systematic mispredictions by adopting values smaller than 0 . This can be seen as an advantage for the *MCC*. Furthermore, it considers all elements of the confusion matrix, consequently representing the classification result in more detail. While this sounds appealing for standard classification tasks, for segmentation the incorporation of the standard case “TN” may turn out as a distracting factor in the evaluation. Depending on the nature of the data, segments are set in different frequencies, resulting in a different proportion

Algorithm 3 Calc. of Average Direction Tendency (*ADT*)

```
procedure CALCULATEADT(Segmentation Points, All SZ)
  Initialize variables PreSeg and PostSeg with 0
  for each Segmentation Point of Segmentation Points do
    if Segmentation Point inside SZ then
      Add 1 to PostSeg if after SZ center or
      add 1 to PreSeg if before SZ center
    end if
  return PostSeg / (PostSeg + PreSeg)
end for
end procedure
```

of positives and negatives, modifying the *MCC*. In addition, factors such as the sampling rate of a time series can have an impact on the *MCC*, resulting in more negatives only (doubling the sampling rate of the same process leads to about twice as many TN). In other words, while the *MCC* considers the whole time series, the F_1 score just accounts for what the segmentation algorithm does (correct and false). Consequently, we think that the F_1 score is more appropriate for the evaluation of most segmentation tasks. Depending on the segmentation task, other forms of the F_n score can make sense (e.g., the F_2 score putting twice as much emphasis on recall). When it comes to adjusting a segmentation algorithm to different applications (with different constraints regarding FN and FP), it also makes sense to use *Precision* and *Recall*

$$\text{Prec.} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (6)$$

$$\text{Rec.} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (7)$$

To set the operating point of an algorithm, the goal may not be the overall optimal classification performance (e.g., regarding F_1 score), but achieving the best possible performance when constraining one type of error. These two measures give insights on how an algorithm performs regarding only type I or type II error, respectively, and thereby help to determine the desired operating point.

3 Segmentation Zone Measures

Besides the measurements related to classification, there exist also other properties of segmentation algorithms which are worth examining. An important measure is the Average Segmentation Count (*ASC*), determining how many times an algorithm triggers a segmentation while being inside a SZ on average. The *ASC* can be calculated by Algorithm 1. The value of *ASC* ideally is close to 1, a value lower 1 means too little segments are set inside SZ while a value greater 1 means too many segments are found. It is normed by the SZ count, resulting in an easily understandable result (“per SZ, the algorithm sets *ASC* segments on average”).

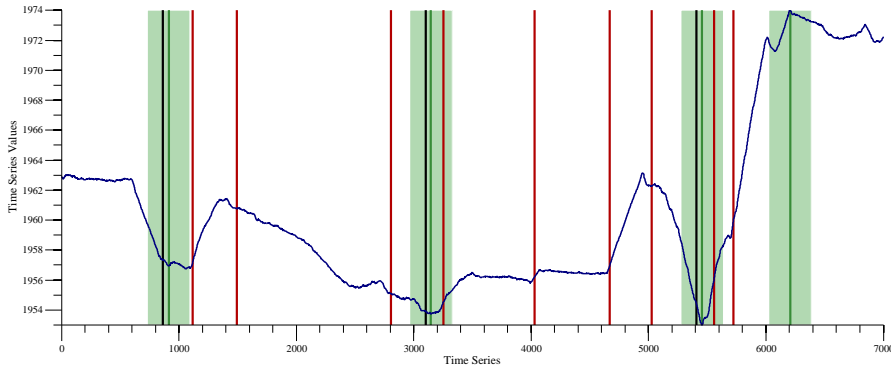


Fig. 2. Segmentation experiment *I* using excerpt from a real data example [5] (symmetric SZ with total size 180). A basic Sliding Window algorithm using a fast polynomial approximation [11] is utilized (Window Size = 90; Segmentation Criteria: *average* < 1970, *slope* < 5, *curvature* > 10^{-4} , re-segmentation suppression 150 steps). The segmentation points colored in black represent true positives while the red segmentation points represent false positives. The last Segmentation Zone is not hit at all, resulting in a false negative. The confusion matrix that can be calculated by summing up the four respective cases is shown in Table 2.

Furthermore, not only the number of segmentation points is important, but also how accurately they hit the target segmentation. To determine the distance between target segmentation point and found segmentation point, we introduce a measure called Absolute Segmentation Distance (*ASD*) calculated by Algorithm 2. It is normed by the number of segmentation points found. Finally, it could be of interest whether an algorithm tends to set its segmentation points too early or too late. To specify this characteristic of a segmentation algorithm, we introduce a measure called Average Direction Tendency (*ADT*) which is described in Algorithm 3. It describes a quotient of early and late segmentation points (“*ADT*% of segmentation points are too late”). If the algorithm tends to set its segments too early, the value will be below 0.5 while a late segmentation will result in a value greater 0.5. It is useful to use this measure in conjunction with the *ASD* measure to quantify the amount of segments and the direction of the deviation.

4 Exemplary Evaluation

Now we want to briefly show the measures in action to get a better overall impression of how the measures perform. In order to do that, we extracted a time series from a real data set [5] showing activity data with a chest-mounted accelerometer and defined segmentation zones which we expect our segmentation algorithm to find. We used a basic Sliding Window segmentation algorithm evaluating polynomial approximations of the window content using fast update

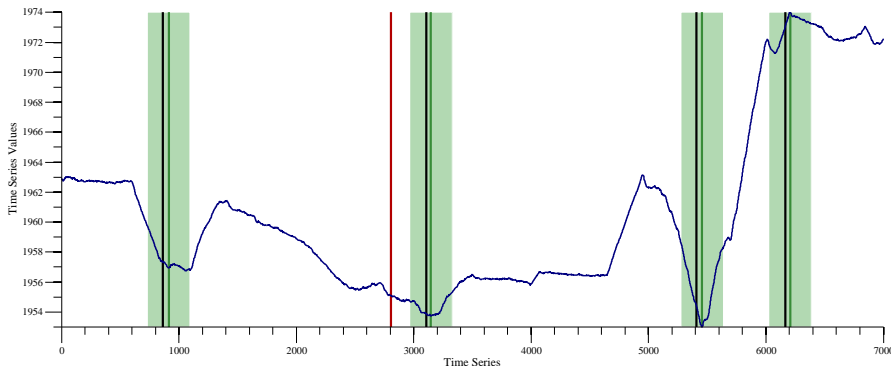


Fig. 3. Segmentation experiment *II* using excerpt from a real data example [5] (symmetric SZ with total size 180). A basic Sliding Window algorithm using a fast polynomial approximation [11] is utilized with two criteria (Window Size = 90; Segmentation Criteria: (1) *average* < 1958, *slope* < 10^{-4} , *curvature* > 10^{-4} , re-segmentation suppression 300 steps; (2) *average* > 1972, *slope* > 10^{-2} , re-segmentation suppression 200 steps). The segmentation points colored in black represent true positives while the red segmentation points represent false positives. It can be seen easily that this experiment produces less false positives and no SZ is missed. The confusion matrix that can be calculated by summing up the four respective cases is shown in Table 3. It can be expected that in the evaluation an improvement in relation to segmentation experiment *I* (Fig. 2) can be observed.

formulas as proposed in [10]. The capabilities of the approximation regarding run-time can be found in [12], an implementation of the approximation algorithm can be downloaded at [11]. To show how the measures behave, we performed the segmentation with two parameter combinations, one of which performs significantly better. The result of the time series for the first (worse) parameter combination (Experiment *I*) is shown in Fig. 2. As we can see from the image, the algorithm tends to set too many segmentation points, some of which are not inside a specified segmentation zone. The points outside the zones are counted as False Positives (FP). Additionally, some segmentation zones are hit multiple times. While the black (valid) segmentation points are counted as True Positives (TP), all further segmentation points are also treated as FP. Furthermore, one of the segmentation zones is not hit at all. This zone is counted as a false negative (FN). From the segmentation results, we can create a confusion matrix as shown in Table 2. Next, we performed the segmentation with a different, apparently better parameter combination (Experiment *II*, Fig. 3). All zones are hit exactly one time, every segmentation therefore counts as exactly one TP. We can see, that one determined segmentation point lies outside a segmentation zone. Consequently, it is treated as a FP. The confusion matrix for this segmentation is shown in Table 3.

| | | Ground Truth | |
|------------|----------|--------------|----------|
| | | Positive | Negative |
| Prediction | Positive | 3 | 9 |
| | Negative | 1 | 6987 |

Table 2. Example confusion matrix resulting from the segmentation performed in Fig. 2. Standard performance measures can now be applied to the matrix.

| | | Ground Truth | |
|------------|----------|--------------|----------|
| | | Positive | Negative |
| Prediction | Positive | 4 | 1 |
| | Negative | 0 | 6995 |

Table 3. Example confusion matrix resulting from the segmentation performed in Fig. 3. Standard performance measures can now be applied to the matrix.

To the confusion matrix elements, we can now apply the classification related measures described in Section 2. For both confusion matrices 2 and 3, we evaluated the Accuracy (ACC), the F_1 score, and the Matthews Correlation Coefficient (MCC). Additionally, we applied our new segmentation zone measures to the segmentation results, namely the Average Segmentation Count (ASC), the Absolute Segmentation Distance (ASD) and the Average Direction Tendency (ADT). The results are shown in Table 4. In addition we added some baseline results for algorithms performing a segmentation at no point in time (NeverSeg), on every time step (AlwaysSeg) or on random with $p(S^+) = 0.5$. In this table, the classification related measures are shown on the left hand side, while the segmentation zone measures are shown on the right hand side of the table. As we can clearly see, ACC is unable to discriminate between the results of experiment *I* and *II*. The $Precision$ drastically increases from a value of 0.25 to 0.80. The $Recall$ also increases from 0.75 to 1.00, as no FP are produced in Experiment *II*. The F_1 score has a range between 0 and 1, here the values are 0.375 or 0.888, respectively. We can see a clear difference between result *I* and *II*. The MCC behaves numerically similar: In experiment *I*, the value is 0.433, while experiment *II* results in a value of 0.894. Both the F_1 score and the MCC behave very similar here. In the realm of the segmentation zone measures, we can see that the algorithm behaves more appropriate with respect to the specified SZ in experiment *II*. The ideal value of ASC is 1 (one found segment for each SZ). While experiment *I* returned too many segments, segmentation *II* yields better results. For the ASD measure, we can see that the segmentation became more accurate regarding the SZ center. It improved from about 70 data samples from the center to only 45 samples on average. Last, the ADT also changed, though this is not a measure for segmentation quality, but more a description of the algorithm characteristics: In experiment *I*, the quotient between early and late segmentation points was roughly balanced (0.5 would be perfectly balanced) with a value of 0.4, which means a slight overweight for early segmentation points. In experiment *II*, all segmentation points were too early.

| | <i>Acc.</i> | <i>Prec.</i> | <i>Rec.</i> | F_1 | <i>MCC</i> | <i>ASC</i> | <i>ASD</i> | <i>ADT</i> |
|-----------|-------------|--------------|-------------|-------|------------|------------|------------|------------|
| NeverSeg | 0.999 | 1.000 | 0.000 | 0.000 | 0.000 | 0.0 | 0.00 | - |
| RandomSeg | 0.500 | 0.001 | 1.000 | 0.001 | 0.000 | 90.0 | 90.00 | 0.5 |
| AlwaysSeg | 0.001 | 0.001 | 1.000 | 0.001 | 0.001 | 180.0 | 90.00 | 0.5 |
| Exp. (I) | 0.999 | 0.250 | 0.750 | 0.375 | 0.433 | 1.25 | 70.40 | 0.4 |
| Exp. (II) | 0.999 | 0.800 | 1.000 | 0.888 | 0.894 | 1.00 | 44.75 | 0.0 |

Table 4. Segmentation measures extracted from experiments of Fig. 2, Fig. 3, and the respective confusion matrices (Table 2, Table 3). Additionally, some baseline measures were added, showing algorithms performing segmentation at no point in time (NeverSeg), on every time step (AlwaysSeg) or random with $p(S^+) = 0.5$ (RandomSeg). As we can see, the accuracy (*ACC*) clearly falls short of describing the segmentation result. *Precision* and *Recall* differ significantly, both favoring Experiment *II*. The F_1 score and the *MCC* behave similarly, though the *MCC* takes into account the TN in contrast to F_1 . The other measures *ASC* and *ASD* improve from Experiment *I* to *II*. The segmentation characteristic of the algorithm changes as well, from a balanced segmentation to an early segmentation, as *ADT* describes.

All in all, these measures help to quantify the quality of a segmentation result. Depending on the application of the segmentation, different measures may be more or less important. Often, a combination of multiple measures helps to specify the characteristics of the segmentation algorithm.

5 Conclusion and Outlook

In this article, we proposed several new evaluation criteria for time series segmentation in the realm of segmentation for the sake of finding specific points such as perceptually important points (PIP), which we categorized in classification related measures and segmentation zone measures. We think our new measures will help to compare the quality of various segmentation approaches better, especially for applications such as motif detection and other applications where the detection of user-defined points turn out to be important. We hope authors will adopt these measures to further increase the comparability of segmentation algorithms. In our future work we aim to evaluate new algorithms for time series segmentation using the introduced measures. We also thought about defining gradual segmentation zones (e. g., by using gaussians) to further specify the quality of a segmentation algorithm.

References

1. Amft, O., Junker, H., Tröster, G.: Detection of eating and drinking arm gestures using inertial body-worn sensors. In: Proceedings of 9th IEEE International Symposium on Wearable Computers. pp. 160–163. IEEE, Osaka, JP (2005)
2. Arora, A.: Matthews Correlation Coefficient - How well does it do? <http://standardwisdom.com/softwarejournal/2011/12/>

- [matthews-correlation-coefficient-how-well-does-it-do/](#), last access 07/02/2014
3. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C., Nielsen, H.: Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* 16(5), 412–424 (2000)
 4. Bellman, R.: On the approximation of curves by line segments using dynamic programming. *Communications of the ACM* 4(6), 284 (1961)
 5. Casale, P., Pujol, O., Radeva, P.: Activity recognition from single chest-mounted accelerometer data set. <https://archive.ics.uci.edu/ml/datasets/Activity+Recognition+from+Single+Chest-Mounted+Accelerometer>, last access 06/30/2014
 6. Chung, F.L., Fu, T.C., Luk, R., Ng, V.: Flexible time series pattern matching based on perceptually important points. In: *Proceedings of 17th International Joint Conference on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data*. pp. 1–7 (2001)
 7. Chung, F.L., Fu, T.C., Ng, V., Luk, R.: An evolutionary approach to pattern-based time series segmentation. *IEEE Transactions on Evolutionary Computation* 8(5), 471–489 (2004)
 8. Esling, P., Agon, C.: Time-series data mining. *ACM Computing Surveys (CSUR)* 45(1), 12–48 (2012)
 9. Fu, T.: A review on time series data mining. *Engineering Applications of Artificial Intelligence* 24(1), 164–181 (2011)
 10. Fuchs, E., Gruber, T., Nitschke, J., Sick, B.: Online segmentation of time series based on polynomial least-squares approximations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(12), 2232–2245 (2010)
 11. Gensler, A., Gruber, T., Sick, B.: Fast Approximation Library. <http://ies-research.de/Software>, last access 05/28/2014
 12. Gensler, A., Gruber, T., Sick, B.: Blazing fast time series segmentation based on update techniques for polynomial approximations. In: *Proceedings of 13th IEEE International Conference on Data Mining Workshops (ICDMW13)*. pp. 1002–1011. IEEE, Dallas, USA (2013)
 13. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. In: *Proceedings of 1st IEEE International Conference on Data Mining (ICDM2001)*. pp. 289–296. IEEE, San Jose, USA (2001)
 14. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: A survey and novel approach. *Data mining in time series databases* 57, 1–22 (2004)
 15. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery* 7(4), 349–371 (2003)
 16. Lemire, D.: A better alternative to piecewise linear time series segmentation. In: *SIAM International Conference on Data Mining*. pp. 545–550. SIAM (2007)
 17. Liu, X., Lin, Z., Wang, H.: Novel online methods for time series segmentation. *IEEE Transactions on Knowledge and Data Engineering* 20(12), 1616–1626 (2008)
 18. Matthews, B.W.: Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405(2), 442–451 (1975)
 19. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. *Information Processing & Management* 45(4), 427–437 (2009)
 20. Van Laerhoven, K., Berlin, E., Schiele, B.: Enabling efficient time series analysis for wearable activity data. In: *Proceedings of 8th International Conference on Machine Learning and Applications (ICMLA'09)*. pp. 392–397. IEEE, Miami, USA (2009)

Mining Implications From Data

Ahcène Boubekki and Daniel Bengs

Deutsches Institut für Internationale Pädagogische Forschung
Frankfurt am Main, Germany
{boubekki, bengs}@dipf.de

Abstract. Item Tree Analysis (ITA) can be used to mine deterministic relationships from noisy data. In the educational domain, it has been used to infer descriptions of student knowledge from test responses in order to discover the implications between test items, allowing researchers to gain insight into the structure of the respective knowledge space. Existing approaches to ITA are computationally intense and yield results of limited accuracy, constraining the use of ITA to small datasets. We present work in progress towards an improved method that allows for efficient approximate ITA, enabling the use of ITA on larger data sets. Experimental results show that our method performs comparably to or better than existing approaches.

1 Introduction

Systematic implications between variables in datasets arise whenever the generating variables are correlated and are at the heart of almost any data analysis procedure. For instance, in the analysis of sales data, knowledge about which products are usually bought together is beneficial in deducing marketing strategies, in the social sciences hierarchical relations in questionnaire data can uncover structures of underlying traits, and in the field of educational data mining knowledge requirements for solving test items can be revealed. We consider the case where the underlying variables form a strict hierarchy, that is, there are deterministic implications between variables. For instance, in educational testing, a testee who solves a difficult test item is very likely to solve all easier items as well. Similarly, in the case of questionnaires in the social sciences, items are often formulated as statements that relate to a latent trait. Here it is natural to expect that agreement with a strong statement implies agreement with all weaker statements.

When responses to test or questionnaire items are observed in a realistic setting, due to random errors in the measurement process, guessing and careless

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

mistakes by testees, response data invariably exhibits answers that are inconsistent with the item hierarchy, making the implications impossible to observe directly. The challenge is then to reconstruct implications from noisy data.

Formally, the problem statement is the following: Consider a finite set I of n binary items, taking values in $\{0, 1\}$. If it holds that variable $j = 1$ whenever variable $i = 1$, we say that i implies j and write $i \sqsubseteq j$. We require the implications to be logically consistent, that is, the assertion of transitivity holds:

$$i \sqsubseteq j \text{ and } j \sqsubseteq k \implies i \sqsubseteq k \tag{1}$$

As each item implies itself, $i \sqsubseteq i$ for all items i , the relation \sqsubseteq is a quasi-order on I . During the measurement process, patterns that obey \sqsubseteq are perturbed by random noise. The implication mining algorithm aims to reconstruct the original quasi-order as closely as possible.

The problem has first been considered by Van Leeuwe [10], who introduced an algorithm called Item Tree Analysis (ITA). Schrepp [6, 8] suggested to construct implications inductively, showing that his method was more accurate than the original algorithm. Sargin and Ünlü [5] also proposed improvements to the inductive ITA algorithm. Still, the state-of-the-art algorithms for item tree analysis are limited in terms of accuracy and computational feasibility for large datasets. In this paper, we present modifications to the inductive ITA algorithm that lead to significantly reduced execution times and increased accuracy.

Association rule mining (e.g. [1], [2]) is related to ITA, as both methods seek to uncover asymmetric relations between items. Association rule mining aims at finding local hierarchies in the data, while ITA builds a global one, meaning that implications need to hold for all cases in the data set, with exceptions being attributed to random noise. In contrast, association rules can be acceptable if they hold for a minimum fraction of cases. Consequently, the difference is what criterion is used to evaluate the relations. The reliability of an ITA implication $i \Rightarrow j$ is given by the probability $P(\neg i \vee j)$, while for an association rule $i \rightarrow j$ the confidence criterion is related to the probability $P(i \wedge j | i)$, which can be expressed as $P(i \wedge j | i) = P(\neg i \vee j | i)$. We compare the results of our algorithm to association rules mined using the `apriori` algorithm [2] of the `arules` package for R [4].

2 Inductive Item Tree Analysis

We will proceed by explaining the current approach as used by [5] based on the algorithm described by [6]. In section 3 we show how both steps can be improved and introduce our algorithm. First, let us set some notation that will be used in the rest of the paper.

2.1 Preliminaries

A binary matrix, Q , is a matrix with coefficients equal to 0 or 1. For example, the incidence matrix of a relation is a binary matrix. A pattern p from such a matrix is defined by :

$$p = \sum_{i=1}^n a_i q_i$$

where q_i are the lines of the matrix and $(a_i) \in \{0, 1\}^n$. The set of all possible patterns, $pattern(Q)$, is obtained by considering all the possible values of the binary vector (a_i) . However the duplicates are considered only once.

Let us consider a set I of n properties and $\mathcal{D} = \{d_1, \dots, d_m\}$ a data set of m observations of these properties. It can be seen as a $n \times m$ binary matrix. From an educational point of view, the columns are the items of a test and the rows represent answers of the students. We define $p_i = |\{s | d_s[i] = 1\}|$ as the number of observations having the property i , and $b_{i,j} := |\{s | d_s[i] = 1 \wedge d_s[j] = 0\}|$ as the number of observations contradicting $\neg i \vee j$. Denote by $(\beta_L)_{L=1}^{n^2}$ the sequence of $b_{i,j}$ in ascending order.

The data generation process is the following: Starting with a quasi-order Q on a fixed number of items, first the exhaustive set of possible pattern, $pattern(Q)$, is constructed. As Q is reflexive, $pattern(Q)$ contains the n -vectors $\mathbb{1}_n$ and 0_n . The data set is then generated from a collection of patterns by adding noise, that is, flipping coefficients with a prescribed probability τ .

An implication between two properties i and j can be written as a disjunctive logic expression : $i \implies j$ is equivalent to $\neg i \vee j$ and its negation is $i \wedge \neg j$. The more the relation is satisfied in the data set, the more the implication is likely ; or by duality : the more the negation is contradicted, the more it is likely. As the dual formulation is a conjunctive expression, it is easier to extract, which is why it is commonly used to evaluate the confidence in the relation. For two items i, j the number of times the implication $i \implies j$ is contradicted is given by $b_{i,j}$. So the smaller is $b_{i,j}$, the more likely is the relation $i \implies j$.

2.2 Inductive ITA Algorithm

The inductive approach to ITA due to [8] is a two-step procedure:

1. Generate candidate set \mathcal{C}
2. Select the best fitting quasi-order

The first step given by Schrepp [6] is a recursive algorithm, as it uses the relation \sqsubseteq_L in the generation of \sqsubseteq_{L+1} , finally returning the exhaustive set of up to $n(n-1) + 1$ candidate quasi-orders.

Original Candidate Set Generation

- Initialisation : $\sqsubseteq_0 = \{(i, j) \mid b_{i,j} = 0\}$ which is a quasi-order.
 - Suppose \sqsubseteq_L is a candidate quasi-order.
 - Build $A_{L+1} = \{(i, j) \mid b_{i,j} \leq \beta_{L+1} \text{ and } (i, j) \notin \sqsubseteq_L\}$.
 - Remove all elements of A_{L+1} causing intransitivity in $\sqsubseteq_L \cup A_{L+1}$.
 - Set $\sqsubseteq_{L+1} = \sqsubseteq_L \cup A_{L+1}$.
-

The second step is the selection of a relation according to a measure of goodness of fit. In [5] the authors suggested the following method based on a supposed expected numbers of contradictions $b_{i,j}^*$, then the quasi-order fitting best to the observations is selected as follows:

Original Fit

For each quasi-order \sqsubseteq in the candidate set.

- Compute $\gamma = \frac{\sum_{\substack{i \sqsubseteq j \\ i \neq j}} \frac{b_{i,j}}{p_j}}{|\sqsubseteq| - n}$.
 - For each pair (i, j) , determine $b_{i,j}^*$:
 - if $i \sqsubseteq j$, then $b_{i,j}^* = \gamma p_j$
 - if $i \not\sqsubseteq j$ and $j \sqsubseteq i$, then $b_{i,j}^* = p_j - p_i + p_i \gamma$
 - if $i \not\sqsubseteq j$ and $j \not\sqsubseteq i$, then $b_{i,j}^* = (1 - p_i/m)p_j$
 - Evaluate $diff(\sqsubseteq) = \frac{\sum_{i \neq j} (b_{i,j} - b_{i,j}^*)^2}{n(n-1)}$.
 - Return $\text{argmin} diff(\sqsubseteq)$
-

3 Critique and Refinements

Up to now, only the step 2 has been criticized and improved by [5], although at least two points of the first step also need consideration. There are three points that we will address: First, the number of candidates can be reduced by only selecting the most salient quasi-orders, for which we propose a principled way. Second, the way transitivity is enforced in the original algorithm by removing offending pairs depends on the order of removal which is not controlled. We propose a modification to reduce the dependency on the order by reintegrating previously removed pairs. Third, concerning the asymmetry of the fitting coefficient has even been reinforced by the modifications proposed in [5]. To overcome this problem, we propose a new fitting coefficient. To support the discussion, we consider an example using a dataset of size $m = 1000$ created from a synthetic quasi-order on 9 items as described above.

3.1 Selecting the Candidates

In a naive approach, the exhaustive set of up to $n(n - 1) + 1$ quasi-orders are included in the candidate set. We propose to reduce the candidate set by considering only the most salient quasi-orders. These are the ones where the number of contradictions rises significantly. Looking at the sequence β_L in the example, there are pronounced steps followed by almost level parts. The problem is now to detect the "steps" in the curve. We do so by computing the standard deviation to the cumulative sets of differences of two consecutive terms of (β_L) and denote the resulting sequence by (σ_L) , thus

$$\sigma_L = \sigma(\{\beta_{l+1} - \beta_l, 1 \leq l \leq L\}, \text{ for } 1 \leq L \leq n^2).$$

As it is evident in Figure 1, where β_L (contradictions, black curve) and σ_L (cumulative std., red curve), taking the cumulative standard deviation effectively magnifies the gap between two steps; moreover, the sequence only increases between each steps and then decreases. Therefore, the steps can easily be identified as the indices where an increase of σ_L occurs. We use the last value of each group, as the algorithm will also include all the values on the same level and those on previous levels.

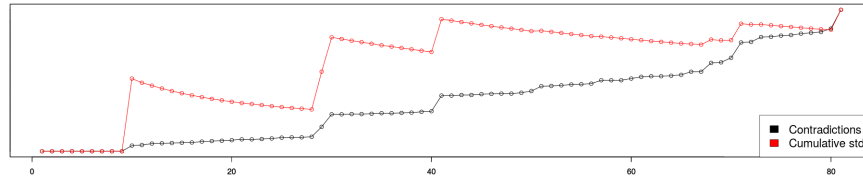


Fig. 1. Number of contradictions and the evolution of the cumulative standard deviation.

Based on this observation, we propose the following method for selecting the candidates:

Selection of Sparse Candidates

-
- Determine the sequence (β_L) of the ascending $b_{i,j}$.
 - Compute the cumulative standard deviation sequence (σ_L) .
 - If $\sigma_{L+1} > \sigma_L$, the quasi-order \sqsubseteq_L is built.
-

Using the sparse quasi-order selection algorithm, less quasi-orders are generated, consequently, for step 2, less computation time is needed. As we will show in the experiments, still the good quasi-orders are captured as long as noise levels are reasonable.

We now address the issue of transforming a relation to a transitive one. In Schrepp's algorithm the couples leading to intransitivity are simply removed. We propose to reintegrate these *rejected* pairs by the following procedure:

Reintegration

- Define $A_L = \{(i, j) \mid b_{i,j} \leq \beta_L\}$.
 - The set A_L is sorted by increasing value of corresponding $b_{i,j}$.
 - As long as A_L is not transitive, the last element of A_L is removed and stored in R_L .
 - Repeat:
 - For each element r in R_L .
 - If $A_L \cup \{r\}$ is transitive, remove r from R_L and reintegrate it into A_L .
 - If no change of R_L occurs, break.
-

We conjecture that the result of this algorithm is in fact the biggest quasi-order included in the set $\{(i, j) \mid b_{i,j} \leq \beta_L\}$. As β_L is strictly increasing, producing the same relation occurs less frequent as in Schrepp's algorithm.

3.2 Fit coefficient

As said before, the fit criterion has been the center of attention in the evolution of the method. The state-of-the-art fit coefficient (see "Original fit" in the previous section) has been proposed by Sargin and Ünlü [5] to improve the one given by Schrepp [8] with regard to quasi-orders with fewer relations. However, there are two problematic aspects to be considered:

Firstly, the formula is not symmetric : for the equivalent cases $i \not\sqsubseteq_L j$ and $j \not\sqsubseteq_L i$, the coefficient $b_{i,j}^*$ takes completely different forms. Also, the formulae for $b_{i,j}^*$ for the three cases do not allow for an intuitive interpretation.

The second point arises from the later. The fit function *diff* is not consistent, meaning that given the set of quasi-orders resulting from the first step, there are cases where even if the correct quasi-order has been computed, it is not the one that will minimize the *diff* coefficient. Consequently, the wrong quasi-order will be returned.

An example is presented in Figure 2. The red curve represents the *diff* coefficient for each quasi-order produced by the first step, while the black curve is the number of coefficient that differs from the original relation. The correct and original quasi-order is the 29th. It is indicated with a black vertical line. The one minimizing the *diff* is the 26th and is indicated with a red vertical line.

To get rid of this asymmetry, we will build another coefficient based on simple considerations. The probability $P(i \implies j \mid \mathcal{D})$ that an implication $i \implies j$ is latently included in the data is related to $b_{i,j}$ by

$$P(i \implies j \mid \mathcal{D}) = 1 - \frac{b_{i,j}}{m}.$$

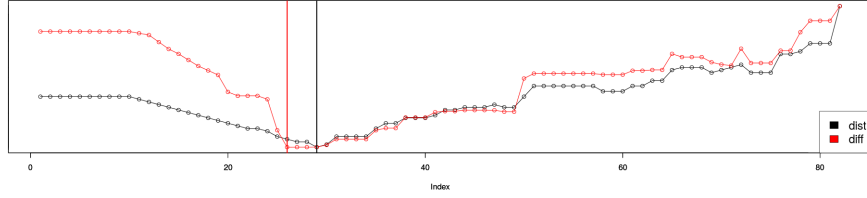


Fig. 2. The correct quasi-order (black vertical line) does not always minimize (red vertical line) the fitting coefficient.

Now the issue is determining which of the candidates fits the data best. Let us call \mathcal{M} the incidence matrix of a retained relation \sqsubseteq . From this, the set of possible patterns, $\mathcal{P} = \text{pattern}(\mathcal{M})$ is determined. If no noise was involved and \sqsubseteq was the correct quasi-order, every observation would be included in $\text{pattern}(\mathcal{M})$. And if every possible pattern had the same probability to happen, then $P(i \implies j|\mathcal{D}) = P(i \implies j|\mathcal{P})$. The closer the relation \sqsubseteq is to the data, the closer are $P(i \implies j|\mathcal{P})$ and $P(i \implies j|\mathcal{D})$. This observation motivates the coefficient *diff* which is computed as follows:

$$\begin{aligned} \text{diff}_{new}(\sqsubseteq) &= \sqrt{\sum_{i \neq j} (P(i \implies j|\mathcal{D}) - P(i \implies j|\mathcal{P}))^2} \\ &= \sqrt{\sum_{i \neq j} \left(\frac{b_{i,j}}{m} - \frac{b_{i,j}^*}{m_{\mathcal{P}}} \right)^2} \end{aligned}$$

where $m_{\mathcal{P}} = |\mathcal{P}|$ and $b_{i,j}^*$ is the number of patterns of \mathcal{P} where the implication $i \implies j$ is contradicted :

$$b_{i,j}^* := |\{p \in \mathcal{P} \mid p[i] = 1 \wedge p[j] = 0\}|$$

Corrected Fit

-
- For each quasi-order in the candidates set.
 - Build the set of possible patterns \mathcal{P} .
 - Compute the numbers $b_{i,j}^*$ of patterns contradicting the implication $i \implies j$.
 - Evaluate the fit coefficient $\text{diff}_{new}(\sqsubseteq)$.
 - Return $\text{argmin}_{\text{diff}}(\sqsubseteq)$
-

4 Experimental setup and results

We test combinations of our proposed modifications to fit coefficient and generation of candidate set against the original versions using synthetic data. The same setting is used for all three comparisons : 100 of different quasi-orders on 9 elements are built, for each 1000 data sets with 1000 observations lines are created with varying noise rate τ .

4.1 Experiment 1

This first set of experiments focuses on the modification of the first step of the procedure. For each data set, the original quasi-order is searched in the list resulting from the first step. The problem of finding it, is not an issue yet. For three different error rates $\tau \in \{0.05, 0.1, 0.15\}$, we compare the following three algorithms: *Original*, *Original with Reintegration*, *Selection with Reintegration*. The minimum distance to the correct quasi-order is then computed. If it is equal to 0, it means the correct relation is included in the candidates set. The results are reported in Table 1. These are means over the $100 \times 1\,000 = 100\,000$ loops.

$\tau = 5\%$

| | Original | Original Reintegration | Selection Reintegration |
|------------------|----------|---------------------------|----------------------------|
| Minimum | 0 | 0 | 0 |
| Mean | 0.02 | 0.02 | 0.05 |
| Maximum | 0.45 | 0.46 | 0.74 |
| Standard Dev. | 0.06 | 0.06 | 0.11 |
| Contains Correct | 98.4% | 98.7% | 96.7% |

$\tau = 10\%$

| | Original | Original Reintegration | Selection Reintegration |
|------------------|----------|---------------------------|----------------------------|
| Minimum | 0 | 0 | 0.08 |
| Mean | 0.56 | 0.54 | 1.14 |
| Maximum | 2 | 2 | 5 |
| Standard Dev. | 0.40 | 0.41 | 0.74 |
| Contains Correct | 69.8% | 70.5% | 59.2% |

$\tau = 15\%$

| | Original | Original Reintegration | Selection Reintegration |
|------------------|----------|---------------------------|----------------------------|
| Minimum | 0.11 | 0.09 | 0.40 |
| Mean | 1.38 | 1.44 | 4.21 |
| Maximum | 3.75 | 4.21 | 15.51 |
| Standard Dev. | 0.68 | 0.78 | 2.68 |
| Contains Correct | 41.4% | 42.0% | 27.0% |

Table 1. Comparison of three first step algorithms for different noise levels.

When noise increases all the algorithms behave badly. The algorithms *Original with Reintegration* and *Original* tolerate higher noise levels, even though the mixed algorithm performs better across all noise levels. It is important to point out that the mean distances stay around 1. The algorithm *Selection Reintegration* quality drops rapidly. Particularly the maximum distance goes up to

around 15, but remarkably, the standard deviation and the mean stay quite low : if *Selection* does not contain the correct quasi-order, it is still close. This shows that the selection of the sparse candidate set works in most cases.

4.2 Experiment 2

Here the interest is put on the second step, which means to compare the different fit coefficients. Again three combinations of algorithms are compared. The original *diff* coefficient is combined with the original first step algorithm to reproduce the original procedure, and with the *Selection Reintegration*. Finally the corrected *diff_{new}* is combined with the *Selection Reintegration* to show the performance of both combined. The settings of the experience is the same as previously. The results are reproduced in Table 2. The row *Found Correct* is the percentage of times the algorithm has found the correct quasi-order, and *Found Closest* is the percentage of times it has found the one in the possible set that is the closest (or equal) to the original one.

The results are clearly in favor of the improvements proposed in the article. The inductive ITA as described by Schrepp [8] and with the fit function proposed by Sargin & Ünlü [5] is able to detect the correct relation hardly 1 time over 3. This is not related to the *Original* first step, because as Table 1 shows, the correct quasi-order has a probability to be in the candidate set varying between 98% and 41% depending on the noise. The combination of the original second step and the proposed first step supports our critique of the original fit coefficient. Indeed, as there is less choice for the fit coefficient, the percentage of correct is bigger than the *Original-Original* combination. Moreover, the *Selection Reintegration* contains the correct quasi-order less frequently . On the other hand, the mix *Corrected* and *Selection Reintegration* produces the best results. It finds the closest relation in the candidates set more than 82.5% of time. This is interesting, because this algorithm for the first step often does not contain the correct relation for higher noise but it is still close to it.

4.3 Experiment 3

In this final set of experiments, we explore to what extent association rule mining can be used to mine implications. Albeit association rule mining targets n -ary antecedents and obviously will not produce transitive relations, we test whether implications are recovered as first order rules, i.e. rules of the form $i \Rightarrow j$. For this purpose we mine association rules with the R-package `arules` developed by M. Hahsler et al., and only extract first order rules. As the package does not allow for easy computation of the incidence matrix, only the number of pairs included in a relation will be considered.

The comparison is done with the proposed combination *Corrected diff - Selection Reintegration*. For the `apriori` method we select rules with confidence greater than 75% and a support greater than 1%, as this gave the best results. The noise level is set to $\tau = .5$ and $.1$. Results are presented in Table 3. The comparison is quite rough, as we do not check whether the correct relations are

$\tau = 5\%$

| <i>diff</i> | Original | Original | Corrected |
|---------------|----------|----------------------------|----------------------------|
| First Step | Original | Selection Reintegration | Selection Reintegration |
| Minimum | 0.03 | 0.01 | 0.00 |
| Mean | 1.48 | 0.71 | 0.07 |
| Maximum | 5.41 | 5.35 | 1.58 |
| Standard Dev. | 1.19 | 1.07 | 0.19 |
| Found Correct | 35.3% | 66.4% | 96.1% |
| Found Closest | 36.3% | 69.1% | 98.3% |

$\tau = 10\%$

| <i>diff</i> | Original | Original | Corrected |
|---------------|----------|----------------------------|----------------------------|
| First Step | Original | Selection Reintegration | Selection Reintegration |
| Minimum | 0.01 | 0.08 | 0.08 |
| Mean | 2.23 | 1.77 | 1.43 |
| Maximum | 7 | 10 | 12 |
| Standard Dev. | 1.38 | 1.51 | 1.39 |
| Found Correct | 19.3% | 43.7% | 56.6% |
| Found Closest | 31.2% | 76.6% | 90.7% |

$\tau = 15\%$

| <i>diff</i> | Original | Original | Corrected |
|---------------|----------|----------------------------|----------------------------|
| First Step | Original | Selection Reintegration | Selection Reintegration |
| Minimum | 0.18 | 0.40 | 0.40 |
| Mean | 3.77 | 6.21 | 5.89 |
| Maximum | 11.74 | 23.65 | 26.63 |
| Standard Dev. | 1.98 | 4.74 | 4.54 |
| Found Correct | 8.4% | 21.9% | 23.4% |
| Found Closest | 28.0% | 79.1% | 82.5% |

Table 2. Comparison of three *diff* coefficients for different noise level.

recovered, but only if their number is correct. The results show that `arules` is outperformed by our ITA algorithm. While ITA gives 98% of good answers, the `arules` only reaches 18%.

5 Conclusion and directions of future work

We proposed three modifications on Item Tree Analysis as presented by [8] and [5]. The first affects the way the transitivity is obtained. This leads to better candidates sets but worsens the computation time. To improve this, we proposed an algorithm that generates a sparse set of candidates containing only the most

$$\tau = 5\%$$

| Algorithm | Selection Reintegration Corrected | Association Rules arules |
|---------------|---|---------------------------------------|
| Minimum | 0 | 0 |
| Mean | 0.12 | 3.68 |
| Maximum | 6 | 15 |
| Standard Dev. | 0.58 | 2.72 |
| Found Correct | 93.9% | 12.3% |

Table 3. Comparison between ITA and Association Rules.

salient quasi-orders. Calculation are much faster but the results do not always behave correctly when noise levels are high. The last contribution is a new definition of the fit coefficient, $diff_{new}$. We showed that our improved fit coefficient outperforms the old definition. It also compensates the weakness of the *Selection* algorithm by finding the closest quasi-order.

Both *Selection* algorithm and $diff_{new}$ new coefficient need to be improved to tolerate high noise levels better. Effort should be put on the candidate set generation, as it conditions the results of the second. Theoretical work should also be done such as estimating the probability that the candidates set contains the correct quasi-order. This will surely reveal new directions for further improvement.

In our setting, association rule mining does not apply directly, but a deeper study should be done to reveal the ties between association rules mining and ITA to leverage ideas behind advanced algorithms for association rule mining for implication mining and ITA.

References

1. Agrawal, R., Imieliński, T., Swami, A. *Mining association rules between sets of items in large databases*. Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93. p. 207, 1993.
2. Agrawal, R., Srikant, R. *Fast Algorithms for Mining Association Rules in Large Databases*. Proceeding VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases p. 487-499, 1994.
3. Hahsler, M., Gruen, B., Hornik, K. *Computational Environment for Mining Association Rules and Frequent Item Sets*. Journal of Statistical Software 14/15 2005.
4. Hahsler, M., Buchta, C., Gruen, B., Hornik, K. *arules: Mining Association Rules and Frequent Itemsets*. R package version 1.1-3. <http://CRAN.R-project.org/package=arules> 2014.
5. Sargin, A., Ünlü, A. *Inductive item tree analysis: Corrections, improvements, and comparisons*. Mathematical Social Sciences 58, 376-392, 2009.
6. Schrepp, M. *On the empirical construction of implications between bi-valued test items*. Mathematical Social Sciences 38, 361-375, 1999.

7. Schrepp, M. *Explorative analysis of empirical data by boolean analysis of questionnaires*. Zeitschrift für Psychologie 210, 99–109, 2002.
8. Schrepp, M. *A method for the analysis of hierarchical dependencies between items of a questionnaire*. Methods of Psychological Research 19, 43–79, 2003.
9. Schrepp, M. *On the evaluation of fit measures for quasi-orders*. Mathematical Social Sciences 53, 196–208, 2007.
10. Van Leeuwe, J. *Item Tree Analysis*.. Nederlands Tijdschrift voor de Psychologie, 29, 475–484. 1974.

IR: Workshop on Information Retrieval

Named Entity Recognition from Tweets^{*}

Ayan Bandyopadhyay¹, Dwaipayan Roy¹
Mandar Mitra¹, and Sanjoy Kumar Saha²

¹ Indian Statistical Institute, India

{bandyopadhyay.ayan, dwaipayan.roy, mandar.mitra}@gmail.com,

² Jadavpur University, India

sks_ju@yahoo.co.in

Abstract. Entries in microblogging sites are very short. For example, a ‘tweet’ (a post or status update on the popular microblogging site Twitter) can contain at most 140 characters. To comply with this restriction, users frequently use abbreviations to express their thoughts, thus producing sentences that are often poorly structured or ungrammatical. As a result, it becomes a challenge to come up with methods for automatically identifying named entities (names of persons, organizations, locations etc.). In this study, we use a four-step approach to automatic named entity recognition from microposts. First, we do some preprocessing of the micropost (e.g. replace abbreviations with actual words). Then we use an off-the-shelf part-of-speech tagger to tag the nouns. Next, we use the Google Search API to retrieve sentences containing the tagged nouns. Finally, we run a standard Named Entity Recognizer (NER) on the retrieved sentences. The tagged nouns are returned along with the tags assigned by the NER. This simple approach, using readily available components, yields promising results on standard benchmark data.

1 Introduction

Microblogging emerged as a form of communication about ten years ago. Over the last decade, microblogging has evolved into an enormously popular platform for communicating via “microposts” (short text messages). According to a study, most tweets are either personal or conversational, but a large number do carry information in the form of Web links, music recommendations, and news [11]. In particular, microblogging has been demonstrated to be a particularly effective communication medium during disasters [10, 16]. Given the growing amount of information available through microblogging sites, techniques for efficiently and effectively processing this information are becoming increasingly important. One such information processing task that has attracted attention within the research community in recent times is Named Entity Recognition

^{*} Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

(NER), the task of locating and classifying names in text [6]. NER from microblogs is challenging for the following reason. Entries in microblogging sites are required to be very short. For example, a ‘tweet’ (a post or status update on the popular microblogging site Twitter) can contain at most 140 characters. To comply with this restriction, users frequently use abbreviations to express their thoughts, thus producing text that is characterised by poor spelling, grammar or structure. Existing named entity recognition (NER) tools have generally been designed for (and tested on) full-text documents. It is quite likely that these tools will not perform well on microposts [14]. In this study, we try a simple approach to NER from microposts using existing, readily available Natural Language Processing (NLP) tools. In order to circumvent the problem mentioned above regarding the use of such tools, we first identify some candidate NEs, and then look for *full-text documents* containing these candidates. For this purpose, we use the Web as a source of pages that are likely to be full-text and properly structured in nature. These pages are expected to contain longer and more grammatical passages that provide better context for the standard NLP tools. We evaluated our method using benchmark data that was created as part of the MSM2013 Challenge (<http://oak.dcs.shef.ac.uk/msm2013/>). Our approach combines simplicity with effectiveness: it compares favourably with the methods that topped the MSM2013 Challenge Task.

2 Related Work

Named Entity Recognition is a well known problem in the field of NLP. Some named entity (NE) taggers like the Stanford Tagger [7] and the Illinois Named Entity Tagger [12] have been shown to work well for properly structured sentences. However, these NE taggers are unlikely to perform satisfactorily on the incomplete, fragmented and ungrammatical sentences typically found in microposts. As a result, NE tagging for microposts has emerged as a challenging research problem. Ritter et al. [14] were among the earliest to study NER from tweets. They show that “the performance of standard NLP tools is severely degraded on tweets.” Their approach, based on Latent Dirichlet Allocation (LDA), utilises the Freebase dictionaries (<http://www.freebase.com>), and significantly outperforms the Stanford NER system. *Making Sense of Microposts* (#MSM) is a workshop series that started in 2011. It focuses on the problem of Information Extraction from microposts in general. A Concept Extraction Challenge (or contest) was organised as a part of #MSM2013. Contest participants were required to correctly identify entities belonging to one of four possible types: ‘Person’, ‘Location’ ‘Organization’ and ‘Miscellaneous’ (please see Section 3 for more details about these categories). The best challenge submission was by Habib et al. [9]. They used a hybrid approach that combines Conditional Random Fields (CRF) and Support Vector Machines (SVM) to tag named entities in microposts. The next best group [15] made use of the Wikipedia for the NER task. Dlugolinsky et al. [5] fused some well-known NER tools like GATE [4], Apache OpenNLP (<https://opennlp.apache.org/>), Illinois Named

Entity Tagger, Illinois Wikifier [13], LingPipe (<http://alias-i.com/lingpipe>) (with English News - MUC-6 model), OpenCalais (<http://www.opencalais.com/about>), Stanford Named Entity Recognizer (with 4 class caseless model), and WikiMiner (<http://wikipedia-miner.cms.waikato.ac.nz>) for named entity tagging in microposts.

3 Our Approach

As mentioned in the Introduction, our goal in this study is to recognise named entities (NEs) in microposts. Specifically, we try to identify and classify NEs belonging to the following four categories.

- **Person (PER)**: full or partial person names, e.g., Isaac Newton, Einstein.
- **Location (LOC)**: full or partial (geographical or physical) location names, including cities, provinces or states, countries, continents, e.g. Kolkata, Europe, Middle East.
- **Organization (ORG)**: full or partial organisation names, including academic, state, governmental, military and business or enterprise organizations, e.g., NASA, Reserve Bank of India.
- **Miscellaneous (MISC)**: any concept not covered by any of the categories above, but limited to one of the entity types: film/movie, entertainment award event, political event, programming language, sporting event and TV show, e.g. World Cup, Java.

| Original string Replaced by | |
|-----------------------------|-------------------|
| AFAIK | as far as I know |
| B4 | before |
| TTYL | talk to you later |
| !!!!!! | ! |
| greeeeat | great |

Table 1. Examples of changes made during preprocessing

1. **Preprocessing.** We replaced commonly used abbreviations with their expanded forms. For this step, we have used a simple lookup table consisting of 4704 commonly used abbreviations and their expansions. These were mostly collected from various Web sites (e.g., <http://osakabentures.com/2011/06/twitter-acronyms-who-knows-them/>). We also replaced strings of consecutive punctuation marks by a single punctuation mark. Finally, if a letter is repeated for emphasis, it is replaced by a single occurrence of that letter. This step is implemented via a simple lookup table of replacements. Table 1 gives some examples of changes made during preprocessing.

This preprocessing generally does not have a direct impact on NEs, but is likely to make the text more grammatical. The subsequent language processing tools that we apply (e.g., a Part of Speech tagger) are thus expected to give more accurate results. However, if a named entity coincidentally matches an abbreviation, it will also be replaced. For example, using Table 1, “B4” — the paper size — is replaced by “before”, leading to a false negative.

2. **Part of speech tagging.** We use a readily available part-of-speech (POS) tagger for microposts [8] to tag each word in a micropost with its POS. Since named entities are proper nouns, we select only the proper nouns from the tagged tweet. Neighbouring proper nouns (words that are tagged as proper nouns and have only space(s) separating them) are taken together as a group. The list of nouns / noun-groups thus extracted constitute the list of candidate NEs.
3. **Google search.** Once the candidates have been identified above, we need to eliminate the candidates that are not actually NEs, and to classify the remainder into one of the four categories listed above. This step can be viewed as a five-class classification problem, with one of the classes being “**Not an NE**”. If enough textual context were provided for each candidate, this classification task would be relatively easier. Unfortunately, because the tweets themselves are very short, they provide very little context. Since the Web can be regarded as a large natural language corpus, we turn to this obvious source in order to find longer texts containing a candidate NE. Each candidate NE is submitted as a query to the Google Search API (GSA) <http://code.google.com/apis/websearch/>. The webpages corresponding to the top 10 URLs (or fewer, if GSA returns fewer results) returned in the result list are fetched. If the original micropost is also returned among the top 10, it is neither counted nor fetched. Since Google may return slight variants of the submitted query term(s), we select only those pages that contain at least one exact match. In other words, if a page does not contain any exact match, it is discarded. If all pages are eliminated in the process, then we repeat the process once more with the next 10 results. The selected pages are likely to contain properly structured, grammatically correct sentences with the candidate NEs.
4. **NE tagging.** From the pages obtained in the above step, we extract sentences containing the candidate NEs and submit these to a standard NE tagger (the Stanford NE tagger [7]).

4 Evaluation

One standard measure used to evaluate (binary) classifiers is the F_β -score or F_β -measure. F_β is a weighted harmonic mean of the precision p and the recall r of the classifier. For the NER task, p and r are defined as follows.

Consider one of the four NE categories considered in the present study, say **PER**. Let N be the number of *actual* **PERs** present in the corpus; let n be the number of entities (words or phrases) that are tagged as **PER** by an NER

| | PER | LOC | ORG | MISC | All |
|-------------|---------------|---------------|---------------|---------------|---------------|
| OurApproach | 0.8402 | 0.3800 | 0.2836 | 0.0233 | 0.6359 |
| StanfordNER | 0.7932 | 0.3211 | 0.1395 | 0.0556 | 0.5112 |
| openNLP | 0.4968 | 0.2235 | 0.0483 | 0.0000 | 0.3889 |
| LabelledLDA | 0.7884 | 0.4227 | 0.4364 | 0.0954 | 0.5881 |
| 14 - 1 | 0.9230 | 0.6730 | 0.8770 | 0.6220 | 0.7740 |
| 21 - 3 | 0.8760 | 0.6030 | 0.8640 | 0.7140 | 0.7640 |
| 15 - 3 | 0.8790 | 0.6860 | 0.8440 | 0.5250 | 0.7340 |

Table 2. Overall and category-wise precision results

system; and let m be the number of actual **PERs** that are *correctly* identified by the NER system. Then p , r and F_β are given by:

$$p = \frac{m}{n} \quad r = \frac{m}{N} \quad F_\beta = \frac{(1 + \beta^2) * p * r}{(\beta^2 * p) + r}$$

For this work, we adopt the common policy of setting β to 1 to allow precision and recall to be weighted equally. With $\beta = 1$, the F_β -measure reduces to the conventional harmonic mean of p and r , and is referred to as the F_1 -measure. The F -measure is computed separately for each of the four NE categories mentioned in Section 3 and then averaged across the four categories to obtain a single overall measure of performance.

5 Results

For evaluation, we used the data set provided by “Making Sense of Microposts (#MSM2013)” [2]. The data consists of 1450 tweets contained in a single file, with one tweet per line. Each tweet has a unique tweet-id and the tweet text.

Tables 2–4 compare our approach with several readily available NER tools applied directly on the tweet text: openNLP tool, Stanford NER [7], and Labeled LDA method [14]. Since we used the MSM2013 data, we also compare our method with the three best submissions to the MSM2013 challenge (these are identified by their submission numbers in the tables). More details about the MSM2013 results can be found in the MSM2013 overview paper [3].

| | PER | LOC | ORG | MISC | All |
|-------------|---------------|---------------|---------------|---------------|---------------|
| OurApproach | 0.6922 | 0.5700 | 0.3305 | 0.0211 | 0.5884 |
| StanfordNER | 0.7269 | 0.6100 | 0.3263 | 0.0632 | 0.6180 |
| openNLP | 0.2794 | 0.1900 | 0.0424 | 0.0000 | 0.2206 |
| LabelledLDA | 0.7358 | 0.4100 | 0.1017 | 0.3053 | 0.5923 |
| 14 - 1 | 0.9080 | 0.6110 | 0.6200 | 0.2770 | 0.6040 |
| 21 - 3 | 0.9380 | 0.6140 | 0.6130 | 0.2870 | 0.6130 |
| 15 - 3 | 0.9520 | 0.4850 | 0.7390 | 0.2690 | 0.6110 |

Table 3. Overall and category-wise recall results

| | PER | LOC | ORG | MISC | All |
|-------------|---------------|---------------|---------------|---------------|---------------|
| OurApproach | 0.7583 | 0.4542 | 0.3041 | 0.0220 | 0.6123 |
| StanfordNER | 0.7586 | 0.4207 | 0.1954 | 0.0591 | 0.5474 |
| openNLP | 0.3576 | 0.2054 | 0.0451 | 0.0000 | 0.2815 |
| LabelledLDA | 0.7612 | 0.4162 | 0.1649 | 0.1454 | 0.5902 |
| 14 - 1 | 0.9200 | 0.6400 | 0.7380 | 0.3830 | 0.6700 |
| 21 - 3 | 0.9100 | 0.6090 | 0.7210 | 0.4100 | 0.6620 |
| 15 - 3 | 0.9180 | 0.5680 | 0.7900 | 0.3560 | 0.6580 |

Table 4. Overall and category-wise F_1 results

In general, we find that our method fails to identify NEs in the MISC category. Though the named entities are recognised, they are misclassified in most cases. One reason for misclassification is the occurrence of named entities like Annie Hall (tweet id 2904). Since this is the name of a fictional character, it is classified as PER. However, the tweet is about the movie by this name; thus, the entity actually belongs to the MISC category. This is one of the reasons affecting the precision of our method.

However, it is encouraging to note that the overall results obtained by our method are not statistically significantly different from the best results reported at MSM2013. We used the Welch Two Sample t-test [17] to determine the statistical significance of the differences between our approach and the top three submissions at MSM2013 (run IDs 14-1, 21-3 and 15-3). Table 5 shows the p-values for the three tests.

| | 14 - 1 | 21 - 3 | 15 - 3 |
|--------------|--------|--------|--------|
| Our Approach | 0.1878 | 0.1916 | 0.2171 |

Table 5. p -values for Welch Two Sample t-test

Discussion Table 6 analyses the nature of false negatives for our method. Our method is based on the following assumption: while the text surrounding an NE may be of poor-quality, users are careful / accurate when mentioning names. This assumption turns out not be completely correct. For example, one tweet mentions ‘britnay spers’ (instead of ‘Britney Spears’). Similarly, tweet ID 4261

| | |
|---|------|
| Total # of NEs in dataset | 1555 |
| (Step 2) # of NEs not tagged as candidate by POS tagger | 396 |
| (Step 3) # of candidates for which no results found | 5 |
| (Step 4) # of candidates misclassified | 239 |

Table 6. Analysis of false negatives

mentions ‘Annie Lenox’ (presumably the Scottish singer-songwriter) whose name is actually spelt ‘Annie Lennox’.

6 Conclusion

The key idea in our approach is to use the Web as a source of documents that are generally longer and better structured than tweets. This enables us to use standard NLP tools without having to redesign or retrain them. Since NER-tagged training data from the micropost domain is a scarce resource, this is an advantage. Significance tests show that our results are comparable to the state of the art.

As mentioned in the preceding section, however, our approach is based on the assumption that NEs in tweets are correctly written. Our immediate goal in future work would be to handle spelling errors / variations. One obvious way to do this would be to leverage the “Did you mean” feature provided by Google (note that this feature is *not* available via GSA). It may also be possible to handle spelling errors using a dictionary-based spelling correction algorithm that uses the Google n -gram dataset [1] as a lexicon. We would also like to explore the possibility of using our method to create labelled data that may in turn be used to train a more direct approach. This would eventually enable us to avoid the use of the GSA as a black box.

References

1. <http://googleresearch.blogspot.in/2006/08/all-our-n-gram-are-belong-to-you.html>
2. <http://oak.dcs.shef.ac.uk/msm2013/>
3. Basave, A.E.C., Rowe, M., Stankovic, M., Dadzie, A.S. (eds.): Proc. Concept Extraction Challenge at the 3rd Workshop on Making Sense of Microposts (#MSM2013): Big things come in small packages. CEUR Workshop Proceedings (May 2013), <http://ceur-ws.org/Vol-1019>
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proc. 40th Ann. Meeting of the ACL (July 2002), <http://gate.ac.uk/sale/acl02/acl-main.pdf>
5. Dlugolinsky, S., Krammer, P., Ciglan, M., Laclavik, M.: MSM2013 IE Challenge: Annotowatch. vol. 1019, pp. 21–26 (2013), In [3].
6. Downey, D., Broadhead, M., Etzioni, O.: Locating complex named entities in web text. In: Proc. 20th IJCAI. pp. 2733–2739. IJCAI’07 (2007), <http://dl.acm.org/citation.cfm?id=1625275.1625715>
7. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proc. 43rd Ann. Meeting of the ACL. pp. 363–370. ACL (2005), <http://dx.doi.org/10.3115/1219840.1219885>
8. Gimpel, K., Schneider, N., O’Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: Part-of-speech tagging for

- twitter: annotation, features, and experiments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2. pp. 42–47. HLT '11, Association for Computational Linguistics (2011), <http://dl.acm.org/citation.cfm?id=2002736.2002747>
9. Habib, M.B., van Keulen, M., Zhu, Z.: Concept extraction challenge: University of twente at #msm2013. vol. 1019, pp. 17–20 (2013), In [3].
 10. Jennex, M.E., de Walle, B.V. (eds.): International Journal of Information Systems for Crisis Response and Management (IJISCRAM). IGI Global (Est 2009)
 11. Kelly, R.: Twitter study – August 2009 (2009), <http://www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf>
 12. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning. pp. 147–155. CoNLL '09, Association for Computational Linguistics (2009), <http://dl.acm.org/citation.cfm?id=1596374.1596399>
 13. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 1375–1384. HLT '11, Association for Computational Linguistics (2011), <http://dl.acm.org/citation.cfm?id=2002472.2002642>
 14. Ritter, A., Clark, S., Mausam, Etzioni, O.: Named entity recognition in tweets: An experimental study. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1524–1534. EMNLP '11, Association for Computational Linguistics (2011), <http://dl.acm.org/citation.cfm?id=2145432.2145595>
 15. Sachidanandan, S., Sambaturu, P., Karlapalem, K.: NERTUW: Named entity recognition on tweets using Wikipedia. vol. 1019, pp. 67–70 (2013), In [3].
 16. Vieweg, S., Hughes, A.L., Starbird, K., Palen, L.: Microblogging during two natural hazards events: What twitter may contribute to situational awareness. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1079–1088. CHI '10, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1753326.1753486>
 17. Welch, B.L.: The generalization of student's problem when several different population variances are involved. *Biometrika* 34(1-2), 28–35 (1947), <http://biomet.oxfordjournals.org/content/34/1-2/28.short>

On the Stability of Signature-based Distance Functions for Content-based Image Retrieval

Christian Beecks^{*} Steffen Kirchhoff[°] Thomas Seidl^{*}

^{*}RWTH Aachen University
^{*}{beecks,seidl}@cs.rwth-aachen.de

[°]Harvard University
[°]kirchhoff@fas.harvard.edu

Retrieving similar images from large image databases is a challenging task for today's content-based retrieval systems. Aiming at high retrieval performance, these systems frequently capture the user's notion of similarity through expressive image models and adaptive similarity measures. On the query side, image models can significantly differ in quality compared to those stored on the database side. Thus, similarity measures have to be robust against these individual quality changes in order to maintain high retrieval performance.

In this paper, we investigate the robustness of the family of signature-based distance functions in the context of content-based image retrieval. To this end, we investigate the generic concept of average precision stability, which measures the stability of a similarity measure with respect to changes in quality between the query and database side. In addition to the mathematical definition of average precision stability, we include a performance evaluation of the major signature-based distance functions focusing on their stability with respect to querying image databases by examples of varying quality. Our performance evaluation on recent benchmark image databases reveals that the highest retrieval performance does not necessarily coincide with the highest stability.

This is a resubmission of previously published papers by Beecks et al. [1, 2].

Keywords: Content-based image retrieval, Distance-based similarity measure, Evaluation measure, Average precision stability

References

1. C. Beecks, S. Kirchhoff, and T. Seidl. On stability of signature-based similarity measures for content-based image retrieval. *Multimedia Tools Appl.*, 71(1):349–362, 2014.
2. C. Beecks and T. Seidl. On stability of adaptive similarity measures for content-based image retrieval. In *Proceedings of the International Conference on Multimedia Modeling*, pages 346–357, 2012.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

Finding the Right Experts and Learning to Rank Them by Relevance: Evaluation of a Semi-automatically Generated Ranking Function

Felix Beierle^{1,2}, Felix Engel², Matthias Hemmje²

¹ Service-centric Networking, TU Berlin; Telekom Innovation Laboratories
beierle@tu-berlin.de

² Multimedia and Internet Applications, University of Hagen
{felix.engel,matthias.hemmje}@fernuni-hagen.de

Abstract. A framework for expert searching developed at the University of Hagen supports the use of contextual factors motivated in the field of expertise seeking. A user can query the system with skills which the person to be found should be an expert in. The result is a list of users from the searched knowledge base, ranked by relevance for the given query. In this paper, we address the semi-automatic generation of training data for the learning-to-rank library that does the relevance ranking. We focus on evaluating the quality of the ranking function.

1 Introduction

In many business situations, it is essential to have the right experts at hand: For instance, for an equipment rental company, failure of equipment is expensive because either downtime has to be financially compensated for or replacements have to be provided to the customer. In order to do maintenance work on a machine, an engineer needs expertise for a certain machine or device.

Finding such an expert is not a trivial task. Besides his knowledge about the machine/device, the field of *expertise seeking* suggests that contextual factors should be considered, most importantly the familiarity between searcher and expert (e.g. [7]). Through *feature vectors*, potential experts can be represented through numerical values. *Learning to rank* (LTR) can be used to rank those feature vectors by relevance. To learn a ranking function, training data has to be provided. As the generation of training data is expensive, in [6], it is suggested to use a rule-based approach to semi-automatically generate such data. In [4], we briefly reported about the development and implementation of a hierarchical system of rules for the generation of training data. After presenting an overview of this system, the focus of this paper is to evaluate the implemented approach with respect to the quality of the ranking function.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

This paper uses results from [3] and is organized as follows: First, we refer to related work (Sec. 2) and give details about the software framework (Sec. 3) that was developed within the SMART VORTEX project (<http://www.smart-vortex.eu>). We sketch the rule system (Sec. 4) introduced in [4] and report about an evaluation (Sec. 5) of the implementation. In Sec. 6 we conclude and point out further work.

2 Related Work

In the following, we will refer to related work in the field of expertise seeking and we will present our concept of *expert seeking parameters*. When searching for an expert, different expert seeking parameters have to be taken into account. Information about the expert is needed to assess his/her relevance; while this information might be available in a company knowledge base, it may be unknown to the searcher [10]. *Quality*-related relevance factors are about the formal qualifications of the potential experts [11]. *Topic* refers to direct links between a user and the asked skills (a skill being, e.g., the knowledge about a specific machine). We use the term *approach* bundling aspects regarding the expert's perspective on the asked field of expertise (e.g., engineer vs. construction worker). *Up-To-Dateness* refers to temporal information like the last time an asked skill was used for a project. *Experience* can include factors like the amount of time someone has been working for the company, years of work experience, number of projects, or the number of connections someone has in a semantically annotated company knowledge base, etc. Studies in expertise seeking come to the conclusion that the familiarity between searcher and expert is the most important relevance factor (with about 10-20%) in the *accessibility* category [11] [7]. We refer to space- and time-constraints with the parameter *proximity* and to other relational aspects with *closeness*. In contrast to the quality-related factors, all of the accessibility-related factors depend not only on the expert, but also on the user that is performing the search.

Regarding the generation of training data for learning to rank processes, so far, besides manual specification, crowd sourcing [5] and log analysis have been suggested [8]. A less cost intensive approach is to use a rule-based system, as motivated in [6], which we will elaborate in this paper.

3 OWIM SemSearch Framework

At the University of Hagen, the OWIM SemSearch Framework (Open Workbench for Information Management) was developed within the SMART VORTEX project; its architecture is shown in Fig. 1. The general approach is as follows: For every person that is modeled within a semantically annotated knowledge base a *feature vector* (or *feature value vector*) is constructed, consisting of several *features* (e.g. the number of finished related projects). Each feature is represented through a *feature value*. Each expert seeking parameter consists of a set of *relevance aspects*. One relevance aspect can consist of a single feature (e.g. age). As motivated in [6], there can be dependencies between features,

and therefore, one relevance aspect can also consist of more than one feature (e.g. number of projects *and* years of work experience). The framework uses a pairwise LTR approach (different LTR libraries can be used, e.g., RankLib, <http://sourceforge.net/p/lemur/wiki/RankLib>): Using a system of rules that express relevance patterns, labeled training data is generated. This data consists of pairs of feature vectors, along with the information which of the two is to be considered more relevant. Using the training data, LTR is applied to learn a ranking function by estimating the weight of every component of the feature vectors. The learned ranking model can be used to classify feature vectors from future searches.

Before the software can be used, a *domain expert* has to configure the system. Similar to the idea of an *application context* in [2] we propose that a domain expert with knowledge about the ontology configures the search. In the *feature vector configuration*, he defines the features and how the feature values are calculated. He also defines rules for all relevance aspects, for example: If a person *A* has completed more projects related to queried skills, and has more years of work experience than person *B*,

then *A* is more relevant with respect to the relevance aspect 'work experience.' Once the configuration, e.g. for *skill*, of the vector and the rule system is completed, a user may enter a query for a particular set of skills and gets a list of the company's employees sorted by relevance with respect to the queried skills; this list is sorted by the previously learned ranking function. Details about the implemented algorithms for retrieving data from the ontology, as well as how they are used in the configuration, are given in [3].

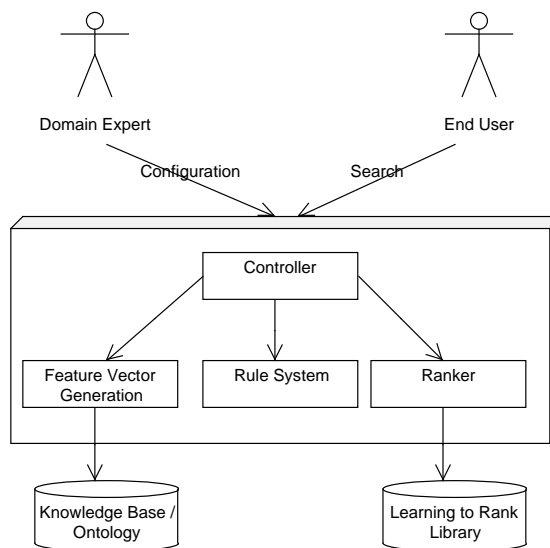


Fig. 1. Architecture

4 Rule System for Relevance Labeling in Pairwise LTR

Single-Feature Relevance Aspect Comparison A relevance aspect can consist of one single feature (e.g. number of publications). For comparing \mathbf{a} and \mathbf{b} with respect to such a single-feature relevance aspect, the corresponding feature values $a_i, b_i \in \mathbb{R}_{\geq 0}$ are compared (the index i indicating the position in the feature vector). Besides providing the two basic comparison operators, greater ($>$) and lesser ($<$), another requirement could be to consider only values that are higher than a certain *threshold* $t \in \mathbb{R}_{\geq 0}$. For instance, looking for an

expert with many publications, the comparison operator is $>$. A threshold can be used to disregard employees that have not published at least a certain number of papers. If a company wants to support their younger employees, the relevance aspect 'age' could be used with the comparison operator $<$, using a threshold to disregard employees that are too young.

The comparison yields either T (if a_i is considered more relevant than b_i), F (if b_i is more relevant than a_i), or 0 (neither of a_i, b_i is more relevant than the other one). Thus, the comparison function $c_\diamond(a_i, b_i, t)$ for single-feature relevance aspects with \diamond being $>$ or $<$ and with threshold t has the target set $\{T, F, 0\}$ and is defined as follows:

$$c_{>}(a_i, b_i, t) = \begin{cases} T & a_i \geq t \wedge a_i > b_i \\ F & b_i \geq t \wedge a_i < b_i \\ 0 & \text{otherwise} \end{cases} \quad c_{<}(a_i, b_i, t) = \begin{cases} T & a_i \geq t \wedge a_i < b_i \\ F & b_i \geq t \wedge a_i > b_i \\ 0 & \text{otherwise} \end{cases}$$

Note that both single feature value comparison functions are complementary in the first two arguments, i.e., $c_\diamond(a_i, b_i, t) = T \Leftrightarrow c_\diamond(b_i, a_i, t) = F$, and $c_\diamond(a_i, b_i, t) = 0 \Leftrightarrow c_\diamond(b_i, a_i, t) = 0$, for $\diamond \in \{<, >\}$ and for all non-negative values a_i, b_i, t .

Multi-Feature Relevance Aspects Comparison Following the example of 'work experience,' a person has to have both more years of work experience and a higher number of finished projects to be considered more relevant. In order to compare such multi-feature relevance aspects, several single feature value comparisons have to be taken into account. For this we introduce a three-value logic for the conjunction of two values in $\{T, F, 0\}$: $T \wedge T = T$ and $F \wedge F = F$ and all other conjunctions are evaluated to 0 . The idea of this conjunction is that one feature vector has to be more relevant for all single comparisons of the multi-feature comparison to be more relevant with respect to the given multi-feature relevance aspect.

Comparison with Respect to Sets of Relevance Aspects For an expert seeking parameter E , let x be the number of comparisons of relevance aspects in E that determine \mathbf{a} more relevant and let y be the number of comparisons that determine \mathbf{b} more relevant. If $x > y$, \mathbf{a} is considered more relevant, if $y > x$, \mathbf{b} is considered more relevant, otherwise they are regarded to be equally relevant with respect to that expert seeking parameter.

Feature Vector Comparison For the comparison of two feature vectors, there is one further level: the aggregation of the results of the comparison with respect to a set of expert seeking parameters. A value between 0 and 1 is assigned to each expert seeking parameter, signifying the percentage of relevance the parameter should take up. The percentages considering a feature vector more relevant are accumulated, and the person with the feature vector rated at a higher cumulated percentage value is labeled as the more relevant person for the given search.

5 Evaluation

The goal of the evaluation is to test the framework with a company knowledge base to determine the factors that play a role regarding the quality of the rank-

ing function, and to check how well the ranking function ranks after learning. Because there is no manually specified ground truth available, the ground truth generated by the rule system will serve as evaluation data. The evaluation is done with the given company knowledge base which has 48 users and 194 skills.

The evaluation is done using k-fold cross validation with Kendall's Tau-b [1]. Kendall's Tau-b is used for the comparison of rankings returned by the rule system and by the learned ranking function, and yields a number between -1 and 1 . The value 1 indicates completely concordant rankings, 0 indicates no specific relationship between the two rankings, and -1 indicates completely discordant rankings. The k-fold cross validation yields the *evaluation value*. The higher this value, the better the relevance pattern expressed in the training data could be generalized by the learning to rank library.

First, we will give detailed information about the design of the evaluation and about the factors that have to be considered when evaluating the framework (Section 5.1). Then, we will present and analyze the results of the evaluation in Section 5.2.

5.1 Design

It is expected that the amount of training pairs has the single most significant impact on the evaluation value. In order to show causality and not just correlation, we will take into consideration all factors that could have an impact on the evaluation value. We can distinguish two phases in which variable factors that could impact the evaluation value have to be considered. Firstly, there is the generation of queries with which to generate ground truth. Here, we will introduce four factors ((Q1)-(Q4)) to consider. Secondly, for the evaluation itself, we will introduce three factors to consider ((E1)-(E3)).

Creating Queries For creating queries, the first three factors to consider are:

- (Q1) the number of skills in one query
- (Q2) the skill(s) in a query
- (Q3) the logged-in user who queries the system

To be able to control the amount of created data, we need to limit the amount of users for which feature vectors will be constructed. If there are n users, there are $1/2 * n * (n - 1)$ possible pairs of users. When generating ground truth, one user will be logged in and will thus not be in the resulting list of experts. Further considering the number of queries, the following equation (1) shows, how the number of users for which feature vectors are constructed is related to the number of queries and the number of possible training pairs³:

$$\left(\begin{matrix} \text{number of} \\ \text{queries} \end{matrix}\right) * \frac{1}{2} * \left(\left(\begin{matrix} \text{number of} \\ \text{users} \end{matrix}\right) - 1\right) * \left(\left(\begin{matrix} \text{number of} \\ \text{users} \end{matrix}\right) - 2\right) = \left(\begin{matrix} \text{number of} \\ \text{training pairs} \end{matrix}\right) \quad (1)$$

Since the used knowledge base contains 48 users, a maximum of $1 * 1/2 * (48 - 1) * (48 - 2) = 1081$ training pairs can be calculated with one query. To get

³ Both those factors - number of queries and number of training pairs - will be considered later, after the queries are created.

lesser amounts of training data, or to be able to use multiple queries with a fixed amount of pairs to calculate, we will need to limit the *number of users* to create feature vectors for. This gives us an additional item to consider in the creation of queries:

(Q4) the users to create feature vectors for

Macdonald et al. report about *sample selection bias* when choosing samples that are used as training data [9]. Choosing specific data as samples, e.g., data known to be relevant, might influence the learned ranking function to prefer the chosen sample data in subsequent searches. To avoid the sample selection bias, we will choose the factors from the four points (Q1), (Q2), (Q3), and (Q4) completely randomly. The idea of the expert seeking framework is that it should work on any company ontology and regardless of who uses the system. The number of skills per query (Q1) is chosen randomly from 1 to 5. This seems like a reasonable range of skills per query a user will generate.

The queries themselves (the skills used in them) (Q2) and the logged-in user (Q3) are chosen randomly for the evaluation as well. When randomly choosing the logged-in user for the calculation of ground truth, the picked user will be remembered by the framework to pick the same one when comparing the results. In dependence on given amounts of queries and of training pairs, the needed amount of users to calculate feature vectors for is calculated with equation (1) above. After calculating the needed amount of users to calculate feature vectors for, the users are chosen randomly (Q4).

Evaluating For the second phase, the actual evaluation after the creation of ground truth, the first two factors to consider are:

- (E1) number of queries
- (E2) number of training pairs

To measure a possible impact on the quality of the ranking function, these two factors have to be analyzed. All generated pairs, the ground truth, are split into k groups. Depending on the number of groups, the number of training pairs varies. The number of training pairs is not only dependent on the number of evaluation groups, but also on the number of pairings for which the rule system considers neither feature vector to be more relevant - those pairings cannot be used for training. Both items, the number of queries and the number of training pairs, will be considered in the analysis of the results of the evaluation.

In [9], Macdonald et al. show that too little training data will yield bad results, and that too much training data will take longer to process. Thus, the goal must be to find the lowest number of training pairs that yields good results for the given knowledge base.

Additionally, there is another factor to be considered regarding the configuration of the tests:

- (E3) the length of the feature vector

According to the concept presented in Section 2, a minimum of six features will be required to be able to represent every expert seeking parameter. We evaluate three different feature vector configurations, with 6, 9, and 12 features.

The framework is evaluated with two different LTR libraries, RankLib (cf. Section 3) and SVMRank (http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html). The two LTR libraries only show negligible differences.

5.2 Results

Regarding the number of queries used to generate the ground truth, no correlation can be found between that number and the evaluation value. A higher or lower number of queries does not contribute to a better or worse ranking function.

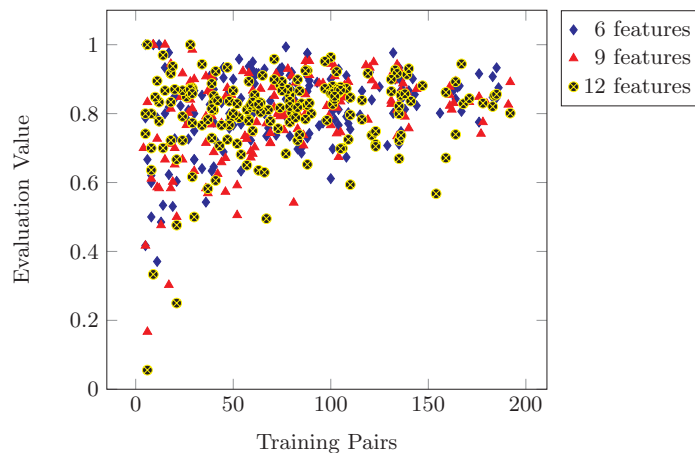


Fig. 2. Number of Training Pairs and Evaluation Value (using SVMRank)

Figure 2 shows the dependencies between the number of training pairs and the evaluation value. Each point in the scatter plot indicates the result of an evaluation. For all given feature vector lengths, roughly the same distribution of points is shown. On the left hand side, for few training pairs, the evaluation value is basically anywhere between 0 and 1. The explanation for this is that with few training pairs, the relevance pattern cannot be properly expressed and overfitting occurs: The learned ranking function is too specific for the training data and does not perform well when classifying unseen data.⁴ The more training data is available, the narrower the range of the distribution of points. With enough training data, the relevance pattern properly expressed in the training data can be generalized by the learning to rank library. Generally speaking, for the given knowledge base, at around 150 to 200 training pairs, the evaluation value is always around 0.8.

The length of the feature vector seems to barely have an impact on the evaluation value. For the vector with 12 features, the range of the evaluation value is larger compared to the other feature vector lengths. An explanation for this observation is that the more features are used, the higher the probability for overfitting becomes.

⁴ Note that with very few training pairs, occasionally the evaluation value was below 0 while the figure shows only positive results.

6 Conclusions and Future Work

In this paper, we addressed the problem of generating training data for LTR processes in an expert seeking application. The implemented framework can be used to semi-automatically generate training data through a hierarchical system of rules. The evaluation shows that the single most important factor for the quality of the ranking function is the number of training pairs that are used to learn the ranking function. Future work that can be addressed is the implementation of online-learning, where the ranking function is updated through user feedback from previous searches. Our future work also includes using our software with other company knowledge bases, and testing and evaluating the framework with respect to standard information retrieval evaluation methods.

Acknowledgments. This work has been partly supported by the FP7 EU project SMART VORTEX.

References

1. Agresti, A.: Analysis of ordinal categorical data. Wiley series in probability and mathematical statistics: Applied probability and statistics, Wiley (1984)
2. Albertoni, R., De Martino, M.: Semantic similarity of ontology instances tailored on the application context. In: OTM Conferences. pp. 1020–1038. LNCS Vol. 4275, Springer (2006)
3. Beierle, F.: Using a System of Rules to Generate Training Data for Learning-to-Rank Processes in an Expert Seeking Application. Master’s thesis, University of Hagen (2014)
4. Beierle, F., Engel, F., Hemmje, M.: Generation of training data for learning-to-rank processes in an expert seeking application. In: Informatiktage 2014 - Fachwissenschaftlicher Informatik-Kongress. Lecture Notes in Informatics (LNI), vol. S-13, pp. 97–100. Köllen Druck+Verlag (2014)
5. Dali, L., Fortuna, B., Tran, T., Mladenović, D.: Query-independent learning to rank for RDF entity search. The Semantic Web pp. 484–498 (2012)
6. Engel, F., Juchmes, M., Hemmje, M.: Expert search in semantic annotated enterprise data: integrating query- dependent and independent relevance factors. In: LWA 2013 - Lernen, Wissen & Adaptivität. Workshop Proceedings. pp. 41–44. Bamberg (2013)
7. Hofmann, K., Balog, K., Bogers, T., Rijke, M.d.: Contextual factors for finding similar experts. *Journal of the American Society for Information Science & Technology* 61(5), 994–1014 (2010)
8. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 133–142. KDD ’02, ACM, New York, NY, USA (2002)
9. Macdonald, C., Santos, R.L.T., Ounis, I.: The whens and hows of learning to rank for web search. *Inf. Retr.* 16(5), 584–628 (2013)
10. Nevo, D., Benbasat, I., Wand, Y.: The knowledge demands of expertise seekers in two different contexts: Knowledge allocation versus knowledge retrieval. *Decis. Support Syst.* 53(3), 482–489 (2012)
11. Woudstra, L., van den Hooff, B., Schouten, A.P.: Dimensions of quality and accessibility: Selection of human information sources from a social capital perspective. *Information Processing & Management* 48(4), 618–630 (2012)

A Comparison of Search Engine Technologies for a Clinical Data Warehouse ^{*}

Georg Dietrich¹, Georg Fette^{1,2}, and Frank Puppe¹

¹ University of Würzburg, Department of Computer Science
{dietrich, fette, puppe}@informatik.uni-wuerzburg.de

² DZHI (Deutsches Zentrum für Herzinsuffizienz)

Abstract. A clinical data warehouse (DW) can be used to recruit patients for clinical studies or statistical analysis. For improved user experience, it is crucial that the search engine technology of the DW answers user queries quickly. In this paper, we investigate the performance of the two most popular technologies for regarding structured and unstructured data query answering: a database and a search engine. Our empiric results show that search engines have advantages for complex queries.

1 Introduction

A clinical data warehouse makes data available for a variety purposes, e.g., information retrieval and statistical evaluations. The data consists of basic data, symptoms, diagnoses and therapies. Use-cases are the retrieval of patients for clinical studies, which have several inclusion and exclusion criteria, statistical analysis of frequencies of patient groups, the search for risk factors for specific diseases and statistical quality checks. For efficient usage, quick query answering is crucial. In this direction, we compare the performance of two alternative techniques in a real world application featuring a clinical data warehouse DWH utilized at the University of Würzburg.

Currently (June 2014) the data warehouse of the University Hospital of Würzburg consists of basic data, diagnoses, laboratory findings and echocardiography data for the years 2012 and 2013. There are about 700 000 cases with more than 25 million facts available. To protect the privacy, all data has been pseudonymised. The mapping of the pseudo-ids to the patient-ids is managed by a third party, which can approve applications to e.g. recruit patients for studies with the data warehouse.

In order to work with the data warehouse, it must be able to answer queries quickly. The response time of every query should be less than one second. Therefore it is necessary to store the data in an efficient way. Furthermore intuitive usability is important. The users should be able to work the tool without a big training period.

There are several ways to design such an information retrieval system. A basic approach is to use a database management system. As a first step, a schema has to be designed. This is a non-trivial task, because the knowledge base consists of about 55 000

^{*} Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

concepts like the laboratory finding of natrium, the age of a patient or the diagnosis heart attack. After that, indices for the tables need to be created to speed up the system. Finally, the algorithms, which automatically create queries for the database, have to be implemented.

Another approach is to use a Resource Description Framework (RDF). Here, the schema has to be specified first, too. The 25 million facts are then stored as RDF triples, such as patient X has laboratory finding of natrium of 140 mmol/l. In this example, patient X is the subject, a laboratory finding of natrium is the predicate and 140 mmol/l is the object. The RDF data model can be queried with the *SPARQL Protocol And RDF Query Language* [8]. A common framework for storing and querying RDF data is Sesame [4]. The user queries contain usually about ten or more parameters. This is quite a lot and the RDF storage did not scale for our challenges. (See section 5)

The third approach is to use a search engine. We used Apache Solr [2], which needs a schema for the documents and their fields. This is similar to the database schema, being flexible to new fields and changes.

Our data warehouse query should provide the following features: (i) An intuitive usable graphical user interface to create easily queries, whose result is displayed in a clear way, (ii) a search for hierarchical structures like a diagnosis-tree, (iii) span and segment queries to search medical concepts, which consist of several words, (iv) the system is able to use synonyms and abbreviations for medical query terms, and (v) very fast response time for complex user queries.

2 Background

The clinical data consists of four data types: (i) Numeric values: Most Laboratory Findings are floats with a few decimal places like haemoglobin = 16.58 g/d, (ii) Boolean values: Diagnoses are represented as boolean values. When a disease is diagnosed it is stored like hypertension = true, (iii) Text values: Several medical reports of findings exist as texts like discharge letters or electrocardiogram reports, and (iv) Enumerations: Many Attributes have a few values like sex (female, male) or type of treatment (residential, semi-residential, ambulant)

2.1 Data Schema

In our first approach all available facts were stored in a relational database. A simplified model of the data schema with two basic tables is shown in Figure 1.

All attributes, which a patient can have, are stored in the *Catalog*-table, which also represents hierarchical relations. Examples for attributes are natrium (lab data), sex (basic data) or I50.22 Chronic systolic heart failure (diagnostic data). The values for the attributes are stored in the *Info*-table: The CaseID represents one "case" of a patient including all data for that patient in that time period. The CaseID, AttrID and the Value form a triple structure: For one case and one attribute exists one value, e.g. the patient of a case has blood pressure of 125. It is possible that one attribute has several values, like multiple measurements of one attribute at different time stamps. This is mapped with several rows in the table.

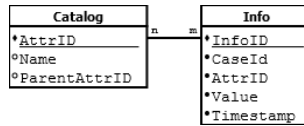


Fig. 1: Simplified relational model of the database with a triple structure in the Info-table: CaseID, AttrID, Value and a timestamp

An alternative schema with one big table and a column for every attribute was tested, but discarded, because only ca. 1 000 columns per table were allowed (just for the diagnoses, we needed more than 16 000 columns).

2.2 Hierarchical search

A special function is the hierarchical search. Our terminology is hierarchically ordered, like the ICD-10 coded diagnoses [1]. The international classification of diseases is a catalog for epidemiology, health management and clinical purposes.

If a specific disease like *I20.0 Instabile Angina pectoris* is diagnosed, then the "parent"-disease (here: *I20. Angina pectoris*) exists, too. Usually, a very specific diagnosis like *I20.0* is documented, which has a high depth in the catalog. But the data warehouse user may search for a more general diagnosis like *I20*. To meet this requirement, new facts were generated by preprocessing, i.e. setting all parent diagnoses of a diagnosis "true", thus propagating the diagnosis up in the tree.

2.3 Graphical user interface

The graphical user interface (GUI) consists mainly of three views. In the catalog view (Figure 2a) all attributes are hierarchically sorted. After every attribute name the total number of occurrences in the data warehouse is shown. Attributes can be dragged from the catalog-view (Figure 2a) and dropped in the query view (Figure 2c). Operators and constraints can be applied to these attributes in the query view, e.g. numeric range selection. If one attribute has more than one value in one case, it is possible to specify which one should be selected (first, last, min, max). Moreover the boolean operators "AND", "OR" and "NOT" are available for combinations. In the result view the query matching cases are displayed tabularly (Figure 2b).

The GUI has not been systematically evaluated or compared with other tools, but it was tested during the development stage by a group of users, who were very satisfied and gave a positive feedback.

3 Evaluation: A speed-test between Solr and a DBMS

A database server and a search platform were tested as storage engine for the data warehouse application.

| |
|---|
| IX : Krankheiten des Kreislaufsystems (71355) |
| + I00-I02 : Akutes rheumatisches Fieber (42) |
| + I05-I09 : Chronische rheumatische Herzkrankheiten |
| I10-I15 : Hypertonie (46983) |
| I10 : Essentielle (primäre) Hypertonie (46113) |
| I10.0 : Benigne essentielle Hypertonie (45 |
| I10.0 : Benigne essentielle Hypertonie |
| I10.0 : Benigne essentielle Hypertonie |
| + I10.1 : Maligne essentielle Hypertonie (35) |
| + I10.9 : Essentielle Hypertonie, nicht näher |
| + I11 : Hypertensive Herzkrankheit (1642) |
| + I12 : Hypertensive Nierenkrankheit (165) |

(a) The catalog view displays inter alia the hierarchical structure of the ICD-10 catalog.

Es wurden 681 Fälle gefunden.

| Alter | Ao-root | Wurzel = ekta | I71 : Aortena | Geschlecht=M |
|-------|---------|---------------|---------------|--------------|
| 65 | 41 | x | | x |
| 76 | 42 | x | | |
| 84 | 42 | | | x |
| 76 | 41 | x | | x |
| 38 | 41 | x | | x |
| 61 | 42 | x | | x |
| 76 | 44 | x | x | x |
| 45 | 43 | x | | x |
| 80 | 44 | | | x |
| 88 | 41 | x | | x |

(b) In the result view of the data warehouse query hits are shown in a tabular style. Numeric values are displayed and boolean values are represented with a x.

| Name | Operator | Wert | Oder | Bezug |
|--|-------------|------|------|---------------|
| Alter (499875) | ▼ | | | |
| Ao-root Wert (mm) (17103) | ▼ > | 40 | | ▼ erster Wert |
| Geschlecht=M (240859) | ▼ | | | |
| Geschlecht=W (259010) | ▼ | | | |
| I71 : Aortenaneurysma und -dissektion (1429) | ▼ | | | |
| Wurzel = ektatisch (996) | ▼ vorhanden | | | |

(c) In the query view of the data warehouse properties of the attributes can be set. For every attribute (1st column) an operator (2nd column) and arguments (3rd column) can be defined.

Fig. 2: The main views of the graphical user interface of the data warehouse query tool (in German).

3.1 Setting

For this test, various queries have been made to the systems and the response time was measured. The database system is a Microsoft SQL Server [6] and the search platform is Apache Solr 4.8 [2]. The database schema is shown in Figure 3. It has been extended to the Example 1 with the columns ValueDec, which is a decimal(8,2) column, and the four columns (first, last, min and max), which can have the values "1" or null. Because some attributes can have more than one value in a case, these four columns mark, if the current is e.g. the first occurrence in the case. String values are stored in the normal value-field and numbers are stored in the decimal-field for a quicker access. The Info-table has in the first runs, shown in Table 1, just a small index on the two columns CaseID and AttrID. In the

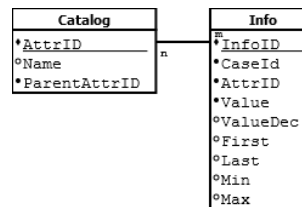


Fig. 3: The relational model of the database with the triple structure in the Info-table and additional flags for the first, last, min or max value of one attribute for one case.

last test-runs, shown in Table 2, the index is extended to the columns CaseID, AttrID, ValueDec and First, Last, Min, Max. In our test, the database did not use caching.

Solr has a document centered approach, so all facts of one case are pooled into one document and these documents are then indexed. The Solr-schema consists of dynamic fields for every attribute, which means, that every attribute has got its own index. An example query with a conjunction of two conditions looks as follows. (Natrium:[* TO 15] AND A_Brief:Grippe)

This query returns all documents/cases containing a value less than 150 in the numeric field Natrium and the term *Herzinfarkt* in the textfield A_Brief.

In our application, the first 100 hits and the total count of hits are displayed. For these two pieces of information are two requests in the database necessary, a top-100-query and a count(*)-query, Solr provides these two information in one query response.

14 settings were tested, three with boolean attributes, eight with numeric attributes and three with a text field. For every setting five queries have been send to the server and the response time has been measured. In Table 1 and 2, the average values of every five queries are shown in milliseconds.

Several diagnostic-attributes were used for the queries with the boolean-values. The average occurrence of an attribute was about 30 000 times, but some attributes had a occurrence of a few thousand, others had up to 100 000 occurrences. For the numeric tests, laboratory findings, which had about 100 000 occurrences, were queried. If a condition was applied to a numeric value, it was always a range query with a lower- and an upper-bound. 25 000 texts were used for the word-queries, which were realized with the like-operator. In the first word-test a single word was requested, in the second test a word with the wildcard * (search for a substring in word) and in the third test three AND-connected words were tested.

3.2 Results

Overall, it has been found, that a fully indexed database is faster than Solr, except the DB must join tables, then Solr is faster. If the DB is not fully indexed, Solr is always faster. As it is shown in Table 1 and 2 the database is only faster, if one attribute was queried and the index covered all used columns. The query for one boolean attribute is on the DB fast, because for a diagnosis query the value column does not need to be checked, because only positive records are stored in the database. So the query can be answered by only using the columns AttrID and CaseID, which are contained in the small index of the table and this is very effective.

But this does not work for the numeric queries, because it must be checked for every record if the value was in the selected range or if the flag was set in the First, Last, Min, Max field. In Table 2 the response times are shown for the small and the extended index for the DB. In the small index not all columns are included, which are required to answer the query. In contrast, the extended index contains all relevant columns. As you can see in Table 2, there is a big difference in the response time, if the DB can use an index or it can't. Solr can use its index on the numeric field to answer quickly, too.

If more than one attribute is queried, the database must join the Info-table with itself, because the facts are stored in a triple structure in the database. This is quite expensive and it explains, why Solr is faster, when more than one attribute is queried.

| | DB top 100 | | DB count | | DB sum | | Solr | |
|------------|------------|-------|----------|-------|--------|-------|------|-----|
| | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| 1 Bool | 40 | 30 | 32 | 25 | 73 | 34 | 115 | 67 |
| 3 Bool And | 2 267 | 1 665 | 140 | 62 | 2 407 | 1 696 | 48 | 55 |
| 3 Bool OR | 6 465 | 4464 | 143 | 97 | 6 608 | 4 537 | 235 | 49 |
| 1 Word | 18 | 2 | 3 280 | 839 | 3 248 | 840 | 218 | 101 |
| 1 Word * | 2 332 | 68 | 2 399 | 166 | 4 731 | 225 | 155 | 82 |
| 3 Words | 39 | 9 | 6 579 | 3 454 | 6 645 | 3 454 | 445 | 133 |

Table 1: Response time (Mean and Standard Deviation *SD*) in milliseconds for various queries. A comparison between a MS SQL DB and Apache Solr for querying boolean values or words in text fields. The boolean attributes were ORed and ANDED.

| | DB with small index | | | | DB with extended index | | | | Solr | | | |
|-----------------------|---------------------|-------|--------|-------|------------------------|-----|---------|-------|-------|-----|------|----|
| | top 100 | | count | | sum | | top 100 | | count | | sum | |
| | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| 1 Num. | 692 | 3 423 | 4 115 | 3 641 | 4 | 50 | 54 | 18 | 167 | 62 | | |
| 1 Num. with cond. | 403 | 966 | 1 369 | 1 083 | 5 | 184 | 189 | 56 | 242 | 140 | | |
| 3 Num. with cond. AND | 590 | 1 904 | 2 494 | 642 | 52 | 253 | 305 | 70 | 136 | 34 | | |
| 3 Num. with cond. OR | 10 438 | 5 086 | 15 524 | 2 028 | 7 207 | 378 | 7 585 | 2 723 | 234 | 106 | | |

Table 2: Response time (Mean and Standard Deviation *SD*) comparison between a MS SQL DB and Apache Solr for querying numeric values with and without conditions. The attributes were ORed and ANDED. All results are in milliseconds. The small DB-index does not contain all required columns, the extended index contains all.

Even the index of the DB doesn't help, which can be seen in the tests with three boolean or numeric attributes.

The database is quite fast with fetching the first 100 results for a single or a multiple word-query, but it is quite slow for a word-*-expression. The DB does not have to join tables to answer the 3-word-query, but the respond time is twice as long. Solr is much faster for text queries, because the texts are indexed here and in the DB they are not indexed.

It can be also observed, that the database is much slower, when the attributes are ORed and not ANDED.

But the main finding is, that Solr is on average drastically faster than the database system. It looks like, Solr doesn't take significant longer, if more attributes were queried.

If the tests are considered, where all necessary data was indexed, Solr is a bit slower, if one attribute was requested only, but if three attributes were queried, Solr is nearly ten times faster than the DB.

4 Additional Features of the Search Engine

By using the search engine, some new text query features are now possible.

Segment and span search Text fields can be efficiently searched. It is not only possible to search by multiple terms, but it is also possible to make span queries. A span query can be used to find multiple terms near each other, without requiring the terms to appear in a specified order. This can be a powerful tool for searching concepts, which consist of several words, like heart failure. Consider the following sentence:

(1) Heart: left ventricular failure.

It is possible to set the maximum distance, the two terms may be away from each other. The words *heart* and *failure* have a distance of three words, so this technique works well here. But it won't work in the next two examples:

(2) Kidney: renal failure. Heart: normal after transplantation.

(3) Heart: sinus rhythm, normal large left ventricle, aortic root normal width, right ventricular failure.

In example 2, the context of *failure* is *kidney* not *heart*. While this can be covered in the tool by determining an order for the terms, the span query approach is not suitable for the third example. The distance of the two terms is too far and the context can not be safely resolved.

Therefore, another approach was implemented: The segment search. Many text documents are structured like the examples above. One text consists of an enumeration of concepts like heart or kidney. Every concept is followed by a colon and list of findings. Therefore, it make sense to split these texts in segments, like in example 1 and 3. Example 2 would be split in two segments. This procedure is a preprocessing step, which makes it possible to search in these segments quickly. A query searches only in individual segments and doesn't mix them up. So, if you query example 2 with the terms *heart* and *failure*, you will get no hit.

Synonym search Another feature of the search engine implementation is, that queries are complemented with synonyms. Every term of a query is analyzed if it is a medical term, which has synonyms or abbreviations, these terms are added to the query. All synonyms are put in a OR-condition, which is satisfied, when one term appears in a document. An alternative approach is to handle the synonyms at index-time and not at query-time. With this feature, a higher recall can be achieved.

5 Related Work

An experimental comparison of RDF data management approaches in a SPARQL benchmark scenario showed, that none of the tested RDF schema was competitive to a comparable purely relational encoding [7].

An empirical study on performance comparison of Lucene and a relational database has been made by Jing et al. [5]. Apache Solr uses the Apache Lucene search library for building and querying the index. Jing tests a MS SQL Server, too, but with a full-text-index. Unfortunately, this feature was not available to us. Furthermore they query only one table and they don't make any joins. But their results also say, that Lucene

is faster than an unindexed database. Except, if combinational queries, with more than one where-clause, which could be ORed or ANDed, were tested, Lucene was on average quicker. Jing uses synthetic generated data and tested only queries without join operations, while we had real data with many joins.

The Léon Bérard Cancer Center in France [3] implemented their information retrieval systems also with Solr, but only as full-text search engine and not for structured data.

6 Conclusions

In this paper, we presented a brief overview over the main functions and the GUI of our clinical data warehouse query tool. We described the setting for our storage engines and our requirements. We showed and explained in several tests the advantages and disadvantages of relational database and Solr for query answering. It has been found, that Solr is faster than an unindexed database. If the DB was fully indexed, then it is faster as Solr, except when the DB must join tables. In that case, Solr is faster again.

References

1. World Health Organization (WHO) : International Classification of Diseases (ICD). <http://www.who.int/classifications/icd/en/> (2014), [Online; accessed 20-June-2014]
2. Apache Software Foundation: Apache Solr. <http://lucene.apache.org/solr/> (2014), [Online; accessed 20-June-2014]
3. Biron, P., Metzger, M.H., Pezet, C., Sebban, C., Barthuet, E., Durand, T.: An information retrieval system for computerized patient records in the context of a daily hospital practice: the example of the léon bérard cancer center (france). *Applied clinical informatics* 5(1), 191–205 (2014)
4. Broekstra, J., Kampman, A., Harmelen, F.v.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: *Proceedings of the First International Semantic Web Conference on The Semantic Web*. pp. 54–68. ISWC '02, Springer-Verlag, London, UK, UK (2002)
5. Jing, Y., Zhang, C., Wang, X.: An empirical study on performance comparison of lucene and relational database. In: *Communication Software and Networks, 2009. ICCSN '09. International Conference on*. pp. 336–340 (Feb 2009)
6. Microsoft: Microsoft SQL Server. <http://msdn.microsoft.com/en-us/library/bb545450.aspx> (2014), [Online; accessed 20-June-2014]
7. Schmidt, M., Hornung, T., Küchlin, N., Lausen, G., Pinkel, C.: An experimental comparison of rdf data management approaches in a sparql benchmark scenario. In: *Proceedings of the 7th International Conference on The Semantic Web*. pp. 82–97. ISWC '08, Springer-Verlag, Berlin, Heidelberg (2008)
8. W3C: W3C, SPARQL 1.1 Protocol. <http://www.w3.org/TR/sparql11-protocol/> (2014), [Online; accessed 20-June-2014]

Is Evaluating Visual Search Interfaces in Digital Libraries Still an Issue?*

Wilko van Hoek and Philipp Mayr

GESIS - Leibniz Institute for the Social Sciences, Unter Sachsenhausen 6-8, 50667
Cologne, Germany

{wilko.vanhoek,philipp.mayr}@gesis.org,
WWW home page: <http://www.gesis.org>

Abstract. Although various visual interfaces for digital libraries have been developed in prototypical systems, very few of these visual approaches have been integrated into today's digital libraries. In this position paper we argue that this is most likely due to the fact that the evaluation results of most visual systems lack comparability. There is no fix standard on how to evaluate visual interactive user interfaces. Therefore it is not possible to identify which approach is more suitable for a certain context. We feel that the comparability of evaluation results could be improved by building a common evaluation setup consisting of a reference system, based on a standardized corpus with fixed tasks and a panel for possible participants.

Keywords: Visual User Interfaces, Digital Libraries, Interactive Information Retrieval, User Studies, Evaluation Methodology

1 Introduction

In the last twenty years of research on visual interfaces for digital libraries (DLs) a variety of approaches has been proposed and many visual search prototypes have been developed to support the user of DLs in his search process. For every part of the search process techniques exist to support the user. However, most of these techniques have not found their way into today's DLs. On the contrary nearly all prototypes have been discontinued. Most ideas have not been evaluated more than once in a relatively small study.

The main question is: Why have most of the research results not been adapted into today's DLs? One simple answer could be that this is the typical evolution of scientific research. Many ideas are not supposed to be commercially beneficial, adaptable in large scale live environments or not successful due to various other reasons. In [5] we took a look at the different techniques and the studies that have been conducted, so far we do not feel that the answer is that simple.

* *Copyright* © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

After taking a closer look at the results of a line-up of different studies [11],[12],[13],[1],[4],[7],[2] and [8], we can observe that usually quantitative results on the task performance and the accuracy of participants in a visual IR system are comparably poor or at least equally good as a strictly text-based system. On the other side, in accompanying questionnaires, the participants' opinions on the same visual IR system were positive and in favour of the system. Here seems to be a mismatch that needs a closer examination.

2 State of the Art

In the following we briefly review a selection of well-known publications which describe and evaluate information visualization systems for digital libraries [5]. The section will introduce seven prototype systems that provide a visual access to data and studies that were conducted to evaluate these prototypes. We will take a closer look at five different facets of how the studies were conducted. We will try to identify:

1. the main aim of the study (measurement of usability, performance or the cognitive effects),
2. the type of evaluation method that was used (e.g. A/B testing, between- or within-subject design),
3. how the study was conducted (e.g. task-based, laboratory),
4. details on the subjects (e.g. group size, expertise),
5. the document corpus that was used (e.g. newspaper articles, digital libraries).

2.1 ENVISION

The ENVISION system [11] is an early attempt to display search results in a 2-dimensional grid. Metadata fields like author or publication year could be selected for the two axes and the system would position the search results represented by icons within the resulting grid.

In the study that was conducted, the main aim was to evaluate the usability of such a system. This was done without A/B testing. The users were asked to fulfill several tasks that involved using different interaction methods in the system. The tasks were not aligned with those of other studies. The study took place in a laboratory environment with one expert, two graduate students and two undergraduate students. As document corpus, scientific publications were used. There are no further details about the corpus.

2.2 NIRVE

In the NIRVE system [12], search results can be displayed on a 3-dimensional globe, where clusters of documents are displayed as boxes emerging from the globe. The thickness of a box represents the number of documents in the cluster. Documents with the same combination of query terms build a document cluster.

Clusters of documents containing only a few query terms are displayed near the south pole, clusters of documents containing more query terms near the north pole (cf. figure 1).

In the study that was conducted to assess the system, a text-based IR system a 2-dimensional version of the globe and the globe-based system were compared. The aim was to assess usability and performance of the globe system. The performance was measured between the three systems. The study was conducted in a task-based laboratory environment. The tasks were not aligned with those of other studies. The subject group consisted of 15 participants of which 6 were experts and 9 students. The underlying corpus consisted of the news stories of the Associated Press from the year 1988.

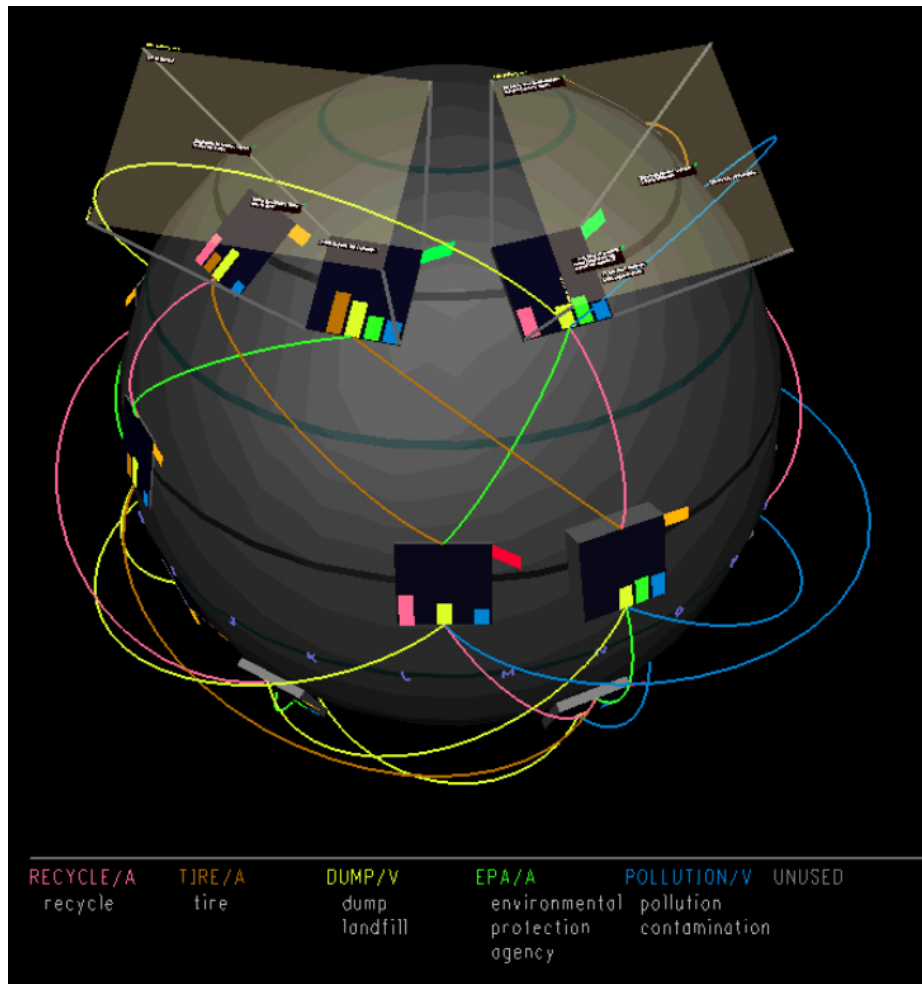


Fig. 1. Globe view of the NIRVE system

2.3 GRIDL

In the GRIDL system [13] the search result list is presented in a 2-dimensional grid similar to the presentation in ENVISION [11]. Here the focus lays on overcoming the problem of overcrowded rows, columns or cells. An attempt was made to overcome this issue by utilizing solutions such as tool tips or further hierarchical grouping.

Two consecutive studies were conducted on the system. The main aim was to assess the usability of the system. The studies were done in a task-based laboratory environment without A/B testing. The tasks were not aligned with other studies. The first subject group consisted of 8 graduate students and the second of 24 subjects, of which 10 came from the field of library science, 8 from the field of computer science and 6 subjects from other fields. As corpus, metadata of scientific publications within the database of the Computer Science Department Library at the University of Maryland was used.

2.4 InfoSky

The InfoSky system [1] provides the user with two different alternatives to browse and query large data sets of hierarchically structured documents. The first one is a tree browser similar to the file browser in operation systems. The second one is a so-called telescope browser. In the telescope browser, documents are represented as points on a black background modelled after the night sky. The documents hierarchy and cosine similarity are used to position and group the documents. In this way clusters are formed, consisting of documents that bear a certain resemblance to each other.

The system was evaluated in a first study [1]. Based on the results of this first study the system has been improved and extended. It then was evaluated in a second study [4]. The aim of both studies was to assess the performance of users using the telescope browser. Therefore A/B testing with a crossover design was used to assess the user performance with telescope and tree browser. Both studies were conducted in a task-based laboratory environment. The tasks were not aligned with other studies. Moreover, the tasks were changed for the second study. The first study took place with 8 subjects, the second with 9 subjects. No further details on the background of the participants were provided. The corpus of both studies was a set of 80,000 newspaper articles from the German *Sueddeutsche Zeitung*.

2.5 VIDLS

In the VIDLS System [7], three visual interfaces have been implemented. An overview for the result list of searches, and two detailed document views. The system relies on full text documents of books, as the visualization uses the content and the index of the documents. The overview uses a 2-dimensional grid layout following the GRIDL System [13]. Here books are represented as circles. The size of the circle resembles the normalized number of pages on which the

search terms occur (cf. figure 2). In the detailed view, the book's index is used to display distribution and frequency of index terms and search terms within the document.

The main aim of the study was to assess the usability of the visualizations. Therefore A/B testing with a within-subject design was used. This was in a laboratory environment. The study was divided in multiple sessions each with three to five students. The only task was to search for books, once with both systems, the text-based system and the VIDLS system. A post search survey was used to assess the usability by asking the users about their impressions on the system. No details on the corpus are provided.

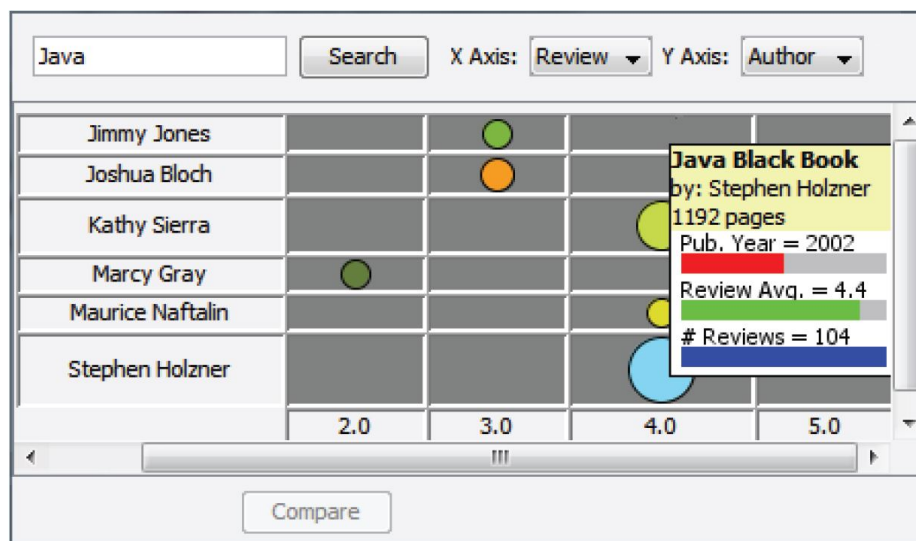


Fig. 2. Result list overview in the VIDLS system

2.6 PivotPaths

In the PivotPath system [2] the search result list as visualization canvas is presented as an information space that contains multiple facets and relations, such as authors, keywords, and citations of academic publications, or actors and genres of movies. PivotPath focuses on selecting items from facet lists (pivot operations) resulting in direct changes on the interface. The PivotPaths interface exposes faceted relations as visual paths in arrangements that invite the viewer to 'take a stroll'.

Two participants' observational studies (academic publications and movie collections) were conducted in an intranet deployment of the system. The authors did semi-structured interviews where participants could comment on questions

and executed tasks. The studies were done in a task-based laboratory environment without A/B testing. The tasks of the user sessions were not aligned with other studies. The intranet study attracted 290 participants with 211 actual-use sessions. The authors report detailed on anecdotal email feedback of their participants. As corpus, metadata of scientific computer science publications from Microsoft Academic Search and movies from the Internet Movie Database were used.

2.7 INVISQUE

In the INVISQUE system [8] the query formulation and result list presentation has been moved into one interface. In this system search results of different search can be displayed on an infinite pane. The result sets of different searches can be merged by dragging one onto another. In this way complex boolean queries can be generated on a visual level by working with the result sets of queries.

In the study the decision was made not to evaluate performance or usability, but to assess the sense-making process of experts using the system. Therefore six senior university librarians were asked to identify three central authors of a field that was unknown to them. Interaction-logs, video recording and survey were analyzed to evaluate the study. As corpus the metadata of publication from the ACM SIGCHI conference from 1982 to 2011 was used.

In the following section we will develop and discuss positions which we think are still crucial in the domain of visual search interface in DL. We will emphasize to develop a more standardized evaluation setup for such interfaces. We are aware of the fact that an experimental approach has already been implemented during the TREC interactive tracks (TREC 3-12) [3] that follows the some identical arguments and observations we are discussing in this paper and thus, that parts of the following positions have already been discussed. Especially in TREC-6 an almost identical approach has been applied to assess cross-site performance [9]. The results of this analysis were mixed. It was not possible to reliably compare the performances of the different systems. It was emphasized that by further investigating cross-sites experiments more reliable methods could be generated. Also, the most problematic factors influencing the results of the comparison were the relevance assessors and the fact that the subjects differed throughout the different studies. We therefore think that the ideas and findings that have led to developing a cross-site analysis are still relevant and in our analysis we could still identify those shortcomings in interface evaluation methodology even in more recent studies.

3 Discussion

Position A: Diversity of evaluation aims. Throughout all studies we could see that there has been a clear aim that was followed. Usability and performance are two central aspects of systems, but as [2] and [8] have shown, there are other aspects that are important when it comes to the question of the suitability of an

interface for a DL. Anyhow, except for [2] and [8], we are missing a real discussion about why usability or performance was considered to be more relevant than the other aspects. Also [2] and [8] make clear that they are interested in other aspects, but then they ignore usability and performance completely. A system that is hard to use cannot be considered to optimize performance or serendipity. In addition, performance has its influence on other aspects as well. We strongly feel that the various aspects of the systems are co-dependent on each other. Instead of assessing only one aspect, multiple aspects should be assessed. At least a usability and a performance study should be conducted. We do see that this implies a more complex study design and costs more effort. However, this might be compensated by creating a standardized evaluation design and environment.

Position B: Missing shared design methodology. When thinking of standardization of the evaluation, one needs to decide which study design to use for which types of study. In all cases where performance was measured, A/B testing was used. Obviously this is a good idea, as performance implies benchmarking, which does not make a lot of sense without reference values. These reference values can be generated by measuring the task performance in a reference system. Usability on the other side can be assessed without a reference system. There might be a way to include a usability study into a performance study and to reduce the need of conducting two separate studies. In our review we observed a missing shared design methodology which would be very essential to reach comparability and reproducibility in this domain.

Position C: Need of a common reference system. In total A/B testing seems to be an important tool to evaluate system. But are results of A/B testing really worth the effort of comparing two systems? In an ideal world, one would assume that when comparing systems A and B and systems A and C one could make assumptions about the relation between B and C. But when A is not fix this transitivity is lost. We have seen A/B testing being used in [12],[1], and [7]. In all three studies an own implementation of a text-based system was used as reference. There is no clarification in how far the three text-based systems are comparable. Thus, we do not know anything about the relation between the three prototypes. We do not know which changes have improved or worsened the usability or performance of a grid-based visualization in comparison to a text-based system. We therefore propose to build a common reference system. This would be a more suitable baseline for evaluating system in an A/B testing scenario. During the TREC-6 evaluation it was impossible to make the reference system and the different experimental system accessible from the same spot. Therefore the participants needed to implement their reference system at their own institute to conduct the user studies [9]. With today's digital infrastructure it would be easier to access a reference system remotely. Thus the effort of conducting a study with a reference system is reduced significantly and it would be possible to test two experimental systems A and C and a reference system B in the same study with the same subjects.

Position D: Need for standardized test collections. If we follow the trail of thought in our position C, it becomes clear that building a reference

system is not enough to improve comparability. The results of a study are also influenced by other factors. A visualization technique might be suitable for a certain set of documents, but not even applicable for another. Thus, the reference system should be combinable with different document corpora. But using different corpora for similar systems is not a good idea. In [12] and [1] for example, the underlying corpus was a set of newspaper articles, but not the same set. This degrades the comparability of the study results. In addition, in TREC-6 of the TREC interactive tracks only one collection (The Financial Times of London 1991-1994 collection) was used. This collection was not suitable for all experimental interfaces as they focused on different aspects. We feel it might be a good idea to create a set of standardized test collections, so that similar systems can be assessed in the same environment.

Position E: Need of shared and standardized tasks. Another crucial point regarding the comparability of task-based studies are the tasks themselves. As long as every study defines its own tasks, it is not possible to compare the results easily. Aligning tasks is not a trivial task as different systems aim at different steps of the search process. On the other hand, the systems ENVISION [11], GRIDL [13], and VIDLS [13] for example, all display search results in a 2-dimensional grid and refer to each other. Here arises the question, why are there no tasks that were aligned with previous studies? Building up a set of tasks for typical activities in DLs, so that researchers can compare the usability and performance of different systems is a next desideratum.

Position F: Subject-based evaluation. The last issue we would like to address is the question on the subject group. In a laboratory environment, it is expensive to have many subjects. Experts are more difficult to get for a study than students. Throughout the eight studies we have seen a variety of expertises and sizes of the subject groups. In how far can results be comparable when the subject groups vary that much? What can be done to improve this situation? One way could be to establish a panel of subjects, that allows to contact the same set subjects for multiple studies (see e.g. [6]). This seems to be a very important aspect. In [9] it is argued that the cross-site analysis results are strongly biased by the differences in the subject groups which were involved at the various studies.

4 Outlook

In this paper we discussed some issues concerning the evaluation of visual interfaces for DLs. The main issue here is the comparability of results. We reviewed a set of studies on well-known interfaces. Comparing the studies we could identify some possible points for improvement. We propose to build up a common reference system, where methods and designs are fixed, based on standardized document corpora. The system should be evaluated with standardized corpora like the iSearch test collection [10] or openly accessible publications from repositories like PubMed Central ¹ or arXiv ². In addition a set of tasks should be

¹ <http://www.ncbi.nlm.nih.gov/pmc/>

² <http://www.arxiv.org>

defined that reflect common information needs in DLs. Combined with a panel of participants [6] a suitable environment to conduct more comparable studies could be created. Altogether, the development of such an environment is a complex and time-consuming project. This effort should be worked on collaboratively to benefit from the experiences of the researchers in the field. A workshop on a topic related conference like the IiX or CHI is in preparation. The project could also benefit from collaboration with the TREC and CLEF groups to make use of their experiences with standardizing corpora and evaluation settings.

References

1. Andrews, K., Kienreich, W., Sabol, V., Becker, J., Droschl, G., Kappe, F., Granitzer, M., Auer, P., Tochtermann, K.: The InfoSky visual explorer: exploiting hierarchical structure and document similarities. *Information Visualization* 1(3-4), 166–181 (Dec 2002)
2. Dörk, M., Riche, N.H., Ramos, G., Dumais, S.: PivotPaths: strolling through faceted information spaces. *IEEE Transactions on Visualization and Computer Graphics* 18(12), 2709–2718 (Dec 2012)
3. Dumais, S.T., Belkin, N.: The TREC interactive tracks: Putting the user into search. In: Voorhees, E.M., Harman, D.K. (eds.) *TREC: Experiment and Evaluation in Information Retrieval*, pp. 123–152. Digital libraries and electronic publishing, MIT Press
4. Granitzer, M., Kienreich, W., Sabol, V., Andrews, K., Klieber, W.: Evaluating a system for interactive exploration of large, hierarchically structured document repositories. In: *Proc. IEEE Symposium on Information Visualization INFOVIS 2004*. pp. 127–134. IEEE (2004)
5. van Hoek, W., Mayr, P.: Assessing Visualization Techniques for the Search Process in Digital Libraries. In: Keller, S.A., Schneider, R., Volk, B. (eds.) *Wissensorganisation und -repräsentation mit digitalen Technologien*, pp. 63–85. DeGruyter Saur (2014), <http://arxiv.org/abs/1304.4119>
6. Kern, D., Mutschke, P., Mayr, P.: Establishing an Online Access Panel for Interactive Information Retrieval Research. In: *Digital Libraries 2014. ACM/IEEE Joint Conference on Digital Libraries (JCDL 2014) and International Conference on Theory and Practice of Digital Libraries (TPDL 2014)*. ACM (2014), <http://arxiv.org/abs/1407.1540>
7. Kim, B., Scott, J., Kim, S.E.: Exploring digital libraries through visual interfaces. *Digital Libraries - Methods and Applications* pp. 123–136 (Jan 2011)
8. Kodagoda, N., Attfield, S., Wong, B., Rooney, C., Choudhury, S.: Using interactive visual reasoning to support sense-making: Implications for design. *IEEE Transactions on Visualization and Computer Graphics* 19(12), 2217–2226 (Dec 2013)
9. Lagergren, E., Over, P.: Comparing interactive information retrieval systems across sites: the TREC-6 interactive track matrix experiment. In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 164–172. ACM
10. Lykke, M., Larsen, B., Lund, H., Ingwersen, P.: Developing a test collection for the evaluation of integrated search. In: *Advances in Information Retrieval*, pp. 627–630. Springer (2010)

11. Nowell, L.T., France, R.K., Hix, D., Heath, L.S., Fox, E.A.: Visualizing search results: Some alternatives to query-document similarity. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 6775. SIGIR '96, ACM, New York, NY, USA (1996)
12. Sebrechts, M.M., Cugini, J.V., Laskowski, S.J., Vasilakis, J., Miller, M.S.: Visualization of search results. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 3–10. SIGIR '99, ACM Press, Berkeley, California, USA (1999)
13. Shneiderman, B., Feldman, D., Rose, A., Grau, X.F.: Visualizing digital library search results with categorical and hierarchical axes. pp. 57–66. ACM Press (2000)

IIRpanel – An Online Access Panel for Interactive Information Retrieval Research

Dagmar Kern¹, Peter Mutschke¹, Philipp Mayr¹

¹ GESIS – Leibniz Institute for the Social Sciences, Unter Sachsenhausen 6-8,
50667 Cologne, Germany

{dagmar.kern, peter.mutschke, philipp.mayr}@gesis.org

Abstract. We propose an online access panel to support the evaluation process of Interactive Information Retrieval (IIR) systems - called IIRpanel. By maintaining an online access panel with users of IIR systems we assume that the recurring effort to recruit participants for web-based as well as for lab studies can be minimized. We target on using the online access panel not only for our own development processes but to open it for other interested researchers in the field of IIR. In this paper we present the concept of IIRpanel as well as first implementation details.

1 Introduction

Interactive Information Retrieval (IIR) becomes more and more important in IR research and thus, user involvement in the developing process of IR systems is essential to produce useful and usable interactive products [3]. Following a user-centered design approach [5] the development of different IIR systems includes user involvement in every step of the design cycle through contextual observations, web-surveys, usability tests, quantitative ratings, retrieval tests and performance measures. However, one of the probably most time-consuming issue during the evaluation process of new IIR systems is recruiting participants for online-surveys, interviews or lab studies. Large research companies like Google and Microsoft address this issue by managing pools of participants (e.g. Microsoft research panel¹ and Google User Experience Research Studies²) where interested users can register to take part in online surveys or even lab studies. These participant pools are exclusively maintained for testing their own products and prototypes. Whereas tools like Mechanical Turk³ offer a general opportunity for researchers who are looking for participants to take part in short web-based user studies [4]. In a relative short period of time a large number of participants can be reached. Mechanical Turk is well suitable for user studies targeting on common internet users with no specific educational or contextual background. Addressing a specific user group with special skills or informational settings is however not relia-

¹ <https://www.microsoftonlinepanel.com/Portal/default.aspx>

² <http://www.google.com/usability/>

³ <https://www.mturk.com/mturk/welcome>

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

bly possible, and therefore its benefit for evaluating specific IIR systems and tasks seems to be very low.

Online access panels, in contrary, provide the opportunity to limit a sample of participants to a specific target group. While the usage of such a panel is common for market research⁴, for data collection in the Social Sciences⁵ or even for evaluating existing websites⁶ it is often not considered during web product development processes, and hence have – to the best of our knowledge – so far not been taken into account for the continuous development of IIR systems. An online access panel is a pool of previously recruited participants who have agreed to take part in regularly (web-based) studies [2]. Online access panels with panelists representing the desired target group or with even real end-users of existing products could help to speed-up the recruiting process for user studies and the management of participants who are needed for repeated examinations. A further major benefit of an online access panel is seen in the opportunity to continuously investigate novel IIR approaches with a stable population of users. Follow-up studies with the same user groups might also provide information about changes of information needs over time. Assuming that the IIR community has the same or at least a similar target group in mind while developing new IIR functionalities, we propose to set up an online access panel for IIR research which is developed and maintained in cooperation. We target on establishing an international group of researchers who collaborate in building up this unique instrument for sharing panelists in one dedicated place.

In the following we describe the basic concept of the proposed online access panel – called IIRpanel⁷, its architecture and its initial implementation at GESIS.

2 IIRpanel – Concept

The initial idea of maintaining an online access panel originates from the need to have a commonly used proband management system for GESIS. So far for nearly all user studies a new process for recruiting participants has been started. The online access panel in contrast aims at establishing a pool of real users of IIR systems having a strong interest in taking part either in improvement processes of existing systems or in evaluating new IIR functionalities. The idea is that participants sign up only once for the IIRpanel and researchers performing user studies can send participant calls to all or to a sample of this panel with very little effort.

On a sign-up page the users are asked for demographical data like gender, year of birth, their current job, their current profession, research field of interest, and so on. The users can decide if they would like to take part in web-surveys only, in lab studies only or both. For the lab studies we also asked for a location to better arrange lab studies with participants available in a specific area. All this data is stored in a database taking current law of privacy into account. Through a self-service page the user

⁴ E.g. <http://www.usamp.com/panel.html>

⁵ E.g. <http://www.gesis.org/en/services/data-collection/gesis-panel/>

⁶ E.g. <http://panel.webeffective.keynote.com/Default.asp?>

⁷ <http://www.gesis.org/iirpanel>

has always access to his data, the ability to delete his own profile, to change his demographical data, to communicate with the panel provider and to access his participation history or even to watch results of studies in which he took part.

Collaborating internal and external researchers performing a user study can log in to the IIRpanel and select a sample based on the stored data. An email with a web-survey link or an invitation for a lab user study can be send directly through the IIR-panel. After the user study is finished further administrative data should be stored like title of the study, date of invitation, if the user participated in the study and the date of the study. The participant history which is derived from this information easily allows the researchers to perform a follow-up study with the same set of users.

Our main intention in the first step is to establish a set of volunteered probands in a specific research domain who are willing to take part in a series of web surveys as well as in lab studies. After reaching a critical mass of participants, the often criticized aspect of representativeness of open access panels [6] shall be addressed for example by establishing different subpanels that fulfill the criteria of representativeness.

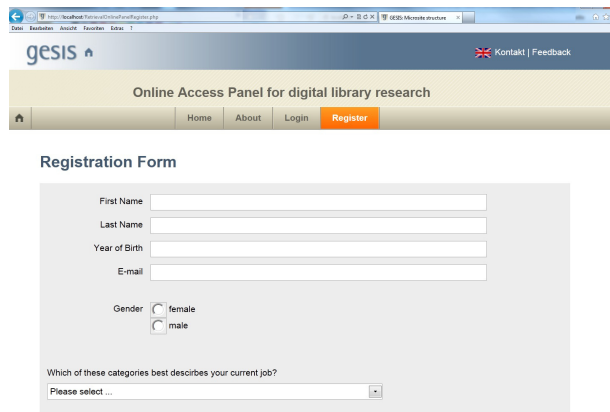


Fig. 1. IIRpanel - Registration form

3 IIRpanel - Architecture

As a basis for the IIRpanel we used phpPanelAdmin developed by Anja Göritz [2]. phpAdmin⁸ is an open source tool under the GNU license for managing online access panels or similar sampling lists. As a web-based platform phpPanelAdmin aims at quickly setting up and managing online panels. As the name already suggests it is developed in php and makes use of HTML and JavaScript. For panelists' data storage a MySQL database is required. The administrative functionalities include searching for panelists, managing panelists' data and variables, identifying duplicates, drawing

⁸ www.goeritz.net/panelware

samples, creating and managing e-mail templates, sending e-mails to panelists, and displaying panel statistics [2]. To fulfill our requirements we completed the tool with additional functionalities for user self-services as well as for managing participation history. In Figure 1 the registration form of the current implementation of the online access panel is shown.

4 Conclusion and Outlook

We introduced a collaborative online access panel for supporting evaluations of IIR systems. The panel infrastructure proposed aims at minimizing the recruiting process of participants for web-surveys as well as for lab studies and at building-up an international pool of various types of IIR system users to speed up IIR evaluations in the future. Our next steps are recruiting panelists through announcements on our websites as well as in our IR systems, by telephone calls, mailings and word-of-mouth advertising (according to [1]), evaluating a pilot application of the panel with GESIS users, and finally establishing a consortium of international researchers who are interested in supporting the maintenance as well as the further development of the IIR panel. Given the fact that we are still at the beginning of the developing process there are a lot of open questions that have also to be discussed with the consortium as well, These are for example how to balance participants' workload (send out one invitation per week vs. sending out 10 invitation on a single day), how to control participants' learning effects with IIR systems, how to make study data available for other researchers, how to deal with different legal aspects, how to address language barriers (usually participants are more comfortable in filling in questionnaires in their mother tongue) and so on. By using IIRpanel we expect benefits for major IIR research issues, such as defining and formulation of adequate search tasks and queries, information behavior, needs over time, usability of search interfaces, utility and usability of search histories, evaluation of whole search sessions performing longitudinal studies on the evolution of information needs, but also the benefits and drawbacks of such an online access panel itself can be addressed as a new area in IIR research.

5 References

1. ISO 26362:2009 Access panels in market, opinion and social research – Vocabulary and service requirements, (2009)
2. Göritz, A.S.: Building and managing an online panel with phpPanelAdmin. In: Behavior research methods 41, 4, (2009) 1177-1182
3. Kelly, D.: Methods for evaluating interactive information retrieval systems with users. In: Foundations and Trends in Information Retrieval 3.1-2, (2009), 1-224
4. Kittur, A., Chi, E.H. Suh, B.: Crowdsourcing user studies with Mechanical Turk. In Proc. CHI'08, Florence, Italy, (2008), 453-456
5. Norman, D.A., Draper, S.W.: User centered system design. In: New Perspectives on Human-Computer Interaction, L. Erlbaum Associates Inc., Hillsdale, NJ, (1986)
6. Stoop, I., Wittenberg, M.: Access panels and online research, panacea or pitfall? In: Proc. DANS Symposium, Amsterdam. Amsterdam, Netherlands, (2006)

Towards Ambient Search ^{*}

Stephan Radeck-Arneth^{1,2}, Chris Biemann², and Dirk Schnelle-Walka¹

¹ Telecooperation, Dep. Computer Science, TU Darmstadt, Germany

² Language Technology, Dep. Computer Science, TU Darmstadt, Germany

Abstract. In ongoing discussions participants tend to pick up their smart phones to retrieve relevant information for clarification, severely hampering the flow of the discussion. We introduce ambient search as a variant of information retrieval where a system unobtrusively provides relevant information snippets in the background without the need to steer devices actively. In this demo paper, we describe a first prototype of our ongoing research activities towards such a system.

1 Introduction

Phubbing describes a social problem where others are being ignored in favor of a mobile phone [3]. This may be done to retrieve information for clarification of facts for the current discussion, but still hampers the flow of conversation. We propose a system, which follows the discussion and returns related information in the background without requiring users to pick up and actively interact with devices. We define this as *ambient search*, featuring (i) real time information retrieval, (ii) presentation of topic-related information snippets and (iii) passive behavior. The system will unobtrusively present topic-related information snippets while the discussion continues. The dialog partners may or may not use them to collaboratively retrieve more detailed information.

We place ambient search into the continuum between Human Computer Interaction and Information Retrieval (IR). In this paper we introduce our first efforts towards the realization of such a system.

2 Related Work

Anzalone et al. [1] introduced a topic recognition system for social robots. They define TF-ITF to calculate the relevance per word to predefined topics. The definition relates to TF-IDF that describes the relevance of words to a document. Similarly, words with a high TF-ITF weight are considered to be more relevant for a topic. They also consider topic recognition to be helpful in the presence of speech recognition errors. Along these lines, Stas et al. [8] suggested an algorithm to build robust language models for a specific domain. They separated heterogeneous text data into binary domain classes that

^{*} Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

improved the model perplexity for a transcription and dictation system for the Slovak language. Such a strategy might be helpful in detecting topics and improve recognition accuracy.

Snippets are a well-known strategy for text summarization [6] and feature characteristics of dynamic summaries. Consequently, their computation needs to be fast since they cannot be precomputed off-line and must be synthesized based on the query results.

Personalized IR systems are, amongst others, investigated by Jeh et al. [4]. They extended the well-known PageRank algorithm to a personalized form.

3 Approach

An overview of the envisioned architecture is shown in Fig. 1. We regard the interplay of components as information streams.

An *automated speech recognizer* (ASR) continuously processes the audio input stream of an ongoing discussion and forwards the recognized utterances to a *topic detector* to extract the meaningful parts. For now, we restrict this to detection of nouns. We are currently investigating the appropriateness of the selection process and strategies to combine the most recent nouns in the streams into subsets with various logical operators for an optimized balance between precision and coverage. Moreover, this module queries a structured document collection for a set of related documents. A *snippet filter* is responsible for filtering these documents to snippets, i.e. the relevant passages within a document. A *formatter* then highlights the topic identifiers (nouns in our case) to make the appearance more comprehensible. Hence, we expect users to be able to easily understand the causal relation between speech and the presented results. At this stage of development, we do not consider possible problems as a result of context switches during discussions or cross-talk.

We see the following advantages of this approach: (i) The assistant stays in the background without disturbing the user. (ii) The user can access the displayed snippets on demand. (iii) The snippet continuously updates the available snippets. However, these advantages will have to be validated in user studies.

Our current prototype provides basic implementations for all needed components and enables us to get first experiences with ambient search. We employed Sphinx [5] for ASR using our own models for German [7]. Nouns are identified by a pretree-based POS-Tagger [2]. For the document collection we are using the German Wikipedia indexed by Solr³.

Our current prototype provides basic implementations for all needed components and enables us to get first experiences with ambient search. We employed Sphinx [5] for ASR using our own models for German [7]. Nouns are identified by a pretree-based POS-Tagger [2]. For the document collection we are using the German Wikipedia indexed by Solr³.

³ <http://lucene.apache.org/solr/>

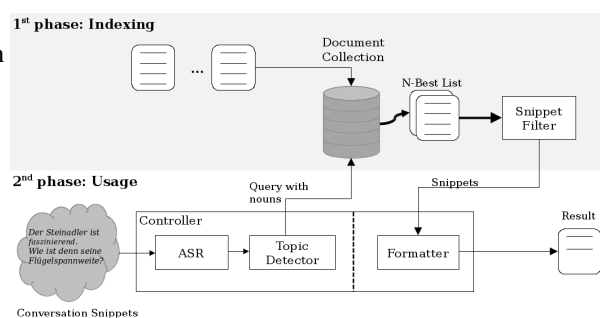


Fig. 1. Ambient Search Architecture

4 Conclusion & Future Work

We took the first steps towards our vision of ambient search as a basis for further investigation. Future and ongoing research activities cope with extracting appropriate keywords from the dialog stream. For this, we are currently inspecting transcribed dialogs to find possible strategies and we are developing an automated evaluation framework. Furthermore we are looking into user interaction with ambient search and its applicability to group discussions. Another important task we have to tackle for making ambient search practicable is to improve ASR performance, especially in noisy environments.

Acknowledgements

This work was partly supported by the Bundesministerium für Bildung und Forschung (BMBF), Germany under the programme “KMU-innovativ: Mensch-Technik-Interaktion für den demografischen Wandel”.

References

1. Anzalone, S.M., Yoshikawa, Y., Ishiguro, H., Menegatti, E., Enrico, P., Sorbello, R.: A topic recognition system for real world human-robot conversations. In: *Intelligent Autonomous Systems*, vol. 12, pp. 383–391. Springer (2013)
2. Biemann, C., Quasthoff, U., Heyer, G., Holz, F.: ASV Toolbox: a Modular Collection of Language Exploration Tools. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*. pp. 1760–1767. European Language Resources Association (ELRA), Marrakech, Morocco (2008)
3. Coehoorn, M.: Phubbing? An absurd design intervention for redefining smart-phone usage. Master’s thesis, TU Delft, Delft University of Technology (2014)
4. Jeh, G., Widom, J.: Scaling personalized web search. In: *Proceedings of the 12th International Conference on World Wide Web*. pp. 271–279. WWW ’03, ACM (2003), <http://doi.acm.org/10.1145/775152.775191>
5. Lamere, P., Kwok, P., Gouvêa, E., Raj, B., Singh, R., Walker, W., Warmuth, M., Wolf, P.: The CMU Sphinx-4 Speech Recognition System. In: *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003)*, Hong Kong. pp. 2–5 (2003), http://www.cs.cmu.edu/~rsingh/homepage/papers/icassp03-sphinx4_2.pdf
6. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to information retrieval*, vol. 1. Cambridge University Press Cambridge (2008)
7. Schnelle-Walka, D., Radeck-Arneth, S., Biemann, C., Radomski, S.: An Open Source Corpus and Recording Software for Distant Speech Recognition with the Microsoft Kinect. In: *Speech Communication; 11. ITG Symposium*. p. 4. VDE (2014), (to appear)
8. Stas, J., Juhar, J., Hladek, D.: Classification of heterogeneous text data for robust domain-specific language modeling. *EURASIP Journal on Audio, Speech, and Music Processing* 22 (2014), <http://dx.doi.org/10.1186/1687-4722-2014-14>

**FGWM: Workshop of the
“Fachgruppe Wissensmanagement”
(SIG Knowledge Management)**

Case-based Reasoning in the Cloud

Mirjam Minor

Goethe University Frankfurt am Main, Germany
minor@cs.uni-frankfurt.de

Abstract. Cloud computing has its roots clearly in industry. A huge variety of successful cloud services have been developed in a rather pragmatic manner. Recently, Cloud computing is increasing its attraction also as a research topic. Many basic questions especially in cloud management still remain open. *Cloud management* deals with management methods for provisioning and use of cloud services [1]. For instance, rapid scalability is often achieved by a massive overprovisioning of resources today, which causes overcharged prices and a waste of energy. A systematic approach including thorough concepts for monitoring, analysis, search, reuse, orchestration and configuration of cloud services might be extremely beneficial for both cloud providers and users.

The keynote will highlight the potential of intelligent cloud management methods, particularly from the field of Case-based Reasoning. *Case-based Reasoning* (CBR) is a sub-area of Artificial Intelligence that deals with the reuse of experience recorded in cases [5]. Recent work on case-based, automated cloud management [3, 4] will be presented. Future research issues for CBR in the cloud will be investigated, including the semantic description and retrieval of cloud services, the case-based analysis of time series [2] applicable to the monitoring of service level agreements, for instance, and the potential "cloudification" of CBR methods such as rapidly scalable case retrieval and case adaptation. Potential business application scenarios will be discussed.

The aim of this keynote is to demonstrate that, beyond the buzzword, cloud computing provides novel, intriguing opportunities for research.

References

1. C. Baun, M. Kunze, J. Nimis, and S. Tai. *Cloud Computing - Web-Based Dynamic IT Services*. Springer, 2011.
2. Odd Erik Gundersen. Toward measuring the similarity of complex event sequences in real-time. In Belén Díaz Agudo and Ian Watson, editors, *Case-Based Reasoning Research and Development*, number 7466 in Lecture Notes in Computer Science, pages 107–121. Springer Berlin Heidelberg, January 2012.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

3. Michael Maurer, Ivona Brandic, and Rizos Sakellariou. Adaptive resource configuration for cloud infrastructure management. *Future Generation Computer Systems*, 29(2):472–487, 2013.
4. Mirjam Minor and Eric Schulte-Zurhausen. Towards process-oriented cloud management with case-based reasoning. In *Proc. ICCBR 2014*, LNCS 8766, pages 303 – 312. Springer, 2014. The original publication is available at www.springerlink.com.
5. Michael M. Richter and Rosina Weber. *Case-Based Reasoning: A Textbook*. Springer, auflage: 2013 edition, November 2013.

Workflow Streams: A Means for Compositional Workflow Adaptation in Process-Oriented CBR

Gilbert Müller and Ralph Bergmann

Business Information Systems II
University of Trier
54286 Trier, Germany
[muellerg] [bergmann]@uni-trier.de,
<http://www.wi2.uni-trier.de>

Abstract. This paper presents a novel approach to compositional adaptation of workflows, thus addressing the adaptation step in process-oriented case-based reasoning. Unlike previous approaches to adaptation, the proposed approach does not require additional adaptation knowledge. Instead, the available case base of workflows is analyzed and each case is decomposed into meaningful subcomponents, called workflow streams. During adaptation, deficiencies in the retrieved case are incrementally compensated by replacing fragments of the retrieved case by appropriate workflow streams. An empirical evaluation in the domain of cooking workflows demonstrates the feasibility of the approach and shows that the quality of adapted cases is very close to the quality of the original cases in the case base.

Keywords: process-oriented case-based reasoning, compositional adaptation, workflows

Resubmission of Müller, G., Bergmann, R.: Workflow Streams: A Means for Compositional Workflow Adaptation in Process-Oriented CBR. In Lamontagne, L., Plaza, E. (Eds.) Case-Based Reasoning Research and Development, 22th International Conference on Case-Based Reasoning, ICCBR 2014, Cork, Ireland, Proc. LNCS, vol. 8765, Springer (2014)

Acknowledgements. This work was funded by the German Research Foundation (DFG), project number BE 1373/3-1.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

Knowledge modelling and maintenance in myCBR3*

Alexander Hundt^{1,A}, Pascal Reuss^{1,B}, Christian Sauer^{2,C}, and Thomas Roth-Berghofer^{2,D}

^Aalexander.hundt@dfki.de ^Bpascal.reuss@dfki.de ^Cchristian.sauer@uwl.ac.uk

^Dthomas.roth-berghofer@uwl.ac.uk

¹DFKI German Research Center for Artificial Intelligence, Kaiserslautern, Germany and

²University of West London, London, UK

Abstract. One of the main aspects of knowledge management is the task of knowledge maintenance. Building and running knowledge intensive Case-Based Reasoning applications requires fundamental design decisions during the system design phase with regard to the knowledge maintenance within the system as well as accurate knowledge maintenance approaches within the running system. In this paper we will detail on the design decisions available in the *myCBR 3* CBR system design software as well as research in the available and future knowledge maintenance approaches within *myCBR 3* CBR. Next to maintaining the standard knowledge of any CBR system, represented by the four knowledge containers after Richter, this paper also presents existing and currently researched approaches to represent and furthermost maintain context knowledge as well as explanatory knowledge within *myCBR 3* CBR. We will give an overview of the *myCBR 3* CBRs Knowledge Engineering workbench, providing the tools for the modelling and maintenance process and detail on currently explored new features to further integrate knowledge maintenance for context-sensitive and explanation aware CBR systems into our *myCBR 3* CBR software.

1 Introduction

Case-Based Reasoning (CBR) is a methodology introduced by Riesbeck and Schank [9] and Kolodner [4]. CBR is mimicking the human approach of reusing past experience to solve new problems. The basic reasoning model of CBR, the so called CBR cycle, was introduced by Aamodt and Plaza [1]. The CBR cycle consists of four processes: *Retrieve*, *Reuse*, *Revise* and *Retain*. The episodes of experience that CBR reasons upon are stored in cases that consist of pairs of problem and solution descriptions. Problem descriptions are described by tuples of attribute value pairs that describe a problematic or critical situation. The corresponding solution description of a case consists of information how the problem described in the problem description was successfully solved. In the *Retrieve* phase of the CBR cycle the attribute value pairs describing a current problem encountered are matched against the problem descriptions in all cases within

* Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

the case base of the CBR system. The CBR system employs similarity measures to calculate the distances between singular attribute values and subsequently the similarity between the current problem description and the problem descriptions in the case base's cases. A selectable number of n best matching cases are then retrieved from the case base. In the *Reuse* step the retrieved cases solutions are then applied to the current problem in order to try and solve the problem at hand. this reuse of the past solution(s) may involve the adaptation of the solutions described in the retrieved past cases. After applying an adapted solution the final outcome of the solution, either being successful or partly successful or failing is revised in the *Revise* step of the CBR cycle. If the solution was successful the new case, consisting of the current problem description and the successfully applied solution, that may have been adapted, is retained in the CBR systems case base in CBR cycles final *Retain* step.

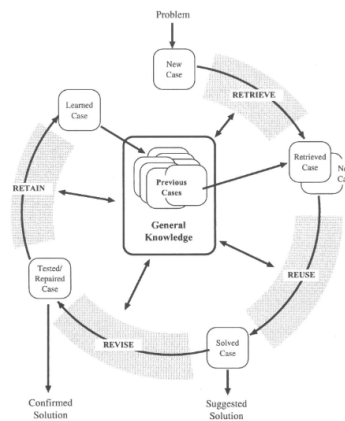


Fig. 1. The CBR cycle

The knowledge that is involved in the reasoning steps of the CBR cycle is represented in formal form within the four knowledge containers of CBR introduced by Richter [7].

- *Vocabulary* defining attributes and their allowed value ranges. This can be value ranges (min, max) for numeric attributes or a list of allowed symbols for symbolic attributes.
- *Similarity Measures* Functions to calculate the similarity between individual attribute values (local similarity measures) as well as the similarity between whole problem descriptions (global similarity measures). These functions are often distance functions for numeric values or comparative tables or taxonomies of symbols for symbolic attributes.
- *Adaptation Knowledge* Often represented by rules that can be applied to adapt the solution description of retrieved cases to match the current problem and enable it to be solved.

- *Cases* The descriptions of past episodes of experience, consisting of pairs of problem and solution descriptions. They employ tuples of attribute value pairs to describe the problematic situation and store a solution to the described problem within a solution description part.

Next to the aforementioned knowledge containers we consider two additional fields of knowledge important and incorporate these into our research: knowledge about context and knowledge about explanations. While these two additional fields do not represent further knowledge containers, they provide significant information that can be leveraged in order to make Case-Based Reasoning systems more comprehensible and also offer ways to improve maintenance capabilities.

In a gist we follow the definition of context as '*a descriptor (such as a word or an image) or set of descriptors that can represent a situation or a scenario.*' [15]. For further detail, context may be perceived as a situation which is depicted as a subset of descriptors that are part of a snapshot of the world at a given moment in time. This snapshot encompasses all existing objects in this world, their relationships and the states they are currently in. For another classification a real world situation, being a real subset of objects can be distinguished from an observed situation which represents the situation determined by the system.

If a system has the ability to classify situations, the knowledge about how to react in a determined situation and furthermore can distinguish between other similar situations, then this classification serves as the context the system is in [19].

We see the benefit of gaining knowledge about a system's context in obtaining a priori knowledge about a given situation without the need for gathering information with additional effort.

An explanation in its basic form may be an answer to a question. As such the knowledge about explanation enables the system to answer questions that are not directly related to a user's search query, yet improve the user experience by providing additional transparency and justification. Usually a user's question may involve certain trigger words like 'why' or 'how'. If put in a computing context the *general explanation scenario* consists of primarily three components. First the user, being the one who interacts with the system via a user interface (UI). Second the problem solver, being the actual software which executes functional tasks to comply with the users request. Finally the explainer itself, being the additional component required to trace the actions of the problem solver and present them via the UI towards the user [11].

As for the expected benefit we see the build up of confidence in a system valuable in terms of confidence in a query result as well as maintenance proposals, depending on how valuable the given explanations are deemed by the user. We primarily pursue the following five kinds and goals of explanations[12].

- Conceptual explanations fulfill the *learning* goal as they offer descriptive information about symptoms
- Why-explanations provide a *relevance* of an answer, thus explaining why the answer is a good answer
- How-explanations elucidate how the reasoner concluded the answer and therefore add *transparency* to the result

- Purpose-explanations present themselves as an explanation similar to conceptual explanations and might therefore be applicable for describing how concepts are related, thus providing *justification* for an explanation
- Cognitive explanations have an exceptional position as they aim at explaining non-physical attributes

The objective of this paper is to demonstrate existing and currently developed approaches to maintain the knowledge within the four knowledge containers as well as the additional explanatory and context knowledge that can be modelled within *myCBR 3*.

The rest of this paper is structured as follows: In section 2 we review related work on knowledge maintenance in CBR. We then introduce the process of knowledge modelling in *myCBR 3* in section 3. Based on the description of how to model CBR knowledge models in *myCBR 3*, we then introduce and discuss existing and currently developing approaches to enable *myCBR 3* to support knowledge maintenance 4. We do so with a focus on the four knowledge containers of CBR in the sections 4.1,4.2,4.3. Finally in section 5 we discuss the introduced approaches to knowledge modelling and maintenance and conclude.

2 Related Work

In the introduction we have already reviewed the four knowledge containers of CBR [8], for each of these containers there already exist a number of approaches to maintain the knowledge represented in the container [20]. Maintenance for the knowledge in the containers is necessary due to the fact that any change in the environment a CBR system operates in can affect the accuracy and competence of the CBR system's knowledge model [10]. Therefore the maintenance of CBR systems, particularly maintaining their knowledge models, is an important and on-going task.

Maintaining a knowledge model comprises tasks such as revising the knowledge within the model to cater for changes in the domain, add new knowledge to the model or remove knowledge that became deprecated. For example for the case base it is vital to control the case base's size as well as to detect inconsistent cases see for example the work of [16,10]. As a concrete example, a case base maintenance approach, introduced by Smyth and McKenna, is based on a performance model of a CBR system [17] which is used to identify less competent cases to delete them from the case base.

Another approach to case base maintenance was introduced by Leake and Wilson [5] highlighting the importance of conducting case base maintenance by balancing the competence-performance dichotomy of a CBR system. In addition Leake and Wilson suggest that case base maintenance should be guided by important constraints including size limits of case base such as long and short term performance goals in expected future problems.

Next to the best researched maintenance of the knowledge container case base, there exists a number of further research on the other three knowledge containers. Out of these the knowledge container similarity measures is the second best researched with regard to the maintenance of the knowledge in this container, see for example the work of [3]. Another approach to maintain the casebase and the similarity knowledge is from [6]. They improve the quality of similarity measures by enhancing the coverage of cases.

3 Knowledge modelling in myCBR

3.1 Case-Based Reasoning framework *myCBR 3*

Developing a CBR systems knowledge model requires a systematic approach to capture and formalise the domain knowledge into the four knowledge containers of CBR. Such a knowledge modelling task is often a process that involves a significant effort, it is therefore desirable to rely on modelling software for this crucial initial development process for a CBR system. *myCBR 3*¹ is such a modelling software. It is an open source tool targeting at developing knowledge models [18]. It is emphasizing the ability to rapidly prototype a cbr knowledge model, especially the contents of the vocabulary and similarity measure containers. Next to the modelling facilities *myCBR 3* also offers a similarity-based retrieval tool as well as a software development kit (SDK). *myCBR 3* offers a variety of GUIs within its Workbench that enables a knowledge engineer to model and test a knowledge model, especially sophisticated similarity measures. The knowledge model can further be tested within the *myCBR 3* Workbench, using the in-built retrieval test tool to refine and update the knowledge model which are both functions that play a key role within the task of knowledge maintenance.

Using the *myCBR 3* SDK allows for an easy integration of the knowledge model into a java-based application. The following code example shows a simple retrieval on a myCBR case base.

```
/*Initialize the retrieval engine*/
Retrieval ret = new Retrieval(concept, casebase);
SequentialRetrieval = new Retrieval(project, ret);

Instance query = ret.getQueryInstance();

/*Here the query has to be set*/
query.mapInputToAttributes();

/*The resulting List contains k cases sorted by similarity */
List<Pair<Instance, Similarity>> result =
    seqret.retrieveKSorted(casebase, query, k);

printResults();
```

3.2 Knowledge modelling with the myCBR Workbench

As mentioned earlier the *myCBR 3* Workbench provides powerful GUIs for modelling CBR knowledge models. A key focus of the Workbench is laid on the modelling of

¹ <http://www.mycbr-project.net>

knowledge-intensive similarity measures. Furthermore the Workbench provides task-oriented view-configurations for either modelling your knowledge model, perform information extraction or case base management. To enable the testing and refinement of developed knowledge models the Workbench offers a similarity-based retrieval functionality. The *myCBR 3* SDK employs a simple-to-use data model which facilitates the integration of the knowledge model into any java-based application. Both, the retrieval process as well as the case loading are fast and therefore allow for a seamless integration and use within applications built on top of a knowledge model developed with *myCBR 3*.

myCBR 3 allows for each attribute to have several similarity measures, which in turn allows for allows for experimenting with different similarity measures to record variations. Next to experimentation this feature can also be used to select an appropriate similarity measure at run-time via the API to accommodate for different contexts such as for example different types of users.

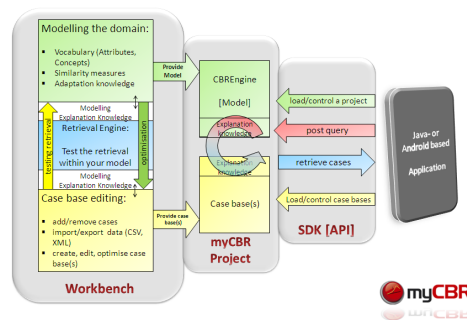


Fig. 2. Integration of the *myCBR 3* workbench and SDK in CBR project development

3.3 Modelling the Case Structure and Similarity Measures

The *myCBR 3* Workbench offers two different views to edit either the knowledge model or the case base(s). In this section we will shortly introduce the modelling view for the knowledge model as shown in 3. The concept of modelling a knowledge model in the Workbench follows the approach that initially a case structure is created. based on the initial case structure the vocabulary is then defined and the necessary individual local similarity measures for each attribute description (eg. CCM in 3) are then created, followed finally by the global similarity measure for a concept description (Car in 3).

The modelling view of the Workbench (see figure 3) is showing the case structure on the left side, available similarity measures for a selected attribute or concept beneath it and the definition of a similarity measure or attribute in the center. The modelling of similarity measures in the Workbench takes place on either the attribute level for local similarity measures or the concept level for global similarity measures.

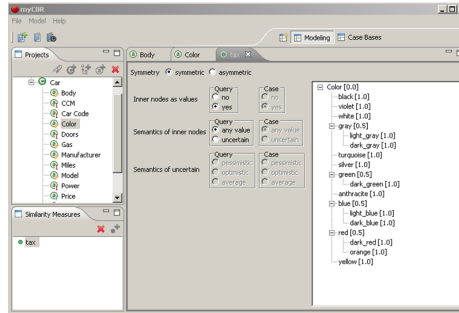


Fig. 3. Example of the knowledge model view in the *myCBR 3* Workbench

3.4 Building a Vocabulary

As mentioned earlier, the vocabulary within a *myCBR 3* knowledge model consists of concepts and attributes. A concept can consist of one or more attribute descriptions as well as attributes referencing different concepts. This representation allows the user to create object-oriented case representations. In the current version *myCBR 3* allows for the import of vocabulary items, e.g. concepts and attributes, from existing CSV files as well as from Linked (Open) Data (LOD) sources. A versatile feature for the case import is the re-construction of case structures when importing case data from CSV files. Within *myCBR 3* an attribute description can have one of the following data types: Double, Integer, Date and Symbol. For each of these data types *myCBR 3* provides similarity functions editors.

Next to the four knowledge containers of CBR, *myCBR 3* allows for the representation of explanatory knowledge, being knowledge used to create explanations of the systems reasoning and results. For example *myCBR 3* allows for providing canned explanations as well as references to online sources for concept explanations. *myCBR 3* further allows to represent context knowledge via the definition of a multiple of similarity measures for both, attributes as well as cases.

4 Knowledge maintenance in myCBR

Having modelled a knowledge model the next important task is to maintain the knowledge represented within the model. In this section we will introduce and review current approaches to knowledge maintenance being implemented at the time for the *myCBR 3* Workbench. We further introduce already published work on approaches to adaption knowledge modelling and the potential to use this approach for future adaption knowledge maintenance.

4.1 Case knowledge

A recent approach to integrate knowledge maintenance facilities for case bases into *myCBR 3* is described in [2]. The approach aimed to provide a maintenance perspective

in *myCBR 3* to assess data on case usage. The case usage data was then used to generate quality measures which were employed to trigger maintenance measures for the knowledge in the case base, vocabulary and similarity measures.

The maintenance perspective was implemented by extending *myCBR 3*, enabling to generate usage-data on the access of individual cases during retrieval. Based on experimentation a set of threshold values for quality measures were established which were used to monitor the top performing and most retrieved cases as well as the least performing and least retrieved cases. Monitoring the best as well as the least performing cases allowed for the analysis of well performing cases and the adjustment of less well performing cases. Next to simply deleting the least performing cases, the less performing cases can be adapted within the maintenance perspective. The approach also included the necessary features to reverse these changes, in case the performance deteriorates after the changes. The conceptual approach of automating the measurement of the quality measures and the subsequent triggering of the maintenance tasks can be seen as a control loop to manage the maintenance of the case base.

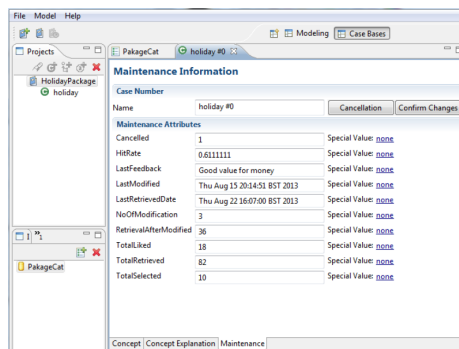


Fig. 4. The added maintenance view in *myCBR 3*

The described approach implemented the maintenance attributes temporarily by simulating these attributes in *myCBR 3*. the simulation of the attributes was achieved by labeling the maintenance attributes in a specific way. Figure 4 shows the implemented maintenance view with the maintenance attributes implemented for a holiday package recommender system.

4.2 Similarity knowledge

As stated earlier, *myCBR 3* allows for similarity based retrieval tests. These tests allow for the evaluation of the efficiency and accuracy of a knowledge model built with *myCBR 3*. The tests can be performed, of course, at the development state of the knowledge model but, more importantly in the context of this paper, also on a knowledge model that is already included in a life application. So, for example, we assume a product recommender system built on top of a *myCBR 3* knowledge model. The knowledge model

can be tested parallel within the Workbench environment, which allows for establishing the need for maintenance measures as well as to test the outcome of maintenance measures in a sort of “dry-dock” environment. If the similarity measure maintenance is finished within the *myCBR 3* workbench the updated knowledge model can be re-loaded seamless in the life application via the *myCBR 3* API. This functionality allows for the seamless incorporation of user feedback as well as for the continuous refinement of the similarity knowledge in the model and thus for maintaining the accuracy of the model. Currently the authors are investigating the approach to adapt the approach to acquire and use usage data and retrieval test data from the knowledge model, described in the previous subsection 4.1. The aim of this on-going work is to provide a similar maintenance view in *myCBR 3* to cater for similarity measure maintenance.

4.3 Adaptation knowledge

As mentioned in the sections above the tool *myCBR 3* supports the *retrieve* step of the CBR cycle. In the near future *myCBR 3* will also support the *reuse* step of the CBR cycle. This subsection describes the functionality that our tool will provide.

To support the *reuse* step *myCBR 3* is combined with the open source tool JBOSS Drools. Drools consists of five projects: Drools Guvnor, Drools Expert, jBPM5, Drools Fusion and OptaPlanner. For our purpose only Drools Expert, which contains the rule engine, is combined with *myCBR 3*. There are several reasons for choosing Drools for the adaptation:

- 100% JAVA, so it is easy to integrate in *myCBR 3*
- license compatible to *myCBR 3* license
- performance of the Rete algorithm
- scalability of the rule bases
- independent lifecycle

The use of Drools Expert allows us to define completion and adaptation rules and process them to enrich the query and adapt the retrieved cases. In our rule concept a rule belongs to a rule base and has five properties: type, case base, precondition, condition expression and conclusion expression. The first property defines the type of the rule, either completion rule or adaptation rule. The second property defines to which case based a rule is assigned. A rule can be assigned to several case bases at once. Completion rules are not assigned to a case base, because they are used to enrich a query. Adaptation rules has to be assigned to at least one case base. The precondition property allows to define a set of conditions, that is used to determine if a rule has to be checked for firing or not. This way the number of rules the Rete algorithm has to process can be significantly reduced and therefor the performance of the rule processing is increased. The condition and the conclusion properties are used to define the rule itself. The condition expression is a set of one or more single conditions which consists of an attribute of the case structure, an operator and a value. The single conditions are combined with logical operators AND or OR. The conclusion expression works the same way, but the logical operator is always AND.

For the implementation of the rule concept the *myCBR 3* SDK and the *myCBR* workbench are extended. The SDK is extended with several new classes to support the

rule properties mentioned above. The myCBR workbench is extended with a new view that contains a rule editor. Figure 5 shows the new rule editor.

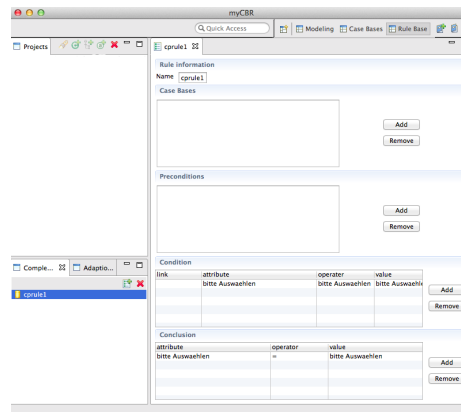


Fig. 5. myCBR rule editor

With the help of this editor new rule base can be created. A rule base can contain completion rules and adaptation rules. When a case base is added the editor presents a list of possible case base the rule can be applied to. This is the first action that has to be done, because based on the assigned case based, the condition and conclusion has different lists of attributes and values that can be chosen. To define the condition and conclusion of a rule the editor uses select lists. The list for the attributes contains all attributes from the assigned case bases and the value list contains all values that are allowed for the selected attribute. The operator list contains at the moment only \leq , \geq , $=$, \neq , but more operator will be implemented in the future. If more than one condition is defined a user can choose from the link list whether the conditions are link with AND, OR or XOR.

The next step after implementing the rule editor and the rule processing, is to define evaluation and maintenance strategies for the adaptation knowledge. A simple evaluation strategy may be to check the selected attributes and values against the defined vocabulary. This way inconsistency could be found after some terms have been removed from the vocabulary. Another evaluation strategy is to check the rules if there are conflicts among themselves. Maintaining the rules can be done with the help of the rule editor. A new maintenance view could be used to display the evaluation results to the knowledge engineer. The engineer then has to decide which maintenance actions must be done, either changing a rule or removing it.

5 Discussion and Conclusions

In this paper we have presented the *myCBR 3* Workbench and its use to model and maintain knowledge for CBR systems. We did so by reviewing the existing process of

modelling knowledge for the four knowledge containers of CBR and from there on elaborate on currently available knowledge maintenance approaches as well as approaches currently being developed within *myCBR 3*.

A topic for discussion can be seen in the question of the pay-off between the initial effort to implement the maintenance functionalities within the *myCBR 3* GUI, establish and create the maintenance measurement attributes and useful threshold values, compared to the actual benefits from the maintenance approaches. The authors conclude, based preliminary testing and feedback from knowledge engineers and non CBR-expert domain experts, that the effort of implementing the maintenance measures is well worth it. From a series of published research project [13] [14] it can definitely said that the pay off of the GUI-based new functionalities is high as these can be used even by non CBR experts to implement maintenance measures with their CBR knowledge models.

Based on the benefits that were gained from the implementation of knowledge maintenance for the existing 4 knowledge containers of CBR the next step in our work is the introduction of maintenance measures for explanatory and context knowledge, as these functionalities, explanation awareness and context-awareness, are themselves still in a prototype status within *myCBR 3*. However the authors assume the benefit from these functions as so high that their implementation in a future release of the software is highly likely. An additional point to argue for the integration of the maintenance measures for explanatory and context knowledge in *myCBR 3* lies in the fact that implementing these measures alongside the implementation of the explanation aware and context aware functions offers the opportunity to take the importance of the maintenance functions into account.

So the authors conclude that the approaches to knowledge maintenance within *myCBR 3* developed so far and currently under development are useful and desirable. This conclusion is based on experimental results as well as on feedback from domain experts working with prototypes of *myCBR 3*, published in a number of workshop and conference submissions. Furthermore it can be concluded that it is a rewarding task to develop these approaches as they reduce the pressure on the initial knowledge modelling with regard to the absolute need of formalising the knowledge 100 per-cent correct at the first (development) step, as the maintenance measures, along with the performance measuring functionalities, for example the case performance measuring, can easily be used to amend in reaction to a changing domain or to refine a probably not optimally designed initial knowledge model.

Additionally the highly modularised code structure of *myCBR 3* allows for easy expansion, adaption, so a lot more approaches to representing maintenance knowledge, maintaining knowledge and controlling the triggering of maintenance measures can be easily developed. Finally, the integration of the approaches presented in this paper is currently on-going and the maintenance functions presented will be part of the a new release version of *mycbr3.x* in the near future.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, 7(1), 39–59 (1994)

2. Aul, V., Sauer, C.S., Bulbul, E., Wilson, D.C., Roth-Berghofer, T.: Knowledge maintenance in mycbr. In: Proceedings of the 18th UKCBR 2013 Workshop. Springer (2013)
3. Bergmann, R., Stahl, A.: Similarity measures for object-oriented case representations. Springer (1998)
4. Kolodner, J.L.: Case-Based Reasoning. Morgan Kaufmann Publishers, Inc. San Mateo (1993)
5. Leake, D.B., Wilson, D.C.: Categorizing Case-Base Maintenance: Dimensions and Directions. In: Smyth, B., Cunningham, P. (eds.) Advances in Case-Based Reasoning, Proceedings of the 4th European Workshop on Case-Based Reasoning, EWCBRY98, Dublin, Ireland. pp. 196–207. Springer-Verlag, Berlin (1998)
6. Patterson, D., Anand, S.S., Hughes, J.: A knowledge light approach to similarity maintenance for improving case-base competence. In: European Conference on Artificial Intelligence Workshop Notes. pp. 65–78 (2000)
7. Richter, M.M.: Introduction. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S. (eds.) Case-Based Reasoning Technology – From Foundations to Applications. LNAI 1400, Springer-Verlag, Berlin (1998)
8. Richter, M.M.: Introduction. chapter 1 in case-based reasoning technology - from foundations to applications. Inai 1400, springer (1998)
9. Riesbeck, C.K., Schank, R.C.: Inside case-based reasoning. Lawrence Erlbaum Associates, Pubs., Hillsdale, N.J. (1989)
10. Roth-Berghofer, T.: Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology, Dissertation. Ph.D. thesis, Universität Kaiserslautern (2003)
11. Roth-Berghofer, T., Sauer, C.S., Althoff, K.D., Bach, K., Newo, R.: Seasaltexp - an explanation-aware architecture for extracting and case-based processing of experiences from internet communities. In: Proceedings of the LWA 2011 - Learning, Knowledge, Adaptation (2011)
12. Roth-Berghofer, T.R.: Explanations and case-based reasoning: Foundational issues. In: Funk, P., Calero, P.A.G. (eds.) Advances in Case-Based Reasoning. pp. 389–403. Springer-Verlag (September 2004)
13. Sauer, C.S., Hundt, A., Roth-Berghofer, T.: Explanation-aware design of mobile mycbr-based applications. In: Case-Based Reasoning Research and Development, pp. 399–413. Springer (2012)
14. Sauer, C.S., Roth-Berghofer, T., Aurricchio, N., Proctor, S.: Similarity knowledge formalisation for audio engineering. In: Proceedings of the 17th UKCBR 2012 Workshop. Springer (2012)
15. Segev, A.: Identifying the multiple contexts of a situation. Modeling and Retrieval of Context pp. 118–133 (2006)
16. Smyth, B.: Case-base maintenance. In: Tasks and Methods in Applied Artificial Intelligence, pp. 507–516. Springer (1998)
17. Smyth, B., McKenna, E.: Building compact competent case-bases. In: Althoff, K.D., Bergmann, R., Branting, L.K. (eds.) Case-Based Reasoning Research and Development: Proceedings of the Third International Conference on Case-Based Reasoning, ICCBR'99, Seon Monastery, Germany. pp. 329–342. Springer-Verlag, Berlin (1999)
18. Stahl, A., Roth-Berghofer, T.R.: Rapid prototyping of CBR applications with the open source tool myCBR. In: Proceedings of the 9th European conference on Advances in Case-Based Reasoning. pp. 615–629. Springer-Verlag, Heidelberg (2008)
19. Turner, R.: A model of explicit context representation and use for intelligent agents. Modeling and Using Context pp. 831–831 (1999)
20. Wilson, D.C., Leake, D.B.: Maintaining case-based reasoners: Dimensions and directions. Computational Intelligence 17(2), 196–213 (2001)

Is learning-by-doing via E-learning helpful to gain generic process knowledge?

Michael Leyer¹, Minhong Wang², Jürgen Moormann¹

¹ Frankfurt School of Finance & Management, Frankfurt, Germany
{m.leyer;j.moormann}@fs.de

² Hong Kong University, HongKong, China
magwang@hku.hk

Abstract. Learning generic process knowledge is important to transform organizations from a function- to process-orientation to gain efficiency benefits. It requires a fundamental change of mind by employees as the required knowledge of process-oriented and function-oriented organizations differs substantially. However, a shift of mind is hard to achieve for employees as processes remain abstract or intangible. Empirical results on the learning method are rare, only showing that learning-by-doing is superior. In addition, e-learning is supposed to be promising to be applied, but due to the context dependency leaving the question open how learning-by-doing helps in the given context. Concluding, the hypothesis is that learning-by-doing in an e-learning setting leads to a significant increase of generic process knowledge.

We set up an e-learning program containing tasks based on a learning-by-doing approach. Generic process knowledge is operationalised with the following dimensions: Customer, goals, teams, hierarchy, management, continuous improvement and process design.

The e-learning phase was integrated with a pre-test-post-test design in an academic course on Management (N=80). The results reveal that learning-by-doing via e-learning leads to a significant learning effect of almost 20 per cent. Thus, the hypothesis can be confirmed ($T(79) = -5.709, p < .001$). Overall, the results can be considered as strong taking into account the relatively short time participants spent, the low number of training repetitions and a limited forum exchange. However, the level of 59.6% still leaves some room for improvement such as more explanation or exchange between participants.

Keywords: process knowledge, e-learning, learning-by-doing

Resubmission of Leyer, M./Wang, M./Moormann, J. (2014), Is learning-by-doing via E-learning helpful to gain generic process knowledge?, in: Sampson, D.G., Spector, J.M., Chen, N.-S., Huang, R., Kinshuk, K. (Hrsg.), Proceedings of the 14th IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society, Piscataway, NJ, S. 711-713.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

TELESUP

Textual Self-Learning Support Systems

Sebastian Furth¹ and Joachim Baumeister^{1,2}

¹ denkbares GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg, Germany

² University of Würzburg, Institute of Computer Science,
Am Hubland, 97074 Würzburg, Germany
{firstname.lastname}@denkbares.com

Abstract. The regular improvement and adaptation of an ontology is a key factor for the success of an ontology-based system. In this paper, we report on an ongoing project that aims for a methodology and tool for ontology development in a self-improving manner. The approach makes heavy use of methods known in natural language processing and information extraction.

1 Introduction

Today, intelligent systems successfully provide support in many complex (production) processes. Typical application areas of such systems are processes in mechanical engineering and in the medical domain. The core of an intelligent system is the knowledge base, that monitors the requirements and derives support actions.

The development of the knowledge base is usually complex and time-consuming, since complex correlations need to be considered for the derivation knowledge. In domains with frequent changes of the knowledge, for instance, new experiences in processes, it is necessary to frequently modify/adapt the knowledge base. This continuous improvement/adaptation of the knowledge base is a key factor for the long-term success of the system. As the original creation of the knowledge the continuous adaptation is also a complex and time-consuming task. The goal of the presented project is the implementation of a development tool for support systems that includes self-learning capabilities to regularly adapt the included knowledge base. In the general context of the project SELESUP (Self-Learning Support Systems) various types of sources for learning can be connected. The project SELESUP comprises the sub-projects STRUSUP (Structural Self-Learning Support Systems) and TELESUP (Textual Self-Learning Support Systems) that exploit structured and textual data respectively.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

In the TELESUP sub-project we

1. define an ontology as the primary knowledge representation for the knowledge base, and
2. use unstructured data, especially text, as the primary resource for the learning method.

The use of such a tool allows for a significant increase of efficiency concerning the development and maintenance of intelligent support-systems.

2 The TELESUP Process

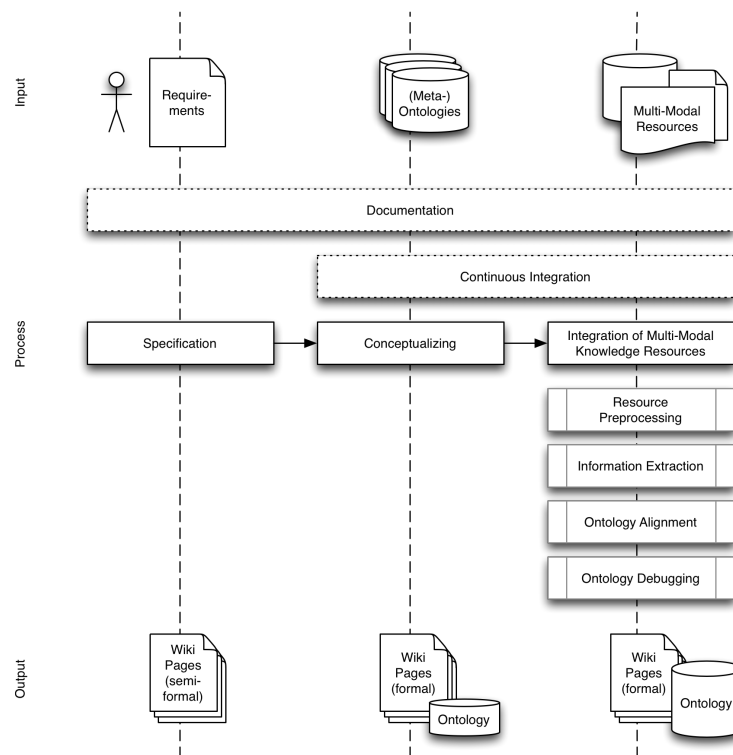


Fig. 1. The input, output and steps of the TELESUP process.

2.1 Problem Description (Distinction)

The TELESUP process, depicted in Figure 1, aims to support the effective development of ontologies for support systems. The application scenario for the

developed ontologies is usually located in the context of technical support systems. In this domain, ontologies are often comprehensive and complex. Therefore we consider the development of the core ontology structure as a manual task that requires intensive coordination between multiple domain experts. In order to ensure scalability we usually follow the model “less semantics, more data”, i.e., we focus on the integration of data instead of the usage of heavy semantics. Technical support systems usually need to consider a lot of domain specific multi-modal resources. The pool of these knowledge resources usually increases constantly over time, e.g., due to the introduction of new machines. As the information contained in these resources should also be represented in the ontology, the population must be considered highly volatile.

The scenario described above leads to a couple of requirements constituting an ontology engineering process for textual self-learning support systems. As multiple domain experts and ontology engineers are involved in the development of the core structure of the ontology the process should support collaboration. The integration of vast amounts of multi-modal knowledge resources is usually a challenging, time- and cost-intensive task during the development of an ontology. Therefore we propose the (semi-)automatic population of the ontology by exploiting these resources with methods adapted from the field of Information Extraction and/or the broader field of Natural Language Processing. The preservation of the ontology’s consistency is a major challenge when incorporating (semi-)automatic ontology population approaches in the ontology engineering process. Additionally the multi-modal resources should be considered as valuable sources for ontology refinement suggestions.

2.2 Specification of the Ontology Structure

The first phase of the ontology engineering process considers the collaborative specification of the ontology’s core structure, i.e. identifying required classes and relations between them. Requirements, scope, and the level of formalization is specified in a semi-structured way. In addition to the core structure of the ontology, tests for the validation and verification are specified. The performance/technical specification known from classical software engineering is an appropriate analogy.

2.3 Conceptualizing the Specification

Baumeister et al. [3] proposed the Knowledge Formalization Continuum, i.e. informal knowledge gets subsequently refined into explicit knowledge. Following this idea the specification of the ontology’s core structure is formalized in this phase. The continuum allows for the stepwise conceptualization of the ontology specification, e.g. by first collecting relevant terms for a domain that subsequently get formalized to concepts, which then are described further by using domain specific properties. The formalization follows the level defined in the specification document and uses a standardized ontology language like RDF(S) [21] or OWL [12]. Additionally the phase allows for the integration of

(meta-)ontologies, e.g. SKOS [19] or specific upper ontologies for the application scenario. The conceptualization also covers the test specification, i.e., concrete test cases need to be formulated that are able to validate and verify the ontology. There exist various ontology evaluation methods that can be utilized, e.g. data-driven [5] or task-based [15] ontology evaluation. We consider the conceptualization of a complex ontology specification a rather manual task that needs a lot of coordination between ontology engineers and domain experts. Ontology Learning [6] techniques might facilitate the conceptualization by providing suggestions that can be used as basis for discussions between the experts.

2.4 Integration of Multi-Modal Knowledge Resources

Multi-Modal Knowledge Resources Knowledge usually exists in a variety of forms, ranging from highly structured documents (e.g. XML) to completely unstructured resources (e.g. scanned texts, images, videos etc.). We call these documents multi-modal knowledge resources. In technical support systems relevant examples are all forms of technical documentation, e.g. handbooks, repair manuals, service plans or schematics. Additionally documents created for the production process contain valuable information, e.g. a bill of material can be exploited to suggest a component hierarchy of a product.

Resource Preprocessing The various kinds of multi-modal knowledge resources usually need to be preprocessed to improve accessibility for the subsequent information extraction tasks, e.g., when confronted with PDF documents. In general the goal of this phase is to incrementally add structure to previously unstructured documents. Despite the conversion of the file format (e.g. PDF to XML) typical preprocessing tasks from the field of Natural Language Processing are applied, i.e. segmentation, tokenization, part-of-speech tagging, and the detection of structural elements (e.g. tables, lists, headlines). Another important topic in this phase is data cleaning, i.e., preprocess the data in a way that the results are free from noisy data that might affect the information extraction results.

Extracting Relevant Information One of the main challenges during the integration of multi-modal domain knowledge is the extraction of the relevant information from the different sources. After the resources have been preprocessed they are accessible for information extraction methods, e.g., extraction rules that are typically used in rule-based Information Extraction. In general extraction rules can either be formulated by domain experts or automatically learned using Machine Learning algorithms, e.g. LP2 [7], WHISK [17] or TraBaL [9]. The process presented here allows both the manual formulation as well as the (semi-)automatic learning of rules. For the latter one, terminology created during the specification and/or conceptualization phase might be exploited, i.e., used to annotate documents that then serve as training data for the Machine Learning algorithms. The extraction rules are mainly used to extract candidates

for the population of the core ontology structure. Additional information that could potentially serve for refinements of the ontology structure might be considered.

Ontology Alignment An important question when handling the extracted candidates for ontology population is whether they are really new concepts or just synonyms for existing concepts. There exist a variety of metrics that can be utilized to measure the similarity between concepts. Besides the well-established string similarity metrics, e.g., Levenshtein distance [13], more elaborated methods exist that use statistics or even consider the semantic relatedness. A combination of several methods can also be used in an ensemble method, as proposed by Curran [8].

Ontology Debugging When using automatic information extraction methods on large sets of resources usually a huge amount of candidates is generated. A major challenge when automatically deploying these candidates to the existing ontology is keeping the ontology in a consistent state. We define a consistent ontology as an ontology that is not only valid in terms of special semantics (e.g. OWL's consistency check) but also pass predefined test cases representing knowledge about the domain (e.g. the tests specified and conceptualized in the preceding phases). When deploying a set of candidates leads to an inconsistent ontology, then abandoning the complete change set is as unrealistic as tracing down the failure cause manually. Consequently, a method for isolating the fault automatically is necessary. We propose the usage of an ontology debugging mechanism, that is able to find the failure-inducing parts in a change set. The faulty parts should be isolated and manually reviewed by a domain expert and/or ontology engineer.

2.5 Continuous Integration

In addition to the debugging mechanisms applied during the integration of multimodal knowledge resources we propose the use of continuous integration (CI) for the development of knowledge systems, enabling the application of automated tests to avoid breaking an ontology. While the main purpose of the ontology debugging mechanism is tracing down the failure-inducing parts in a large change, CI ensures that the ontology is always in a consistent state. Again the test cases formulated on basis of the test specification can be used in CI.

2.6 Documentation

As in Software Engineering the documentation of the developed ontology is a critical success factor as it is the basis for the deployment of the final ontology. When following the phases described so far one can yield not only an ontology but also huge parts of the documentation. Starting with the specification of the ontology the described phases propose the continuous formalization of the

ontology. As this specification is the basis for the development of the ontology's structure it can also serve as a basis for the documentation. Additionally we proposed exploiting multi-modal knowledge resource for the population of the ontology. As the employed information extraction techniques are usually able to hold references to the relevant text occurrences huge parts of the ontology population can be documented by providing links to the original text source. The tool used for the development of the ontology should provide an export feature for the documentation in order to ensure convenient distribution/delivery.

3 Tool Support

3.1 KnowWE

Most of the steps in the ontology engineering methodology described above require tool support. We envision an integrated tool that supports the entire process. KnowWE [4] is a semantic wiki that has recently encountered a significant extension of its ontology engineering capabilities. Besides the ontology engineering features KnowWE also offers an elaborated plugin mechanism that allows for the convenient extension of KnowWE. Thus KnowWE provides a reasonable platform for the implementation of the tool support.

3.2 Ontology Engineering

KnowWE provides the possibility to define and maintain ontologies together with strong problem-solving knowledge. As outlined in the following it provides the typical features of an ontology management component [6]. Ontologies can be formulated using the RDF(S) or OWL languages. KnowWE provides different markups for including RDF(S) and OWL: proprietary markups and standardized turtle syntax [20]. In addition, KnowWE already offers possibilities to import (meta-)ontologies, e.g., SKOS. The ontologies are attached to wiki pages, referenced in a special import markup, and can then be used for the development. Besides these ontology management and editing features KnowWE offers a variety of ontology browsing and explanation features. For each concept an info page gives information about the usage of the concept in focus, e.g. which statements or SPARQL queries reference the concept. Additionally arbitrary SPARQL queries can be formulated and even visualized. Besides this a variety of other ontology visualizations are available which are usually used to support the manual ontology engineering, e.g., to explain the existing structure. The ontology engineering process is already supported by the use of continuous integration (CI) as described in [2], enabling the application of automated tests to detect the regression of an ontology.

3.3 Ontology Population

KnowWE already provides possibilities for the basic knowledge engineering, management, and browsing, it thus covers the tool support necessary to specify

and conceptualize the ontology structure. However, it lacks support for automatically populating an ontology by exploiting multi-modal knowledge resources. As described above for the exploitation of these resources, the access to preprocessing and information extraction algorithms is necessary. In the TELESUP project we extend KnowWE with connectors to preprocessing and information extraction algorithms. These connectors allow the configuration and the execution of the specific algorithms and provide potential candidates for the population of the ontology. In order to provide a convenient processing of the resources it will be possible to define pipelines of preprocessing and information extraction algorithms. For each pipeline the resources they shall process will be selectable.

3.4 Ontology Evaluation and Debugging

As described before KnowWE already offers different features for evaluating and debugging an ontology. The continuous integration extension of KnowWE allows to test an ontology continuously against specified test cases. Currently these test cases are mostly based on explicitly defining the expected results of SPARQL queries. We already proposed the usage of these test cases in order to find failure-inducing statements in a change set [11]. Therefore we developed a debugging plugin (see Figure 2) that is based on the Delta Debugging idea for software development proposed by Zeller [22]. Within the scope of the TELESUP project we will extend KnowWE's evaluation and debugging features in order to allow for constraint-based and/or task-based evaluation and debugging. For the latter we will also improve KnowWE's revision handling of formal knowledge, e.g. by introducing a time-machine plugin for different knowledge representations that allows the access of specific snapshots of an ontology.



Fig. 2. KnowWE's delta debugger presenting a failure-inducing change.

4 Related Work

We presented an ontology engineering methodology that proposes to (1) manually specify and conceptualize the core ontology structure, (2) semi-automatically populates the ontology by exploiting multi-modal knowledge resources and (3) strongly emphasizes the quality management. The idea of guiding the ontology

engineering process with a methodology is not new. The presented methodology is loosely related to METHONTOLOGY [10]. METHONTOLOGY is a methodology that starts with a formal specification of the ontology, and then acquires and conceptualizes relevant knowledge. It concludes with explicit implementation, evaluation and documentation steps and also allows for the integration of (meta-)ontologies. The major difference to the presented methodology is that TELESUP is able to analyze multi-modal knowledge resources and then automatically populates the ontology. Additionally TELESUP provides more sophisticated evaluation and debugging approaches. Pinto et al. [14] proposed DILIGENT, a methodology that strongly emphasizes the coordination in a distributed ontology engineering process. The methodology proposed by [18] is a five step approach that starts with a feasibility study, specifies the requirements in a kickoff and then continuously refines, evaluates and evolves the ontology. While the continuous refinement and evaluation of the ontology is comparable to TELESUP, we do not focus on the continuous evolution of the ontology, as we consider the ontology structure to be rather static.

There is a lot of related work regarding the automatic population of ontologies using information extraction technologies. As the identification and selection of appropriate information extraction techniques is subject of the TELESUP project and we focus on the underlying methodology in this paper we do not give a detailed description of related work in this field, but the BioOntoVerb proposed by Ruiz-Martinez et al. [16] is an example for a framework that transforms un-structured, semi-structured and structured data (i.e. multi-modal knowledge resources) to instance data.

Parts of the presented methodology can also be considered related to approaches known from Case-Based Reasoning (CBR) [1], e.g. the specification of the ontology structure and its subsequent conceptualization correspond to the definition of a vocabulary and similarity measures in CBR, while populating the ontology is similar to creating cases.

5 Conclusion

We proposed an ontology engineering methodology that is based on the Knowledge Formalization Continuum and incorporates information extraction techniques for the automatic population of an ontology structure by exploiting multi-modal knowledge resources. The underlying process emphasizes the quality management using Continuous Integration and the ability to trace down failure-inducing changes automatically. We presented the actual state of KnowWE and its already available ontology engineering abilities. In order to ensure proper tool support for the proposed methodology, we have also outlined the extensions to KnowWE that will be implemented as part of the TELESUP project. Besides the actual implementation of the presented extensions to KnowWE an extensive case study will be the main subject of our future work. The goal of the case study will be to evaluate whether the methodology and the tool support can sig-

nificantly increase the efficiency concerning the development and maintenance of intelligent support-systems.

Acknowledgments

The work described in this paper is supported by the Bundesministerium für Wirtschaft und Energie (BMWi) under the grant ZIM KF2959902BZ4 "SELE-SUP – SELF-LEARNING SUPport Systems".

References

1. Althoff, K.D.: Case-based reasoning. Handbook on Software Engineering and Knowledge Engineering 1, 549–587 (2001)
2. Baumeister, J., Reutelshoefer, J.: Developing knowledge systems with continuous integration. In: Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies. p. 33. ACM (2011)
3. Baumeister, J., Reutelshoefer, J., Puppe, F.: Engineering Intelligent Systems on the Knowledge Formalization Continuum. International Journal of Applied Mathematics and Computer Science (AMCS) 21(1) (2011), <http://ki.informatik.uni-wuerzburg.de/papers/baumeister/2011/2011-Baumeister-KFC-AMCS.pdf>
4. Baumeister, J., Reutelshoefer, J., Puppe, F.: KnowWE: a Semantic Wiki for knowledge engineering. Applied Intelligence 35(3), 323–344 (2011)
5. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data driven ontology evaluation (2004)
6. Cimiano, P., Mädche, A., Staab, S., Völker, J.: Ontology learning. In: Handbook on ontologies, pp. 245–267. Springer (2009)
7. Ciravegna, F.: $(LP)^2$: Rule Induction for Information Extraction Using Linguistic Constraints (2003)
8. Curran, J.R.: Ensemble methods for automatic thesaurus extraction. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. pp. 222–229. Association for Computational Linguistics (2002)
9. Eckstein, B., Kluegl, P., Puppe, F.: Towards learning error-driven transformations for information extraction (2011)
10. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering (1997)
11. Furth, S., Baumeister, J.: An Ontology Debugger for the Semantic Wiki KnowWE. In: under review (2014)
12. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer. W3C Recommendation (27 October 2009), available at <http://www.w3.org/TR/owl2-primer/>
13. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics-Doklady 10(8), 707–710 (1966)
14. Pinto, H.S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolvInG. In: Ecai 2004: Proceedings of the 16th European Conference on Artificial Intelligence. vol. 110, p. 393. IOS Press (2004)

15. Porzel, R., Malaka, R.: A task-based approach for ontology evaluation. In: ECAI Workshop on Ontology Learning and Population, Valencia, Spain. Citeseer (2004)
16. Ruiz-Martinez, J.M., Valencia-Garcia, R., Martinez-Bejar, R.: BioOntoVerb framework: integrating top level ontologies and semantic roles to populate biomedical ontologies. In: Natural Language Processing and Information Systems, pp. 282–285. Springer (2011)
17. Soderland, S.: Learning information extraction rules for semi-structured and free text. *Machine learning* 34(1-3), 233–272 (1999)
18. Sure, Y., Staab, S., Studer, R.: Ontology engineering methodology. In: Handbook on ontologies, pp. 135–152. Springer (2009)
19. W3C: SKOS Simple Knowledge Organization System Reference – W3C Recommendation: <http://www.w3.org/TR/skos-reference> (August 2009)
20. W3C: RDF 1.1 Turtle – W3C Recommendation. <http://www.w3.org/TR/turtle/> (February 2014)
21. W3C: RDF Schema 1.1 – W3C Recommendation. <http://www.w3.org/TR/rdf-schema/> (February 2014)
22. Zeller, A.: Yesterday, my program worked. Today, it does not. Why? In: Software EngineeringESEC/FSE99. pp. 253–267. Springer (1999)

The Connectivity of Multi-Modal Knowledge Bases^{*}

Joachim Baumeister^{1,2} and Jochen Reutelshoefer¹

¹ denkbares GmbH, Friedrich-Bergius-Ring 15, 97076 Würzburg

² University of Würzburg, Am Hubland, 97074 Würzburg

Abstract. Today, large knowledge bases are developed collaboratively and in an incremental manner. Often the engineering starts with the collection and organization of informal elements, that are subsequently refined into explicit knowledge. Due to the size of knowledge bases and the collaborative setting, the analysis of the current development progress becomes an important issue. The results of that analysis usually steer the further development direction and efforts.

In this paper, we introduce a graph-based representation of general knowledge bases, containing formal and informal knowledge. We use this representation to define general and tailored connectivity measures for knowledge bases. We briefly report on the application of these measures in an industrial case study.

1 Introduction

Despite significant progress, the development of large knowledge systems is a challenging task. One of the most pressing problems is the so-called *knowledge acquisition bottleneck* stating that the success of a system mainly depends on the successful acquisition/maintenance of knowledge [13]. The bottleneck describes the following problem areas: The high development costs of knowledge acquisition and the sustainable maintenance of knowledge. Process models have been introduced to weaken the problems of the knowledge acquisition bottleneck. Furthermore, state-of-the-art knowledge acquisition tools have introduced many advances such as support for collaboration and intuitive user interfaces to minimize the efforts of knowledge acquisition, e.g., see examples in [1, 8].

Recently, the understanding of *knowledge* in a system was defined in a broader sense by the introduction of the *knowledge formalization continuum* [2]. In general, the knowledge formalization continuum is a conceptual metaphor emphasizing that the entities of a knowledge base can have different facets ranging from very informal representations (such as text and images) to very explicit representations (such as logical formulae), see Figure 1. All facets of knowledge

^{*} Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

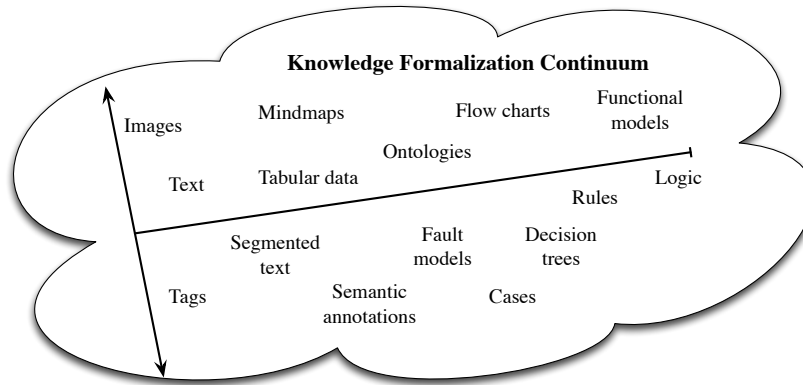


Fig. 1. The knowledge formalization continuum.

are considered as first class citizens. Thus, it is not necessary to commit to a specific knowledge facet at the beginning of a development project. Rather, it supports concentrating on the knowledge actually existing, by providing a flexible understanding of the knowledge formalization process. It is important to note that the knowledge formalization continuum is neither a physical model nor a methodology for developing knowledge bases. The concept should help domain specialists to consider even plain data, such as text and multimedia, as helpful knowledge that can be transformed incrementally to more formal representations when required. Data given by textual documents denote one of the lowest instances of formalization, represented on the left side of Figure 1. Functional models in contrast store knowledge at a very formal level, located on the right side. The term *formality* cannot be precisely defined in a general manner. In the context of our work, formal knowledge can be interpreted automatically by an inference machine. Whereas this is not possible for images today, it might be possible in a decade. A discussion of the notion of formality and the problem of its clear and useful definition (with a focus on mathematics) is also given in [4].

The formalization of knowledge within the knowledge formalization continuum was defined as an incremental process, where knowledge is initially provided as informal chunks of documents. In iterated phases the documents are then refined into an explicit formalization that is computer-interpretable. That way, explicit resources such as input concepts, outputs, decisions, and rules are connected with the corresponding documents. Also, documents itself are connected with other documents or already existing concepts. Such a process is described for instance in [6]. Incremental knowledge formalization has some advantages:

- It is possible to fill the entire knowledge into the system very early, at least in an informal manner.

- Some areas of the knowledge are only transferred to a formalized version when beneficial. In large projects it is often reasonable to leave some parts of the knowledge base in an informal manner, see [3] for a detailed discussion.
- Informal parts of the knowledge can be used as documentation/support of the formalized counter-part.

The incremental formalization process, however, requires the regular analysis of the formalization status in order to answer the following questions:

1. How is the knowledge base generally connected by formal concepts?
2. Which parts of the knowledge base have a formalized version?
3. Which parts of the knowledge base are candidates for the next formalization increment?
4. Which formal parts of the knowledge base need further improvement?

In this paper, we propose an approach to continuously determine the connectivity of the formalization. The connectivity and especially its visualization helps to interactively answer the questions stated above. The presented approach is abstract and reusable in a way, that it can be applied to a large variety of formalization approaches, since it builds on standardized semantic technologies. In the past, the approach was applied on (scoring) rule bases, OWL ontologies, and workflow knowledge bases.

The rest of the paper is structured as follows: Section 2 introduces a graph-based notion of multi-modal knowledge bases and shows how incremental formalization is represented. The subsequent Section 3 explains the use of semantic technologies to implement the approach in a systematic manner. In Section 4 a case study is briefly described, followed by a conclusion in Section 5.

2 Connectivity Measures

2.1 Multi-Modal Knowledge

Incremental knowledge formalization is implemented on a knowledge base. In the context of our work, we define a knowledge base as an abstract graph structure.

Definition 1 (Knowledge Base as Named Graph). *Let \mathcal{R} be a universal set of resources and \mathcal{P} a finite set of predefined properties. A knowledge base then is a subset of all possible knowledge tuples, i.e. edges:*

$$K_{(\mathcal{R},\mathcal{P})} \subset \mathcal{R} \times \mathcal{P} \times \mathcal{R}$$

Please note, that the graph spanned by $K_{(\mathcal{R},\mathcal{P})}$ is not necessarily *connected*, i.e., some resources $r_i \in \mathcal{R}$ can be isolated, i.e., r_i has no property $p_j \in \mathcal{P}$ connecting it to another resource.

Definition 2 (Multi-Modal Knowledge Base). *Let \mathcal{R} be a universal set of resources. In a multi-modal knowledge base a type from a finite set \mathcal{L} of types is assigned to each resource. Further, the minimal set of properties is defined as $\mathcal{P} = \{\text{serves}, \text{refines}\}$.*

In a multi-modal knowledge base each type from the type set denotes that a resource represents a kind of knowledge resource from the knowledge formalization continuum as discussed in Section 1. For instance, $\mathcal{L} = \{M, T, D\}$ could define a type set where D represents an output value of a knowledge system, T a text paragraph, and M a multimedia object, respectively. The resources are connected by properties, for instance a *serves* property states that one resource serves as a justification for another resource. Please note, that a property can not only connect resources but also properties, e.g., the *refines* properties usually states that one property instance is refined by another property instance.

Example 1 We introduce $\{M, T, D\} \subseteq \mathcal{L}$ to be a set of types, the set of resources $R = \{\langle T \rangle r_1, \langle D \rangle r_2, \langle M \rangle r_3\} \subseteq \mathcal{R}$, and $P = \{serves, refines\} \subseteq \mathcal{P}$. It defines that r_1 is a text paragraph, r_2 is a decision output, and r_3 is a multimedia object, such as an image for instance. The knowledge base $K_{(R,P)}$ defines the following connections:

$$K_{(R,P)} = \{serves(r_3, r_1), serves(r_1, r_2)\}$$

The property $serves(r_1, r_2)$ defines the semantic relation that the first resource r_1 fulfills a supporting/serving function for the second resource r_2 . The described resources and properties are depicted in Figure 1.

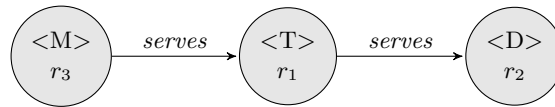


Fig. 2. A simple knowledge base $K_{(R,P)}$.

Incremental knowledge formalization represents the process of iterative extensions of a knowledge base $K \rightarrow K'$, where previously informal parts of the knowledge base K are extended by formal definitions and included in K' .

Example 2 We refer to Example 1 as the original knowledge base K . Let $R' = R \cup \{r_4, F\} \subseteq \mathcal{R}$ a set of resources and a set of properties $P' = P \cup \{refines\} \subseteq \mathcal{P}$. The type F stands for a class of formal knowledge, e.g., a rule. Then, the incremental extension $K'(R', P')$ adds a new formal input concept r_4 with the type F and the edge $serves(r_4, r_2)$, that refines the original edge $serves(r_1, r_2)$. For instance, r_4 is a rule deriving the resource r_2 . The refinement relation is represented by the additional edge $refines(serves(r_4, r_2), serves(r_1, r_2))$. The incremental extension K' of the knowledge base K is depicted in Figure 2.

For knowledge based applications we distinguish two different kinds of resources (not necessarily disjoint): Resources that will be in the focus of our analysis (target resources) and resources that support the derivation of those resources (serving resources).

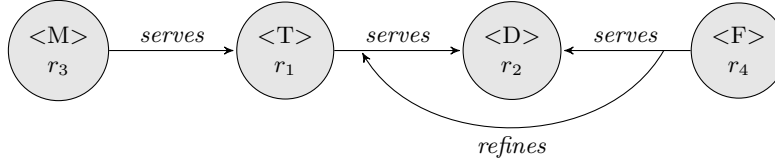


Fig. 3. The incremental extension K' of the knowledge base K .

Definition 3 (Target Output Resources and Serving Resources). Let \mathcal{R} be the universal set of resources and \mathcal{P} the universal set of properties. For a knowledge base $K_{(R,P)}$ with $R \subseteq \mathcal{R}$ and $P \subseteq \mathcal{P}$ we introduce two types of special resources: We call the subset $O \subseteq R$ the target output resources.

Further, the set $S \subseteq R$ of resources which are source nodes of a serves relation are defined as the serving resources:

$$S = \{ r \mid \exists \text{ serves}(r, x) \in KB, x \in R \}$$

Target output resources are used as possible outputs of the system, whereas serving resources support the derivation of target resources. Please note, that in larger settings also target resources can serve for the derivation of other (often more specialized) target resources. It is important to notice, that both sets—target resources and serving resources—usually grow during the knowledge formalization. For example, the refinement of one target resource can yield three more specialized target resources.

2.2 Connectivity Measures for Multi-Modal Knowledge

In the context of this paper we are interested in the connectivity of *target resources*, i.e., the use of knowledge that serves the derivation of these resources.

Simple Connectivity We define a very simple connectivity measure for general knowledge bases. Here, the connectivity of formal resources together with informal ones is calculated.

Definition 4 (Direct Connectivity). Let $K_{(R,P)}$ be a knowledge base with $R \subseteq \mathcal{R}$ and $P \subseteq \mathcal{P}$ and a set of target resources $O \subseteq R$. Let

$$\text{inc}(t) = \{ p(r, t) \in K \mid r, t \in R, r \neq t \}$$

be the set of all incoming edges for a given resource t . For each target resource $t \in O$ the direct connectivity $\text{dcc}(t)$ is the number of ingoing edges in $K_{(R,P)}$:

$$\text{dcc}(t) = |\text{inc}(t)| \quad \text{with } t \in O$$

The direct connectivity measure simply counts all direct links to the target resource. This measure can be refined for different types of knowledge bases: For

a decision support system, we may introduce a *static connectivity measure* that only counts edges representing explicit knowledge contained in the knowledge base. In contrary, a *dynamic connectivity measure* counts all property occurrences representing actual user input and derivation of a target resource.

Also, different subclasses of the direct connectivity measure may discriminate between the formality of the originating resource, a *formal knowledge connectivity measure* will only count edges that are describing explicit derivation knowledge. An *informal knowledge connectivity measure* will count all edges with informal knowledge as source nodes, such as text paragraphs or multimedia.

Aggregated Connectivity Often the simple counting of incoming links of a target resource does not sufficiently reflect the connectivity. When introducing strong problem-solving knowledge for the derivation of target resources, the simple connectivity is less interesting. Rather the connectivity of a target resource is reflected by its principal derivability, i.e, whether incoming edges are able to actually derive the resource or not. In this case, we need to define sub-properties of *serves*, that reflect the different possibilities of the used problem-solving knowledge. For instance, score-based knowledge requires a special *serves* property for each possible score weight. When representing Bayesian network knowledge bases, special *serves* properties need to reflect the probability.

Besides the sub-properties of *serves*, we also need to introduce an aggregation function *agg*, that merges all edges pointing to a target resource. It is important to note that this aggregation function *agg* needs to be tailored to the particularly knowledge representation used. We generalize the direct connectivity measure to the aggregated connectivity measure.

Definition 5 (Aggregated Connectivity). Let $K_{(R,P)}$ be a knowledge base with $R \subseteq \mathcal{R}$ and $P \subseteq \mathcal{P}$ and a set of target resources $O \subseteq R$. For each resource $t \in O$ the aggregated connectivity *acc* is computed by the outcome of an aggregation function *agg* applied on all incoming properties:

$$acc(t) = agg(inc(t)) \text{ where } t \in R$$

Please note, that the measure is not a monotonic function with respect to different formalization phases, since the set of target resources can grow during formalization.

Example 3 For the representation of a score-based knowledge base, we introduce the following three sub-properties of *serves*: *serves*₁, *serves*₂, and *serves*₃. Each property *serves*_{*i*} represents a positive score weight and the respective weight can be retrieved by $w(serves_i) = i$. For the aggregation of incoming edges *E* we define a target resource to be connected iff the sum of weights of these properties exceeds a given *min* threshold:

$$agg_{sc}(E) = \begin{cases} 1 & : \sum_{e \in E} w(e) > min \\ 0 & : otherwise \end{cases}$$

3 Semantic Technologies and Connectivity

In the previous chapter we introduced an abstract model to jointly represent knowledge at different levels of formality. We now describe an implementation of these concepts by using semantic technologies.

```
@prefix ex: <http://example.org/ns#> .

# Triples of Example 1

ex:Target rdf:type rdfs:Class ; rdfs:label "Target resource" .
ex:Serves rdf:type rdfs:Class ;
  rdfs:label "serves" ;
  rdfs:comment "The subject serves/supports the object." .
ex:D rdf:type rdfs:Class ; rdfs:label "Decision" ;
  rdfs:comment "Represents the class of all target concepts." .
ex:T rdf:type rdfs:Class ; rdfs:label "Text Paragraph" ;
  rdfs:comment "Represents the class of all text paragraphs." .
ex:M rdf:type rdfs:Class ; rdfs:label "Multimedia" ;
  rdfs:comment "Represents the class of all multimedia resources." .
ex:r1 rdf:type ex:T ;
  rdfs:label "Lorem ipsum..." .
ex:r2 rdf:type ex:D ;
  rdfs:label "Decision 1" .
ex:r3 rdf:type ex:M ;
  rdfs:label "A picture" .
ex:p3 rdf:type ex:Serves ;
  rdf:subject ex:r1 ;
  rdf:object ex:r2 .
ex:p5 rdf:type ex:Serves ;
  rdf:subject ex:r3 ;
  rdf:object ex:r1 .

# Incremental formalization of Example 2

ex:F rdf:type rdfs:Class ; rdfs:label "Formal" ;
  rdfs:comment "Represents formal knowledge." .
ex:r4 rdf:type ex:F ;
  rdfs:label "Rule 1" .
ex:p7 rdf:type ex:Serves ;
  rdf:subject ex:r4 ;
  rdf:object ex:r2 .
ex:refines rdf:type rdf:Property .
ex:p7 ex:refines ex:p3 .
```

Program 1: RDFS implementation of the previous examples in Turtle language.

3.1 Semantic Representation

The presented concepts can be instantly represented as RDF(S) ontology [12]. Additionally, as the de-facto standard the SKOS ontology [9] will be used to represent the hierarchical relations between resources. For the later definitions we use the Turtle language [11]. Turtle was recently published as a W3C recommendation to describe RDF data.

Within a multi-modal knowledge base we transfer all resources to RDF resources. In RDFS, we distinguish classes and instances, whereas classes are all resources that are the target of a *type* property. This convention is implemented by transferring all *type* properties to `rdf:type` properties. Analogously, we implement the *broader* property of the general definitions as the `skos:broader` property in the ontology. In Program 1 we implement the resources and properties of Example 1 as an RDFS ontology.

Please note that we added the class `Target` to represent instances of *target resources*. The class `D` represents decisions of the knowledge base and thus is a sub-class of `Target`. Also the property *serves* was not directly implemented as an RDF property but was reified as a class in order to represent refinements of *serves* relations; see for instance the implementation of relation `ex:p7`.

3.2 Querying the Connectivity

The following query shows the direct connectivity as introduced in Definition 4 as a SPARQL query [10].

```
SELECT ?broaderTarget ?targetObject ?covCount
WHERE {
  {
    SELECT ?targetObject (COUNT(?servesRel) AS ?covCount)
    WHERE {
      ?targetObject rdf:type ex:Target .
      ?servesRel    rdf:type ex:Serves ;
                   # rdf:subject/rdf:type ex:F ;
                   rdf:object ?targetObject .
    }
    GROUP BY ?targetObject
  }
  {
    SELECT ?targetObject ?broaderTarget
    WHERE {
      ?targetObject rdf:type ex:Target .
      OPTIONAL { ?targetObject skos:broader ?broaderTarget . }
    }
  }
}
```

Program 2: SPARQL query to retrieve the count of direct `serves` relations to target resources.

Besides the identifier of the target object (`targetObject`) and its number of ingoing *serves* relations (`covCount`) also the broader target resource is retrieved when available. The broader resource is required in many cases, for instance, when defining a more complex connectivity measure that also integrates the connectivities of predecessor or successor resources.

The query counts all *serves* relations independent of its degree of formality. By adding the `rdf:subject/rdf:type ex:F` to the `?servesRel` block, the query will show only *serves* relations from formal sources (the line is commented in the SPARQL query).

3.3 Visualization of Connectivity

Especially for larger knowledge bases it is essential to visualize the retrieved connectivities in order to allow for an intuitive access and overview to the connectivity state. Often target resources are organized in a hierarchical structure. Then, visualizations such as TreeMap or SunBurst are appropriate, see [7] for an evaluation work. We provide some examples for concrete visualizations in the following section.

4 Case Study

To demonstrate the ideas of this paper we report on the incremental formalization of knowledge bases in two different projects. The first project considers the development of the collaborative decision support system KnowSEC. The second case study shows the usage of a function hierarchy defined for a machine-building company.

4.1 Derivability of Decisions in KnowSEC

KnowSEC is used to support substance-related work and workflows within a unit of the Federal Environment Agency (Umweltbundesamt) by the application of knowledge based decision modules. The name KnowSEC stands for "Managing Knowledge of Substances of Ecological Concern" and the system only considers substances under REACH [5]. The multi-modal knowledge representation of the system was recently described in [3]. The KnowSEC system is an extension of the semantic wiki KnowWE [1], where informal knowledge as well as formal problem-solving knowledge and ontologies are managed. It provides plugins for automated testing and debugging knowledge bases including continuous integration.

The KnowSEC system supports the work on substances where a substance is classified according to a large number of criteria. Relevant criteria of a substance are for instance toxicity, persistence, bioaccumulation, mobility. These criteria are determined by using specialized decision modules, i.e., knowledge-based interviews that are able to automatically derive that a substance is toxic for instance. Due to the large number of criteria and its complexity of knowledge

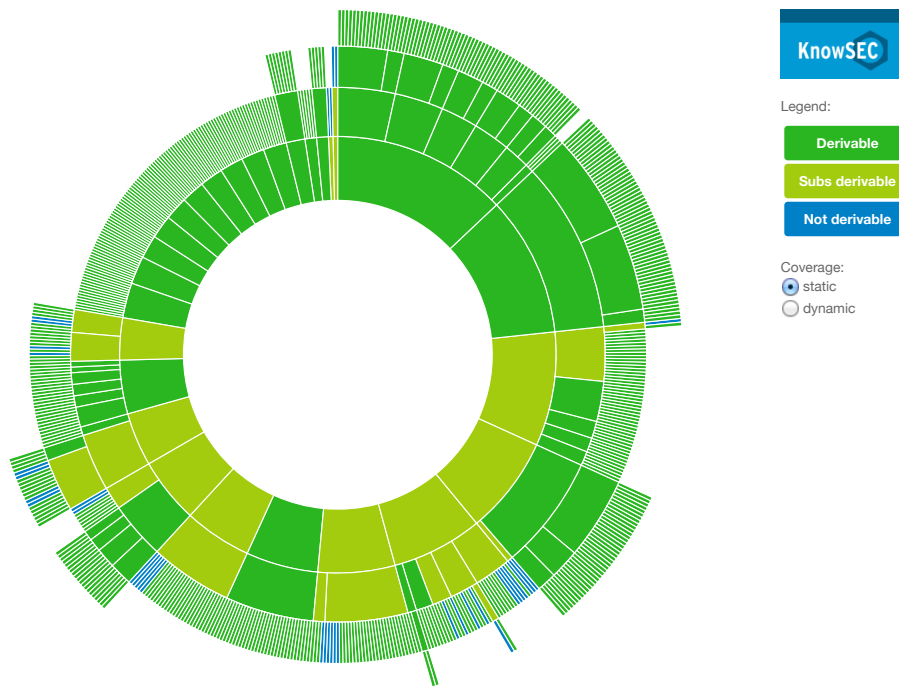


Fig. 4. Current connectivity status of the KnowSEC knowledge base.

not all criteria are covered by decision modules. Then, members of the team are writing informal justifications when applying a criteria to a substance.

The formal and informal criteria justifications are represented in an ontology as well as the possible decisions and substances. Figure 4 depicts a recent static connectivity status of the formal part of the KnowSEC knowledge base as a Sun-Burst visualization [7]. All decisions on criteria were selected as *target resources* and the direct connectivity measure was applied. We instantly see that almost all decisions of the 670 target resources or their successors are derivable, i.e., by having a *serves* relation. Also, we can point and click on segments to retrieve the name of the particular decision. Until today, different versions of the shown visualization were used during the planning.

4.2 Usage of a Function Structure

The second case study was implemented in a project with a mechanical engineering company. The developed ontology describes a function hierarchy of a large machinery, where functions/features of a broad range of machines are represented in a common hierarchical structure. During the development and right after its completion the applicability and utility of the structure was evaluated by using it in real-world use cases. In Figure 5 the hierarchical structure is shown in



Fig. 5. Current connectivity status of an ontological symptom hierarchy.

a SubBurst visualization. Outer partitions are narrower functions, whereas inner partitions represent broader functions. The colors of the partitions represent the use of the particular function in a real-world use case. Each use was represented as a *serves* relation. Here, an aggregated connectivity measure was applied to include also functions into the analysis, that do not directly occur in the use cases but are nevertheless represented because of an occurrence of narrower functions (transitive use).

5 Conclusions

Incremental formalization can help to reduce the development risks of large knowledge bases. It proposes to initially fill the knowledge base with informal chunks of knowledge, e.g., documents and multimedia. In subsequent steps (relevant) parts of the knowledge base are incrementally formalized into a computer-interpretable format. We introduced an abstract graph-like interpretation to cope with such multi-modal knowledge representations and we showed how this interpretation can be implemented by using semantic technologies. In the introduction we posed four questions that are relevant during the formalization of knowledge bases: the connectivity, the formality, the next formalization steps, and the improvement steps. With the introduced measure the first two questions

can be answered, but it also supports the analysis of the remaining questions, e.g., unconnected and unformalized parts are typical candidates for the next formalization phase. In the best case the measures are visualized for intuitive interpretation. In two case-studies we briefly sketched their application and visualization in an industrial setting.

References

1. Baumeister, J., Reutelshoefer, J., Belli, V., Striffler, A., Hatko, R., Friedrich, M.: KnowWE - a wiki for knowledge base development. In: The 8th Workshop on Knowledge Engineering and Software Engineering (KESE2012). http://ceur-ws.org/Vol-949/kese8-05_04.pdf (2012)
2. Baumeister, J., Reutelshoefer, J., Puppe, F.: Engineering intelligent systems on the knowledge formalization continuum. *International Journal of Applied Mathematics and Computer Science (AMCS)* 21(1) (2011), <http://ki.informatik.uni-wuerzburg.de/papers/baumeister/2011/2011-Baumeister-KFC-AMCS.pdf>
3. Baumeister, J., Striffler, A., Brandt, M., Neumann, M.: Towards continuous knowledge representations in episodic and collaborative decision making. In: The 9th Workshop on Knowledge Engineering and Software Engineering (KESE2013). vol. CEUR Proceedings Vol-1070 (2013), http://ceur-ws.org/Vol-1070/kese9-03_05.pdf
4. Kohlhase, A., Kohlhase, M.: Towards a flexible notion of document context. pp. 181–188. <http://kwarc.info/kohlhase/papers/sigdoc2011-flexiforms.pdf>
5. Nendza, M., Müller, M., Wenzel, A.: Regulation under REACH: Identification of potential candidate chemicals based on literature, environmental monitoring, (non)european regulations and listings of substances of concern. Final report FKZ 360 12 019, Federal Environment Agency (UBA), Dessau, Germany (2009)
6. Reutelshöfer, J., Baumeister, J.: Supporting direct knowledge acquisition by customized tools: A case study in the domain of cataract surgery. In: FGWM'13: Proceedings of German Workshop of Knowledge and Experience Management (at LWA'2013) (2013)
7. Stasko, J., Catrambone, R., Guzdial, M., Mcdonald, K.: An evaluation of space-filling information visualizations for depicting hierarchical structures. *Int. J. Human-Computer Studies* 53, 663–694 (2000)
8. Tudorache, T., Nyulas, C.I., Noy, N.F., Musen, M.: Using semantic web in ICD-11: Three years down the road. In: The 12th International Semantic Web Conference (ISWC), In-Use Track. pp. 195–211. Springer (2013)
9. W3C: SKOS Simple Knowledge Organization System reference: <http://www.w3.org/tr/skos-reference> (August 2009)
10. W3C: SPARQL 1.1 recommendation: <http://www.w3.org/tr/sparql11-query/> (March 2013)
11. W3C: RDF 1.1 Turtle – W3C Recommendation. <http://www.w3.org/TR/turtle/> (February 2014)
12. W3C: RDF Schema 1.1 – W3C Recommendation. <http://www.w3.org/TR/rdf-schema/> (February 2014)
13. Wagner, C.: Breaking the knowledge acquisition bottleneck through conversational knowledge management. *Information Resources Management Journal* 19(1), 70–83 (2006)

How can SMEs assess the risk of organisational knowledge?

Susanne Durst¹, Michael Leyer^{2*}

¹ University of Skövde, Skövde, Sweden
susanne.durst@his.se

² Frankfurt School of Finance & Management, Frankfurt, Germany
m.leyer@fs.de

Abstract. Understanding how processes are executed is essential for all companies. While a certain amount of this knowledge can be explicated, a considerable amount is tacit, thus, it is in the mind of the employees. If this knowledge is not shared between organisational members knowledge loss/knowledge attrition is likely to occur. Especially SMEs have a high danger of knowledge loss as knowledge is concentrated on a limited number of individuals. To overcome this problem, we propose a risk-oriented knowledge map for SMEs. Based on the process architecture, risk of processes can be assessed. This allows identifying the knowledge risks associated with staff and thus providing the fundamental starting point for management to promote knowledge sharing as well as other knowledge management practices in the company to better cope with the danger of losing relevant knowledge.

Keywords: Organisational knowledge, Knowledge Management, SMEs, Risk evaluation

1 Introduction

Knowledge of process execution is essential for any company. But small and medium sized enterprises (SMEs) in particular often lack the capacities and time to set up a profound knowledge management system that could assist developing this understanding. This is problematic as these companies heavily rely on the knowledge of a small number of organization members [1].

* Both authors contributed equally to this work and should be considered co-first authors.

Copyright © 2014 by the paper's authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>

In situations in which questions/problems arise, sooner or later the organization member knowing the information is being asked, but before the person having a question wastes his/her working time and energy by asking others and/or searching through the organisation's documentation for a solution. This is not only costly [2] but it also increases the danger that organization members permanently reinvent the wheel instead of developing new knowledge. In the worst case, the firm's survivability is at risk.

SMEs typically do not have established process documentations, because either the whole organisation is still evolving, or missing time or resources keep them from establishing a systematic documentation. Additionally, it may be the case that individuals in charge are not aware of the benefits of a systematic documentation. However, at the same time people need an overview of who is working in which process, especially in firms where the processes are closely linked. Documentation is a systematic approach, a system that has to be designed and which requires people who have time for documentation. The potential of documentation, however, should not be overestimated as not everything can be documented (e.g. tacit knowledge) and thus stored. Indeed, small amounts of knowledge can be documented where necessary (explicit knowledge), but overall most of the knowledge is in the minds of the organization members (implicit and tacit knowledge) [3].

Not all knowledge is critical to organizations. Critical knowledge is typically more complex, abstract, and context dependent, so the knowledge to be retained is implicit or tacit [4]. Additionally, knowledge that has been relevant in the past may become obsolete over time or it has simply be forgotten because of time elapses [5]. Therefore, knowledge is in a constant state of change and should be continuously updated. Due to a comparably smaller number of employees, SMEs have the advantage of a reduced division of tasks within processes and thus less knowledge exchange is required between employees. However, this is also a disadvantage as knowledge may be concentrated on a limited number of employees. Those persons own a lot of knowledge but may not share it because of missing capacities and time and a feeling that they cannot gather additional knowledge outside their area of responsibility [6]. Consequently, there are key individuals who dispose of critical knowledge which in turn causes an increased danger of knowledge loss/knowledge attrition if they are not available (e.g. temporarily).

Against this background, the aim of this paper is to examine knowledge management from a knowledge at risk perspective. More precisely, the emphasis lies on knowledge risks associated with production processes. The discussion is conducted from the viewpoint of SMEs.

2 Theoretical background

2.1 Foundations of knowledge management

Knowledge can be characterized differently. For example, the distinction between explicit knowledge and tacit knowledge can be discussed [7]. Explicit knowledge consists of the means by which information is made physical, identifiable, and trans-

ferable, for example, on a compact disc or document. Explicit knowledge can be purchased, repeated, reinvented, and stolen. It dwells separately from the individual or the company. Whereas tacit knowledge “refers to the real-time, often subconscious, cognitive, or other processes that is utilized and taken for granted” [8, p. 10]. Previous experiences are combined with these processes to make a decision go forward. Despite the importance of explicit knowledge, tacit knowledge is believed to be the higher value knowledge [9], or as Haldin-Herrgard [10] regards it as the topping to reach excellence in a job. Unless shared with others, tacit knowledge dies with the individual. Tacit knowledge can be acquired through watching and replication, which often represents vocational training [11].

Kogut and Zander [12] divide knowledge into information and know-how. According to these authors, “information implies knowing what something means”, whereas know-how is “a description of knowing how to do something” (p. 386). Grant [13] discusses the types “knowing how” and “knowing about”, thereby he associates the former with tacit knowledge that is exposed through application, and the latter with explicit knowledge that is exposed through communication.

There have been some debates whether knowledge can be managed or not. Among proponents of knowledge management there is agreement that there is no single way for a firm to manage its knowledge, as the nature of the market, the intensity of competition, the firm's strategy, its product/service organization, the type of knowledge process that is emphasized, and the nature of labor the firm recruits will influence the type of knowledge management strategy suitable for the firm [14]. Based on these aspects, only a broad definition of knowledge management might be useful. Bounfour [15, p. 156] defines knowledge management “as a set of procedures, infrastructures, technical and managerial tools, designed towards creating, circulating (sharing) and leveraging information and knowledge within and around organizations”. Among the different knowledge management activities (e.g. knowledge identification, knowledge creation, knowledge dissemination etc.), it seems that knowledge creation and knowledge transfer are viewed as more important than the other activities. Markus [16], however, stresses (she talks about reuse) that the effective reuse of knowledge should take a stronger role as it is clearly associated with organizational effectiveness.

In the same vein, researchers have highlighted the link between the reuse of knowledge and developing competitive advantage [17] or in the context of innovation [18]. Consequently, one can assert that a strong consideration of existing knowledge can help firms to improve performance and thus sustain competitive advantage. Given the competitive pressure firms are facing in today's business environment, a non-utilization or waste of knowledge is not only costly [2] but also dangerous. As initiatives which are, after all, repeating already existing knowledge instead of creating new knowledge or recombining it in new ways can result in situations in which valuable resources and time are bound and thus not available to other more important business operations. Consequently, this may be damaging not only for the company concerned but also for the economy, as continuously reinventing the wheel blocks from developing. Therefore, in this paper we take a knowledge at risk perspective that is, addressing situations in which knowledge not used becomes a liability or a risk [19].

2.2 Specifics of knowledge management in SMEs

The owners' or managing-directors' centrality often found in SMEs [20] signifies that particularly these persons are responsible for the recognition of the benefits related to knowledge management as otherwise the necessary structures and systems are not supported and therefore not implemented. Additionally, day-to-day operations require high attention, resulting very often in the situation that time is missing to identify and recognise the benefit of knowledge management as well as other managerial issues [21]. This often results in situations in which knowledge is being kept in the heads of the owner and some key employees rather than physically stored [3].

Yet some SME specific characteristics speak for knowledge management implementation in SMEs. For example, employees and owner are usually close, a fact that can facilitate the flow of knowledge [22]. Additionally, informal communication and not through documentation or other written documents represents the main basis for knowledge transfer [3, 23].

The empirical studies on knowledge management practice in SMEs have indicated that they are less advanced when dealing with the topic [24, 25]. Furthermore, they are "having a more mechanistic approach to knowledge construction and relying less on social interaction" compared to large businesses [24, p. 240]. The study by Beijerse [26] showed that not a single SME had a knowledge management strategy in place. Furthermore, it appeared that the companies use a variety of instruments to evaluate, to acquire, to develop, and to share knowledge. Yet, these tools are often not considered as instruments for knowledge management. A similar result was obtained in a study conducted by Desouza and Awazu [22], they call the SMEs' way of dealing with knowledge "the humanistic way" (p. 40). Additionally, the authors found that the SMEs surveyed have a tendency to put knowledge generated immediately into practice instead of storing it. Moreover, their study stressed that smaller firms make themselves less susceptible to knowledge loss if it does not reside in the brain of only one employee. Nunes et al. [27] conducted a study that was targeted to obtain a better understanding of knowledge management awareness, perceptions, and requirements in SMEs. The results showed that these companies do not see knowledge management as a crucial function. However, even though they do not have a knowledge management strategy, guidelines and other procedures set to deal with knowledge management related issues have been observed. Additionally, the creation, storage, and dissemination of knowledge is not linked to the accessibility of appropriate IT systems. Hutchinson and Quintas [28] investigated knowledge practices in SMEs. They found that within SMEs certain processes and measures are available which indicate that they do knowledge management, but it happens mostly in an informal matter. Among the few firms having established formal knowledge management, the authors found that those interviewees themselves used the term knowledge management for their activities. Based on these insights, Hutchinson and Quintas concluded that the concept and vocabulary of knowledge management are increasingly acknowledged and applied in SMEs. Durst and Wilhelm [23], who studied how an SME cope with the danger of knowledge attrition due to personnel turnover or long-term absence, showed the influence of a precarious financial situation on activities related to knowledge management and succession planning. Even though the individuals concerned are aware of needs for improvement, their actual scope of action is centered on the execu-

tion of current orders. Wee and Chua's [29] study confirmed the central role of SME owners with regard to KM activities. Their findings also indicate that knowledge reuse is supported by close proximity of employees. These findings are in line with attributes typically associated with SMEs [20].

2.3 Importance of Risk Management in Knowledge Management

According to Bessis [30, p. 5], risks can be "defined by the adverse impact on profitability of several distinct sources of uncertainty". Risk is assumed to be calculated which displays a clear distinction to the term 'uncertainty', which cannot be calculated [31]. Risk can be divided into financial and non-financial risks. As signalled by the word 'financial', the former classification establishes a relationship with something monetary and quantifiable, whereas the latter does not. Summing up, risk management is primarily aimed at identifying, assessing, monitoring and controlling firm risks [30]. Firms should thereby focus on all types of risk and their management.

In the extant literature, it seems that knowledge is mainly discussed as something of value, i.e. an asset or a skill. Potentially negative aspects, like knowledge as a liability, apart from a few exceptions [e.g. 32, 33, 34] seem to be underestimated. Consequently, knowledge risk management (KRM) is in its infancy as well [35]. In order to address this situation, Massingham proposed a conceptual KRM model that calculates a risk score and a knowledge score. The addition of the latter is considered as a way of gaining deeper insights into the real nature of organizational risk.

Besides this promising move forward, one can determine that our discussion on knowledge is rather unbalanced. Yet companies that fail to properly manage their critical knowledge to secure its value-creation potential undergo significant risks, for example loss of expertise or reinvention of knowhow. Therefore, the need to carefully manage the downside risks of knowledge is high too. Managers and entrepreneurs cannot afford to neglect knowledge risks even though they might be more familiar with financial capital and the risks related to this asset category [19]. Given the resource constraints, an integration of a risk management approach in knowledge management activities is particularly relevant for SMEs [32].

3 Process-oriented knowledge risk map in SMEs

3.1 Overview

Business processes are essential for companies as they define how input (e.g. raw material) is transformed into output (products and services) [36]. The knowledge regarding process execution can partly be explicated and is partly tacit, i.e. within the mind of employees [37]. To determine the risk level of process-oriented knowledge, the subsequent steps have to be followed:

1. The process architecture has to be captured which describes the main connections between processes and sub-processes [38].

2. Each micro process is rated regarding its importance from a business perspective resulting in risk profiles of processes.
3. Employees possessing knowledge and explicit knowledge have to be linked to the respective processes allowing for the desired risk assessment.

3.2 Process architecture

The first step regarding the knowledge risk assessment is to identify the relevant processes, i.e. the procedural knowledge which is relevant for value creation. Processes provide the basis for assigning relevant knowledge in an organization. Nevertheless, recording processes in detail is not the objective of this step, as only the processes and their main activities are relevant for the purpose of the process architecture. The purpose of a process architecture is to describe the basic structure of an organization and the main connections between its processes and sub-processes [38]. If a significant number of processes is mapped, the illustration should comprise multiple levels. In this way, the core processes can be mapped on the top level, the more detailed processes (macro processes) on the middle level and the micro processes on the lowest level [39]. A core process can be for example “consultancy of SMEs” of a tax consultant. Macro processes of this core process can be “investment consultancy” and “tax declarations”. On the micro process level “tax declarations” can be further split into “preparing balance sheet” and “gathering documents”. The details of process execution, containing explicit work instructions, are not incorporated in the micro processes.

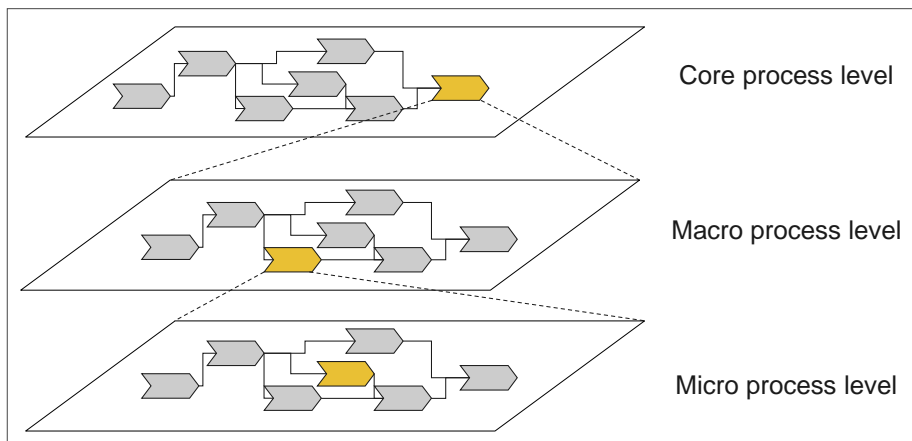


Fig. 1. Generic process architecture

Gathering the necessary information can either take place manually or electronically. In the first case, process owners have to be questioned and information has to be summarized in the above displayed structure. In the second case, electronically documented processes can be used. In such a setting, process execution is recorded by event logs which contain information on which process was executed, when and by

which employee. To set up the process architecture from such data, process mining can be applied which helps to avoid redundant and incorrect process acquisition [40].

3.3 Risk profile for processes

The second step is determining the importance of process-related knowledge. This has to be done on the micro process level and results can be aggregated on the macro and core process level. Three essential characteristics of micro processes have to be determined:

- Frequency of execution: The frequency allows evaluating how often a micro process is executed and is the first characteristic to determine importance. Using manual data, experts have to estimate the frequency or measurements have to be conducted. In the case of existing event logs, the frequency of execution of micro processes can be determined with techniques of process mining [40].
- Value added: Value added covers the cost and profit of each micro process. Activity based costing is the starting point to determine the costs for every micro process [41]. In addition, it has to be estimated how much value is added with each micro process execution. As the value might differ with products or services, this can be different for the execution within different macro processes [42]. The information about assignment of micro processes to macro processes is contained in the process architecture.
- Legal requirements: Lastly, legal requirements should be rated to identify external restrictions impacting cost of the micro processes. Importance of a micro process is enhanced if legal requirements are high and problems as well as fines can occur in case of non-conforming process execution.

Aggregating these three characteristics, risk profiles for micro processes (RPMiP) can be set up following the basic rule of multiplying the occurrence of a micro process with the value added.

Additionally, the resulting value is divided through the number of employees assigned to the process. There are three categories for assignment: Currently working in the micro process, supervising work in the micro process and having worked previously (operational or supervising) in the micro process. Employees (e) are assigned in one of the three categories to the relevant micro processes. In case of working currently in a micro process variable a is used whereas variable b is used if an employee has worked previously in the micro process but the last involvement is not older than one year.

Our formula to calculate knowledge relevant risk profiles is as follows:

$$RPMiP_x = \frac{\sum_y (Occ_{x,y} (VA_{x,y} - C_{x,y})) + (PLR_x * CLR_x)}{n(e_{x,a}) + 0.5 * n(e_{x,b})}, \quad (1)$$

$$1 \leq x \leq n(MiP), 1 \leq y \leq n(MaP)$$

Occ is the number of occurrences of a micro process, VA the value added, C the cost, PLR the probability of a legal risk, CLR the cost of a legal risk and n(MiP) is the total number of micro processes as well as n(MaP) is the total number of macro processes in which a micro process is occurring. n(e) is the number of employees being assigned to a micro process. Each micro process x receives one risk profile considering that the micro process is executed within different macro processes y.

3.4 Identification of critical organisation members

The third step is to link organisational members to micro processes. Two indicators are relevant from a risk perspective.

First, an aggregated risk score per employee is calculated, i.e. the respective RPMiP values are aggregated per employee. The aggregated risk profile formula per employee is:

$$ARPE_e = \sum_x RPMiP_{x,e}, 1 \leq e \leq n(E) \quad (2)$$

Employees (n(E) indicates the number of employees in the organisation) can be ranked according to these values, thus indicating the most critical organisation members from an aggregation point of view.

Second, it can be counted in how many cases only one employee regarding operational knowledge is available regarding specific micro processes. The number of these occurrences can be aggregated per employee, thus, employees with the highest count are more critical.

$$IRPE_e = \sum_x e \{ n(e_{x,a}) n(e_{x,a}) = 1 \text{ and } n(e_{x,b}) = 0 \}, 1 \leq e \leq n(E) \quad (3)$$

4 Discussion

The proposed process-oriented knowledge risk map has several benefits to offer. Firstly, managers and owner-managers of SMEs will obtain an in depth overview of the knowledge needed to perform the firm's business processes. This understanding will make possible a more proactive knowledge management in terms of developing and initiating training and further education of process-based knowledge and competences. On the other hand, and perhaps more critical, this understanding can help reduce risks related to the business processes, e.g. business is not disrupted in case of illness or leaving employees. Having information about process-related knowledge of critical organisation members will also provide the necessary knowledge for succession planning or contingency planning. As a consequence time and resources are gained that can be invested in business operations or strategic planning that are more relevant for the firm's organizational development. For example, the potential exit of key employees may be addressed with a reduction of individual tasks assigned to

them or with increased team or project work [43]. In order to have this kind of situation in an organization, it is important to determine the risk level of processes and the subsequent knowledge, the presented formula for calculating relevant risk profiles can help on this road.

5 Conclusions

In this paper the aim was to stress the importance of having a sufficient understanding of business processes and their execution. It was argued that this can help firms to assess the risk of knowledge loss. In view of SMEs and their specific characteristics, reducing this danger should be an area of particular interest. In order to address this topic we propose a process oriented knowledge risk map that is intended to support SMEs not only in getting a better overview of their business processes in general but also in obtaining a more fine-grained understanding of the different sub-processes and their sequences. This in turn makes visible specific areas where knowledge loss is likely to occur. Therefore, it can increase the awareness towards critical knowledge and possible costs of losing it (DeLong, 2004).

From a theoretical point of view, this study provides novel insights into the study of knowledge reuse as it draws particular attention to the downside risks of knowledge. These insights thus expand our body of knowledge regarding knowledge management in SMEs and knowledge risk management in general.

The present study also offers SMEs insights and ways of how to cope with the danger of knowledge loss in their business processes. Forward-looking SMEs that manage and distribute their process-oriented knowledge actively are those that can most successful reduce this danger.

The process oriented knowledge risk map has been developed based on a synthesis of existing literature. The present paper should therefore be viewed as a promising basis for further theorising and empirical testing. For example, an analysis of the SMEs' handling of business processes would provide a useful basis for the further development of the proposed process oriented knowledge risk map. In addition, a better understanding of SME business processes would help to develop SME-specific solutions that keep the danger of knowledge waste at a minimum. Future research may also focus on the weighting of different process specific risks.

References

1. Durst, S., Wilhelm, S.: Knowledge management in practice. Insights into a medium-sized enterprise's exposure to knowledge loss. *Prometheus* 29, 23-38 (2011)
2. Bolisani, E., Paiola, m., Scarso, E.: Knowledge protection in knowledge-intensive business services. *Journal of Intellectual Capital* 14, 192-211 (2013)
3. Wong, K.Y., Aspinwall, E.: Characterizing knowledge management in the small business environment. *Journal of Knowledge Management* 8, 44-61 (2004)
4. DeLong, D.W.: *Lost knowledge. Confronting the Threat of an Aging Workforce*. Oxford University Press, Oxford (2004)

5. Tan, H.C., Carrillo, P., Anumba, C., Kamara, J.M., Bouchlaghem, D., Udejaja, C.: Live capture and reuse of project knowledge in construction organisations. *Knowledge Management Research & Practice* 4, 149-161 (2006)
6. Durst, S., Edvardsson, I.R.: Knowledge management in SMEs. A literature review. *Journal of Knowledge Management* 16, 879-903 (2012)
7. Nonaka, I., Takeuchi, H.: *The Knowledge-Creating Company. How Japanese companies create the dynamics of innovation.* Oxford University Press, New York (1995)
8. Tollington, T.: *Brand Assets.* Wiley, Chichester (2002)
9. Treleaven, L., Sykes, C.: Loss of organizational knowledge. From supporting clients to serving head office. *Journal of Organizational Change Management* 18, 353-368 (2005)
10. Haldin-Herrgard, T.: Difficulties in diffusion of tacit knowledge in organizations. *Journal of Intellectual Capital* 1, 357-365 (2000)
11. Nonaka, I.: A dynamic theory of organizational knowledge creation. *Organization Science* 5, 14-37 (1994)
12. Kogut, B., Zander, U.: Knowledge of the firm, combinative capabilities, and the replication of technology. *Organization Science* 3, 383-397 (1992)
13. Grant, R.M.: Toward a knowledge-based theory of the firm. *Strategic Management Journal* 17, 109-122 (1996)
14. Hislop, D.: *Knowledge management in organizations.* Oxford University Press, Oxford (2009)
15. Bounfour, A.: *The Management of Intangibles. The Organization's Most Valuable Assets.* Routledge, London, New York (2003)
16. Markus, M.L.: Toward a Theory of Knowledge Reuse. Types of Knowledge Reuse Situations and Factors in Reuse Success. *Journal of Management Information Systems* 18, 57-93 (2001)
17. Szulanski, G.: Exploring Internal Stickiness. Impediments to the Transfer of Best Practice Within the Firm *Strategic Management Journal* 17, 27-43 (1996)
18. Haefliger, S., von Krogh, G., Spaeth, S.: Code Reuse in Open Source Software. *Management Science* 54, 180-193 (2008)
19. Brunold, J., Durst, S.: Intellectual capital risks and job rotation. *Journal of Intellectual Capital* 13, 178-1495 (2012)
20. Bridge, S., O'Neill, K.: *Understanding Enterprise, Entrepreneurship and Small Business.* Palgrave Macmillan, New York (2013)
21. Hofer, C.W., Charan, R.: The Transition to Professional Management. Mission Impossible? *International Small Business Journal* 26, 131-154 (2008)
22. Desouza, K.C., Awazu, Y.: Knowledge management at SMEs. Five Peculiarities. *Journal of Knowledge Management* 10, 32-43 (2006)
23. Durst, S., Wilhelm, S.: Knowledge management and succession planning in SMEs. *Journal of Knowledge Management* 16, 637-649 (2012)
24. McAdam, R., Reid, R.: SME and large organisation perceptions of knowledge management. Comparisons and contrasts. *Journal of Knowledge Management* 5, 231-241 (2001)
25. Wong, K.Y., Aspinwall, E.: An empirical study of the important factors for knowledge-management adoption in the SME sector. *Journal of Knowledge Management* 9, 64-82 (2005)
26. Beijerse, R.P.: Knowledge management in small and medium-sized companies. *Knowledge Management for Entrepreneurs. Journal of Knowledge Management* 4, 162-179 (2000)
27. Nunes, M.B., Annansingh, F., Eaglestone, B.: Knowledge management issues in knowledge-intensive SMEs. *Journal of Documentation* 62, 101-119 (2006)

28. Hutchinson, V., Quintas, P.: Do SMEs do Knowledge Management? *International Small Business Journal* 26, 313-154 (2008)
29. Wee, J.C.N., Chua, A.Y.K.: The peculiarities of knowledge management processes in SMEs. The case of Singapore. *Journal of Knowledge Management* 17, 958-972 (2013)
30. Bessis, J.: *Risk Management in Banking*. Wiley, Chichester (1998)
31. Bullen, E., Fahey, J., Kenway, J.: The knowledge economy and innovation: Certain uncertainty and the risk economy. *Discourse studies in the cultural politics of education* 27, 53-68 (2006)
32. Durst, S.: Innovation and intellectual capital (risk) management in small and medium-sized enterprises. *International Journal of Transitions and Innovation Systems* 2, 233-246 (2012)
33. Massingham, P.: Measuring the Impact of Knowledge Loss. More Than Ripples on a Pond? *Management Learning* 39, 541-560 (2008)
34. Neef, D.: Managing corporate risk through better knowledge management. *The Learning Organization* 12, 112-124 (2005)
35. Massingham, P.: Knowledge risk management: a framework. *Journal of Knowledge Management* 14, 464-485 (2010)
36. Wernerfelt, B.: A Resource-Based View of the Firm. *Strategic Management Journal* 5, 171-180 (1984)
37. Hawryszkiewicz, I.: *Knowledge Management. Organizing Knowledge Based Enterprises*. Palgrave, New York (2010)
38. Österle, H., Winter, R.: *Business Engineering*. Springer, Berlin (2003)
39. Leyer, M., Claus, N.: Toward an agile knowledge connection of employees with regard to business processes. In: *Proceedings of the 46th Hawaii International Conference on System Sciences*, pp. 3436-3445. IEEE Computer Society, (Year)
40. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining. Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering* 16, 1128-1142 (2004)
41. Babad, Y.M., Balachandran, B.V.: Cost Driver Optimization in Activity-Based Costing *The Accounting Review* 68, 563-575 (1993)
42. Anderson, S.W., Young, S.M.: The impact of contextual and process factors on the evaluation of activity-based costing systems. *Accounting, Organizations and Society* 24, 525-559 (1999)
43. Schmitt, A., Borzillo, S., Probst, G.: Don't let knowledge walk away. Knowledge retention during employee downsizing. *Management Learning* 43, 53-74 (2012)

Clarification KBS as Consultation-Justification Mash Ups ^{*}

Proposing A Novel Paradigm for All-in-One Knowledge-based Systems

Martina Freiberg, Felix Herrmann, and Frank Puppe

Department of Artificial Intelligence and Applied Informatics, Institute of Computer
Science, University of Würzburg, Am Hubland, D-97074 Würzburg, Germany
`martina.freiberg@uni-wuerzburg.de`
`felix.herrmann@uni-wuerzburg.de`
`frank.puppe@uni-wuerzburg.de`

Abstract. Regarding knowledge-based systems (KBS), the seminal paradigm—perfectly mimicking human experts—is gradually replaced by an increasing demand for enabling users to influence the reasoning process according to their domain knowledge. Therefore, we propose a novel KBS paradigm: *Clarification KBS* as a mash up type of consultation and justification interaction—intended to foster active user participation according to users’ competency, the KBS’ explicability, and the support for learnability. We introduce the theoretical concept of clarification KBS, as well as appropriate UI-/interaction variants. Further, we discuss the results of iteratively evolving and evaluating *ITree*, a specific clarification KBS implementation for the legal domain.

Keywords: Knowledge-based System, Clarification, Justification, User Participation, Learnability, Explicability

Resubmission of Freiberg, M., Herrmann, F., Puppe, F.: Clarification KBS as Consultation-Justification Mash Ups—Proposing A Novel Paradigm for All-in-One Knowledge-based Systems. *Submitted to:* Proceedings of International Conference on Knowledge Engineering and Ontology Development (KEOD 2014)

^{*} Copyright © 2014 by the paper’s authors. Copying permitted only for private and academic purposes. In: T. Seidl, M. Hassani, C. Beecks (Eds.): Proceedings of the LWA 2014 Workshops: KDML, IR, FGWM, Aachen, Germany, 8-10 September 2014, published at <http://ceur-ws.org>