ECML-PKDD 2011 Discovery Challenge Overview

Nino Antulov-Fantulin[1], Matko Bošnjak[1], Martin Žnidaršič[2], Miha Grčar[2], Mikołaj Morzy[3], and Tomislav Šmuc[1]

[1] Rudjer Boškovic Institute, Zagreb, Croatia
[2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] Poznań University of Technology, Poznań, Poland

Abstract. This year's Discovery Challenge was dedicated to solving of the video lecture recommendation problems, based on the data collected at VideoLectures.Net site. Challenge had two tasks: task 1 in which new-user/new-item recommendation problem was simulated, and the task 2 which was a simulation of the clickstream-based recommendation. In this overview we present challenge datasets, tasks, evaluation measure and we analyze solutions and results.

1 General description of the challenge

VideoLectures.Net (VL.Net)[4] is a free and open access multimedia repository of video lectures, mainly of research and educational character. The lectures are given by distinguished scholars and scientists at the most important and prominent events like conferences, summer schools, workshops and science promotional events from many fields of science. The website is aimed at promoting science, exchanging ideas and fostering knowledge sharing by providing high quality didactic contents not only to the scientific community but also to the general public. All lectures, accompanying documents, information and links are systematically selected and classified through the editorial process taking into account also users' comments.

This challenge was organized through the support of the EU project e-LICO[5]. The aims of the challenge are multifold: from research in recommender systems, improvement of the current recommender system of the VL.Net site, to provision of the problem and datasets to the research community. The challenge consisted of two main tasks. Due to the nature of the problem, each of the tasks has its own merit: task 1 simulates new-user and new-item recommendation cold-start mode, while task 2 simulates clickstream (implicit preference) based recommendation. Due to the privacy preserving constraints, data from VL.Net website includes neither explicit nor implicit user profiles. Instead, implicit profiles embodied in viewing sequences (clickstreams) have been transformed, so that no individual viewing sequence information can be revealed or reconstructed. This transformed, viewing related data includes: i) lecture co-viewing frequencies ii) pooled viewing sequences (whose construction will be described later) and is accompanied with rich lecture description related information available: lecture category taxonomy, lecture names, descriptions and slide titles (where available), authors, institutions, lecture events and timestamps. Unlike most of the other publicly available recommendation problems datasets, this dataset contains original content, names and taxonomy. The dataset of the challenge[6] (including

[4] `http://videolectures.net`
[5] `http://www.e-lico.eu`
[6] `http://lis.irb.hr/challenge/`

the leaderboard and the test set), together with task and evaluation descriptions is publicly available for the non-commercial research purposes [28].

We have ensured prize-sponsoring (5500 €) from the European Commission through the e-LICO EU project, 2009-2012 whose primary goal is to build a virtual laboratory for interdisciplinary collaborative research in data mining and data-intensive sciences.

The prizes, for each of the tracks are:

- 1500 € for the first place
- 700 € for the second place
- 300 € for the third place

The prizes, for the Workflow contest are:

- 500 € for the best workflow
- Free admission to RapidMiner Community Meeting and Conference 2012 for the best RapidMiner workflow (sponsor: Rapid-I)

The challenge has been hosted on TunedIt[7].

2 Background

Recommender systems have become an important research area since the first appearance of the information overload for the typical user on the internet. Personalized recommender systems take user profiles into account when the prediction for particular user and item is generated. The prediction techniques for the recommender systems [1–3] can be divided into three main categories: content-based, collaborative-based and hybrid-based prediction techniques.

Content-based techniques [4, 5] are based on interactions between a particular user and all the items in the system. Content-based recommender systems use information about items and the user's past activities on items in order to recommend similar items.

Collaborative filtering techniques [6–8] analyze interactions between all users and all items through users' ratings, clicks, comments, tags, etc. Collaborative filtering recommender systems do not use any specific knowledge about the items except their unique identifiers. These prediction techniques are domain-independent and can provide serendipity recommendations for users. However, collaborative filtering needs sufficient amount of collaborative data in order to recommend for the new user or the new item (the cold-start problem) [9, 10].

Hybrid prediction techniques [11–13] merge collaborative-based and content-based techniques and are more resistant to cold start problems. This challenge was designed to tackle the problems of cold start and hybridization of content and collaborative data in realistic setting of the VL.Net website. In comparison to recommender challenges of recent years (Netflix challenge, KDDCup challenge 2008, KDDCup challenge 2011) this challenge relies on indirect collaborative data, and is more focussed on utilization of content and descriptions of items.

[7] http://tunedit.org

3 Description of the challenge dataset

The data snapshot which is the basis for the VideoLectures.Net dataset was taken in August 2010. At that time, the database contained 8 105 video lectures. 5 286 lectures were manually categorized into taxonomy of roughly 350 scientific topics such as Arts, Computer Science, and Mathematics.

VideoLectures.Net dataset includes:

1. **Data about lectures**: every lecture has a title, type (e.g. lecture, keynote, tutorial, press conference, etc.) language identifier (e.g. *en*, *sl*, *fr*, etc.), number of views, publication date, event identifier, and a set of authors. Many lectures come with a short textual description and/or with slide titles from the respective presentations. Specifically, 5 724 lectures are enriched with this additional unstructured data. The training part of data contains also lecture-pairs coviewing frequencies (CVS - common view score), and pooled sequences related collaborative data, which is not available for the set of test lectures. Test set contains lectures with publication date after July 01, 2009, which are used for task 1 scoring. Neither CVS nor pooled viewing sequences containing these lectures are available in the training data.
2. **Data about authors**: each author has a name, e-mail address, homepage address, gender, affiliation, and the respective list of lectures. The dataset contains 8 092 authors. The data about the authors is represented by authors' names, VL.Net url, e-mail, homepage, gender, affiliation, and pairwise relations to the lectures delivered by the author at VL.Net
3. **Data about events**: a set of lectures can be associated with an event (e.g. a specific conference). In a similar fashion, events can be further grouped into meta-events. An event is described in a similar way as a lecture: it has a title, type (e.g. project, event, course), language identifier, publication date, and a meta-event identifier. The VideoLectures.Net dataset contains data about 519 events and meta-events (245 events are manually categorized, 437 events are enriched with textual descriptions).
4. **Data about the categories**: The data about the categories is represented in the shape of the scientific taxonomy used on VL.Net. The taxonomy is described in a pairwise form, using parent and child relations.
5. **View statistics**: The VideoLectures.Net software observes the users accessing the content. Each browser, identified by a cookie, is associated with the sequence of lectures that were viewed in the identified browser. Temporal information, view durations, and/or user demographics are not available. The dataset contains anonymized data of 329 481 distinct cookie-identified browsers. The data about view statistics is given in the form of frequencies: (i) for a pair of lectures viewed together (not necessarily consecutively) with at least two distinct cookie-identified browsers; (ii) for pooled viewing sequences - triplets of lectures viewed together prior to a given sequence of ten lectures. This is a special construct based on aggregation of click-streams, which is used for training and scoring in task 2.

3.1 Creating pooled viewing sequences

In order to comply with privacy-preserving constraints, lecture viewing sequences for the task 2 have been transformed into what we named pooled sequences. Pooled viewing sequence is given by the set of three lectures on the left side (triplet) and a ranked list of at most ten lectures on the right side. The set of three lectures does not imply an ordering, it is merely a set that comes *upstream* of lectures given on the right of a pooled viewing sequence. Ranked list on the right side of some pooled viewing sequence is constructed from all the clickstreams with the particular triplet on the left side. The transformation process for the construction of pooled viewing sequences is given below.

Consider a sequence of viewed lectures:

$$id_1 \rightarrow id_7 \rightarrow id_2 \rightarrow id_1 \rightarrow id_4 \rightarrow id_5 \rightarrow id_6 \rightarrow id_3$$

We first filter out duplicates (here - id_1):

$$id_1 \rightarrow id_7 \rightarrow id_2 \rightarrow id_4 \rightarrow id_5 \rightarrow id_6 \rightarrow id_3$$

Then, we determine all possible unordered triplets in the sequence. For each triplet, cut the sequence after the right-most lecture from the triplet.

In the above example, if $\{id_1, id_4, id_5\}$ is the triplet, the sequence is cut right after id_5. Finally, increase triplet-specific counts for all the lectures after the cut. In the above example, given the triplet $\{id_1, id_4, id_5\}$, the triplet-specific counts for id_6 and id_3 are increased:

$$\{id_1, id_4, id_5\} \rightarrow id_6 : 1, id_3 : 1$$

Suppose there is another click-stream sequence, that amongst others, contains unordered triplet id_1, id_4, id_5 and that id_6, id_3, and id_7 are lectures appearing after the cut. Then the counts for the $\{id_1, id_4, id_5\}$ are increased as follows:

$$\{id_1, id_4, id_5\} \rightarrow id_6 : 2, id_3 : 2, id_7 : 1$$

3.2 Creating lecture co-viewing frequencies

Consider two sequences of viewed lectures:

$$id_1 \rightarrow id_7 \rightarrow id_2 \rightarrow id_1,$$

$$id_2 \rightarrow id_3 \rightarrow id_7.$$

We first filter out duplicates in sequences:

$$id_1 \rightarrow id_7 \rightarrow id_2,$$

$$id_2 \rightarrow id_3 \rightarrow id_7.$$

Then, we determine lecture co-viewing frequencies (CVS):

$$CVS(id_1, id_2) = 1, CVS(id_1, id_7) = 1,$$

$$CVS(id_2, id_7) = 2, CVS(id_2, id_3) = 1,$$

$$CVS(id_3, id_7) = 1.$$

Table 1: Train-test data statistics

Moment t_2	05.08.2010.
Moment t_1	01.07.2009.
Total number of lectures in the train set	6983
Total number of lectures in the test set	1122
Number of common-view pairs in the train set	363 880
Number of common-view pairs in the test set	18 450

3.3 A train-test split logic

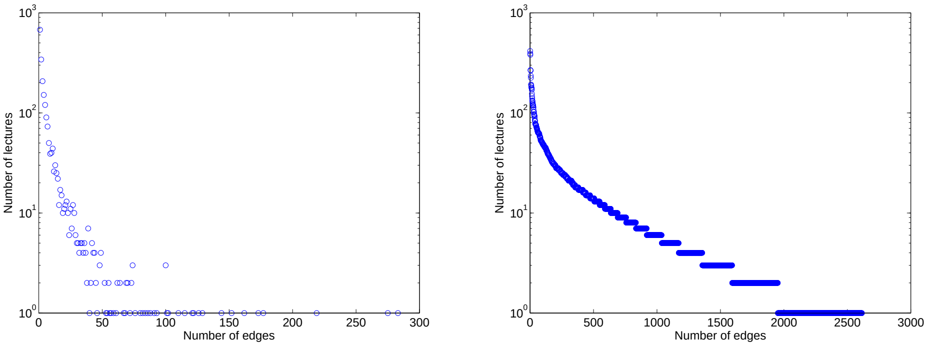
Basic statistics of lectures in the training and test sets are given in Table 1. Common view score matrix CVS is a lecture co-viewing frequency matrix collected at the site at some moment t_2 and represents lecture viewing adjacency matrix of lecture-lecture graph G at the moment t_2. G is undirected weighted graph of all lectures. Each lecture in this graph has associated temporal information - date of publishing at the VideoLectures.Net site. We partition G using the publishing date by some threshold t_1, into two disjoint graphs G_1 and G_2: each lecture in G_1 has publishing date before the date threshold while each lecture in G_2 has publishing date after the date threshold t_1. We define pair common viewing time as a period that two lectures spend together in the system. All lecture pairs $(x_i, x_j) : x_i \in G_1, x_j \in G_1$ have pair common time strictly greater than $(t_2 - t_1)$ value and all lecture pairs $(x_i, x_j) : x_i \in G_1, x_j \in G_2$ have pair common time strictly less than $(t_2 - t_1)$ value.

In oder to make proper the training and test set split based on G_1 and G_2, we had to ensure similar distribution of pair common times in both training and test sets. We have divided nodes from subgraph G_2 in randomized fashion (with some constraints) into two approximately equal sets (G_{21}, G_{22}) and we have appended G_{21} to the training set. Now, the subset of lecture pairs $(x_i, x_j) : x_i \in G_1, x_j \in G_{21}$ from the training set has similar distribution of pair common times that overlaps with times $(x_i, x_j) : x_i \in G_1, x_j \in G_{22}$ from the test set. Figure 1 gives the distribution of edges related to the graphs G_1, G_{22}.

Finaly, the train-test split logic was implemented through the series of steps:

1. Split the lectures by publication date into two subsets: *old* (publication date $<$ July 01, 2009) and *new* (publication date $\geq$ July 01, 2009). Put the *old* lectures into the training set;
2. Move all *new* lectures with parent id occuring in the *old* lecture subset to the training set;
3. Split the rest of the *new* lectures randomly into two disjoint sets of similar cardinality, taking care of their parent ids:
   (a) lectures with the same parent id can be only in one of the sets;
   (b) lectures without parent id are just randomly divided between two sets.
4. Finally, add one of the disjoint sets to the training set; the other disjoint set represents the test set.

At the end of the process, we get the training set consisting of all the lectures with publishing date prior to July 01, 2009, together with approximately half of the lectures after the aforementioned date, and the test set consisting of the rest of the lectures published after the aforementioned date.



(a)

(b)

Fig. 1: Distribution of edges of lecture-lecture graph with adjacency matrix of co-ocurrences between lectures. (a) for all lecture pairs $(x_i, x_j) : x_i \in G_1, x_j \in G_{22}$ (b) for all lectures pairs $(x_i, x_j) : x_i \in \{G_1 \cup G_{21}\}, x_j \in \{G_1 \cup G_{21}\}$

4 Challenge task definiton

Due to the nature of the problem, each of the tasks has its own merit: task 1 simulates new-user and new-item recommendation (cold start mode); task 2 simulates clickstream-based (implicit preference) recommendation.

4.1 The cold start task

The first task of the challenge is related to solving the so called *cold start problem*, commonly associated with pure collaborative filtering (CF) recommenders. Generally, *cold start* recommending quality should be measured through user satisfaction surveys and analysis. For the challenge, one needs a quantitative measure and a simulated cold start situation. In order to be able to score solutions, new video lectures are those that entered the site more recently, but for which there is already some viewing information available.

In this task, we assume that the user has seen one of the lectures which are characterized by the earlier times of entering the site (*old* lectures). As a solution for this task a ranked list of lectures from the *new* lectures set, is to be recommended after viewing some of the *old* lectures. The length of the recommended list is fixed at 30 lectures. Overall score for the submission/solution is based on the mean average R-precision score ($MARp$) (explained in Section 5).

Solution for the task 1 is based on ranking of lectures according to withheld lecture co-viewing frequencies in descending order. Suppose, the co-viewing frequencies (CVS) for some old lecture id_1 to new lectures $\{id_2, id_3, id_4, id_5\}$ are:

$$CVS(id_1, id_2) = 12, CVS(id_1, id_3) = 2,$$

$$CVS(id_1, id_4) = 43, CVS(id_1, id_5) = 3,$$

then we construct solution ranked list for old-lecture id_1:

$$id_1 : id_4, id_2, id_5, id_3.$$

4.2 Pooled lecture viewing sequences task

In task 2 contestants are asked to recommend a ranked list of ten lectures that should be recommended after viewing a set of three lectures. In contrast to the task 1, this is the situation close to typical recommendation scenario (submission and evaluation for the task 2). Solution for the task 2 is based on ranking of lectures according to frequencies in withheld pooled lecture viewing sequences in descending order. Test lectures from the task 1 are in this case not included into training pooled sequences, but can be a part of the ranked solution list for the task 2.

Suppose, there is a pooled lecture viewing sequences:

$$\{id_1, id_4, id_5\} \to id_6 : 5, id_3 : 4, id_7 : 2, id_2 : 1,$$

then we construct solution ranked list for triplet $\{id_1, id_4, id_5\}$:

$$\{id_1, id_4, id_5\} \to id_6, id_3, id_7, id_2.$$

5 Challenge evaluation function

Taking into account relative scarcity of items available for learning, recommending and evaluation (esp. in case of cold start task), we have defined an R-precision variants of standard evaluation measures in information retrieval $p@k$ and MAP. The overall score of the submission is mean value over all queries R (recommended lists r) given in the test sets:

$$MARp = \frac{1}{|R|} \sum_{r \in R} AvgRp(r)$$

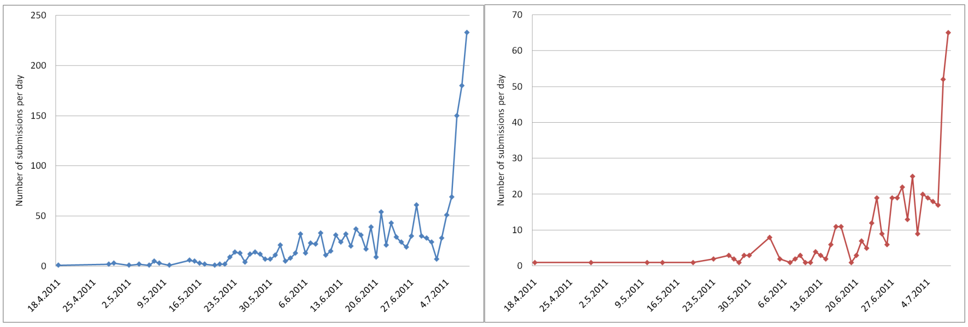
Average R-precision score - $AvgRp(r)$ for a single recommended ranked list r is defined as:

$$AvgRp(r) = \sum_{z \in Z} \frac{Rp@z(r)}{|Z|}$$

where $Rp@z(r)$ is R-precision at some cut-off length $z \in Z$. $Rp@z(r)$ is defined as the ratio of number of retrieved relevant items and relevant items at the particular cut-off z of the list:

$$Rp@z(r) = \frac{|relevant \cap retrived|_z}{|relevant|_z} = \frac{|relevant \cap retrived|_z}{min(m, z)}$$

Number of relevant items at cut-off length z is defined as $min(m, z)$, where m is the total number of relevant items. When $m \leq z$, number of relevant items at z is m, while for other situations it is limited to top z relevant items from the (real) solution ranked list s. A special situation happens when there are more equally relevant item$s$ at the same rank (ties) at the cut-off length of the s list. In that case, any of these items are treated as relevant (true positive) in calculating $Rp@z(r)$. For the task 1, cut-off lengths z for the calculation of $MARp$ are $z \in \{5, 10, 15, 20, 25, 30\}$. For the task 2, cut-off lengths z for the calculation of $MARp$ are $z \in \{5, 10\}$.



(a) (b)

Fig. 2: Number of submissions per days for (a) task 1 (b) task 2

Why average R-precision?

We have introduced R-precision because it is more apt to our situation: it adjusts to the size of the set of relevant documents. Typically, in information retrieval tasks one has to filter and rank from a large pool of both relevant and irrelevant items. This is not the case with the simulated cold start situation of this challenge. As an example, if there were only 4 items (lectures) in the whole collection relevant to the particular query, a perfect recommender system would score 1, measured by $Rp@10$, whereas its $p@10$ would be only 0.4. Using this measure for our application makes more sense, as the number of relevant items can vary from 1 to above 30, and in such situations $Rp@z$ expresses the quality of retrieval more fairly at some predefined retrieval (cut-off) length, than $p@z$. The reason why we use $AvgRp(r)$ over set of different $Rp@z$, is that through the averaging we can also take into account ranking and at the same time improve the ability to differentiate between similar solutions (recommenders).

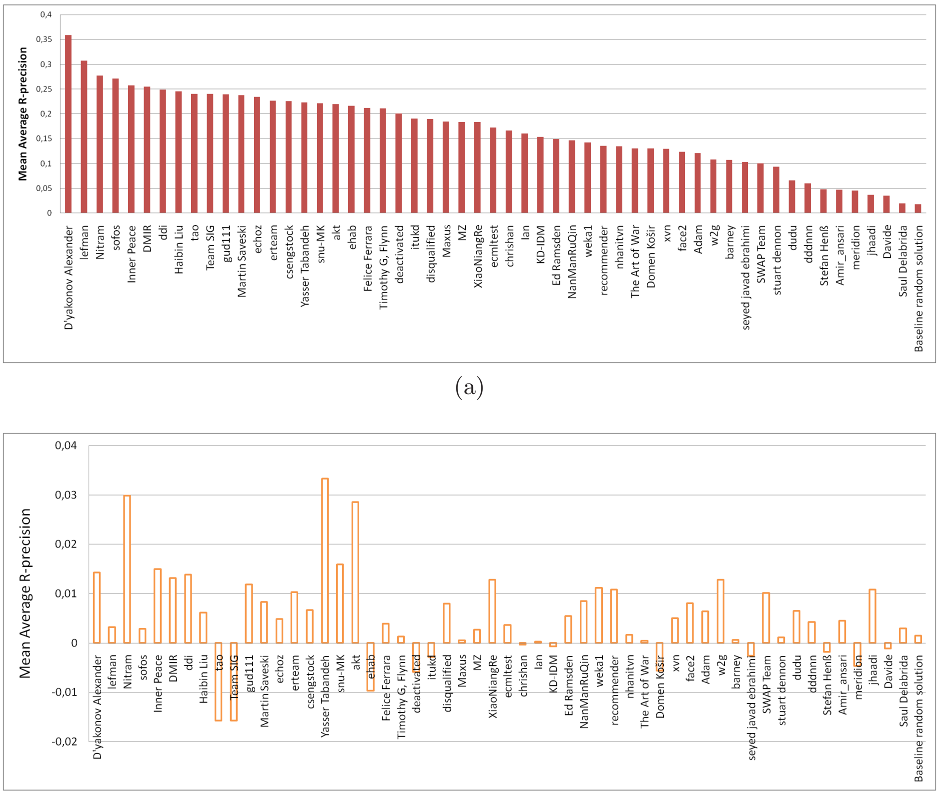
We have also considered MAP (mean average precision) measure, which is the closest to the proposed measure. However, MAP does not take into account absolute ranking positions of recommended items since permutations of relevant or true positive items in recommended list do not affect MAP score.

Normalized discounted cumulative gain ($NDCG$) [16, 17] takes into account that relevant documents are more useful when appearing earliear in a recommendation list. It is the most common measure used for ranking the results of the search list in information retrieval. This measure has also been used in other challenges where the main task was to learn ranking [14, 15].

If ranking order is not to be so strict for the top-n item recommendations [18], the ”granularity” of ranking can be relaxed. This is the main reason why we are using $MARp$ measure instead of the $NDCG$. Proposed measure $MARp$ takes into account absolute ranking positions with granularity of five items. This granularity was chosen after studying the ranking-recall influence on recommender system evaluation.

6 Challenge submissions results

ECML-PKDD 2011 Discovery Challenge started on 18th of April and ended on 8th of July 2011. The competition attracted significant number of participants: 303 teams



(a)

(b)

Fig. 3: (a) The *MARp* scores of final submissions for task 1. (b) Difference between *MARp* preliminary score and the *MARp* final score for task 1.

with 346 members, with 62/22 active teams per task. More than 2000 submissions were sent and best approaches outperformed baseline solution several times.

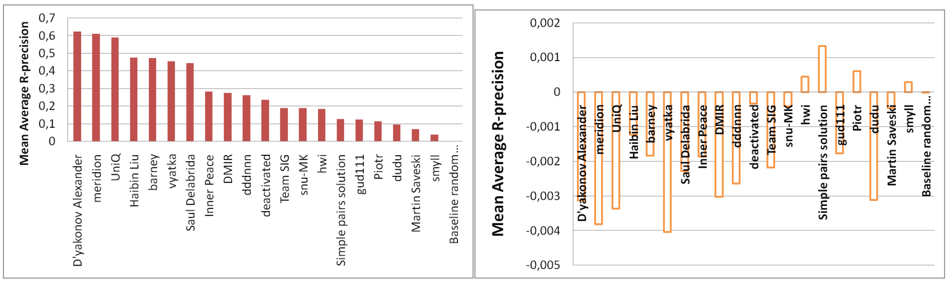
Winners of the challenge for task 1 are:

1. D'yakonov Alexander, Faculty of Computational Mathematics and Cybernetics, Moscow State University (Username: "D'yakonov Alexander")
2. Eleftherios Spyromitros-Xioufis and Emmannouela Stachtiari, Department of Informatics, Aristotle University of Thessaloniki (Username: "lefman")
3. Martin Možina, Faculty of Computer and Information Science, University of Ljubljana, Slovenia (Username: "Nitram")

Winners of the challenge for task 2 are:

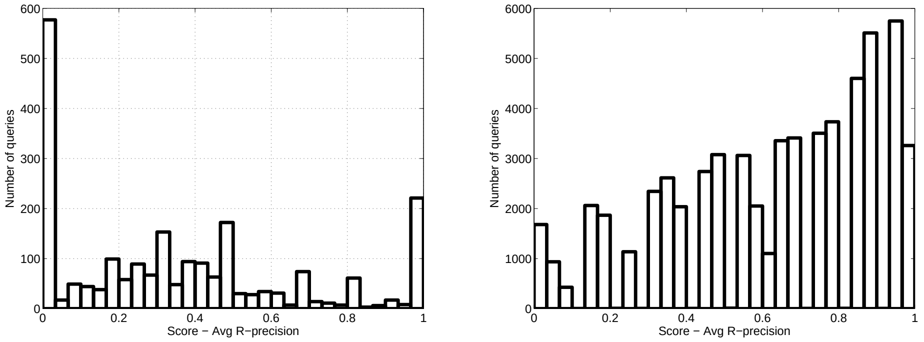
1. D'yakonov Alexander, Faculty of Computational Mathematics and Cybernetics, Moscow State University (Username: "D'yakonov Alexander")
2. Javier Kreiner, University of Trento, Italy (Username: "meridion")
3. Vladimir Nikulin, Department of Mathematics, The University of Queensland, Australia (Username: "UniQ")

The final scores, for the teams that scored better than the random recommender, are presented in the Figures 3 and 4, for each of the tasks respectivelly. The scores are



(a) (b)

Fig. 4: (a) the *MARp* scores of final submissions for task 2. (b) Difference between *MARp* preliminary score and the *MARp* final score for task 2.



(a) (b)

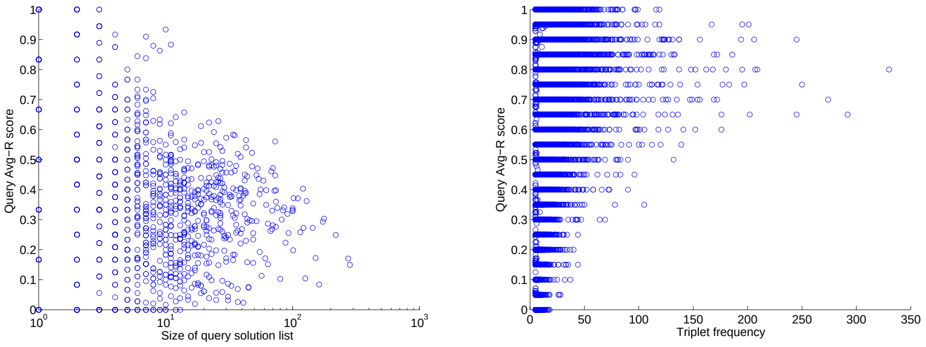
Fig. 5: Number of queries in the solution vs. *AvgRp* scores for the winning entries (a) task 1, (b) task 2

.

accompanied with the graphs of differences between preliminary *MARp* score on the leaderboard set and the final *MARp* score on the test set. For the task 1, from Figure 3, we can conclude that majority of the teams had positive difference scores, which may suggest overtraining. To the contrast, the majority of the teams had negative difference scores in tasks 2 (see Figure 4).

The distributions of the average R-precision over queries for the winning entry on each of the tasks are presented in Figure 5. Difference in distributions between the tasks reflects also the difference in the approaches used: while for the first task main features for solving the problem are constructed from lecture content and metadata similarity, for the second task only co-viewing information is utilized. We have also noted that these distributions are qualitatively very similar between first three positioned entries on each of the tasks, reflecting general similarity in approaches of different teams.

Dependence of query average R-precision score on the size of the solution list for the task 1 is presented in Figure 6 (graph on the left). On average, query score just slightly diminishes with the increase of the solution list. To the contrast, dependence of the query average R-precision score to the triplet frequency, for the task 2 (graph on the right in Figure 6) shows that on average the quality of result for the query is proportional to the triplet frequency.



(a)

(b)

Fig. 6: (a) Dependence of individual query *AvgRp* score on the size of the query solution list (number of relevant lectures in the solution list); (b) Dependence of individual query *AvgRp* score on the triplet frequency in the test set (reflecting popularity and co-viewing connectedness of the lectures in the triplet)

.

6.1 Methods used and discussion of approaches

The teams approached task 1 using quite different learning techniques with the primary effort focussed on feature engineering and optimization. Almost all of the participants have utilized all the lecture content related data (lecture taxonomy, event tree, types of lectures, descriptions, etc.), differing however slightly in definitions of the similarity of any two lectures. Important with respect to the overall score was the process of filling the missing values for the lectures that lack some of the content related data. Winning solutions used more sophisticated approach of filling lecture content and meta-data features' missing values using lecture co-viewing information (weighted CVS feature vector expansion [19], query expansion[20]) - thus utilizing collaborative information to ”enrich” content-based features.

Table 2 gives a summary of the feature engineering approaches and learning methods used in solving challenge tasks.

7 Conclusion

In the last couple of years, a number of challenges was organized in the field of recommendation problems. Most of them were focussed on prediction problems related to large scale explicit or implicit user preference matrices, in some cases combined with (mostly obfuscated) user, item and/or context related information. ECML-PKDD 2011 Discovery Challenge differed from this mainstream through two aspects: (i) instead of user preferences, only item to item preference information is available in the shape of the co-viewing frequencies graph; (ii) a rich and explicit description of lectures is available in the form of structured and unstructured text. On both tasks participants have obtained significantly higher *MARp* values than set by the baseline solutions.

Table 2: Approaches in solving challenge tasks

Solution (Track/pos)	**Feature engineering approach**	**Model learning approach**
1/1	combined similarity vector; uses feature expansion with CVS graph weighting; uses temporal transform for final similarity indices (LENKOR methodology)	optimization of weights of the linear model by the coodinate descent
1/2	textual based features synthesis; *tf-idf* based; use "query" expansion for missing terms; use temporal information (co-existence similarity)	cosine-similarity based *k-nn*, fitting to the optimal *k*; train the model utilizing "temporal" split in the training set
1/3	meta-data into categorical features (do not use in the model: lecture viewing, lecture description, slides' content); use co-viewing information to expand the content based rank prediction	content based linear regression for learning rank; utlize stochastic gradient descent to learn parameters of the linear model; learn hyperparameters from leaderboard submissions
2/1	two level normalization of pooled sequence vectors from the training set constructed from "pairs" and "singles" of triplets from the training set (LENKOR methodology) (do not use in the model: content and other lecture meta-data)	optimization of weights of the linear model by the coodinate descent
2/2	forming conditional probabilities for RHS lectures of triplets based on triplet lecture training co-viewing; no use of content based data	probabilistic model; entropy "like" scoring formulation; greedy grid based search fitting of coefficients in the scoring function
2/3	use singles', pairs' and couples' frequencies and define score updating	resampling based combination of the individual scoring functions; propose gradient based matrix factorization and recommendation model
Other track 1 approaches	tf-idf for different content/meta-data; Jaccard similarity based on LDA for topics	SVM for co-viewing prediction (binary classification, regression and ranking); random walk model using CVS graph (Katz and RF based), ordinary linear regression
Other track 2 approaches	using pairs and single lecture co-viewing data to construct test triplet RHS solution frequency vectors; hybridization of content similarity and co-viewing data	frequent item-set formulation - support/confidence based scoring and ranking

The analysis of the results shows that the most important part of a successful solution was careful feature engineering. Definition of the similarity scoring function capable of capturing content, context and temporal information turned out to be crucial for the success in the cold start (task 1) competition. Task 2, pooled sequence completion problem, was easier to solve and both approaches and results of the participants were mutualy much more similar. Rather unexpectedly, content related information was not used in ranking lectures to be viewed in succession to test set triplets. Most of the participants have also reported about the complexity/scaling of their solutions.

Our opinion is that the results of the challenge could be quite useful for constructing a new recommendation system for the VideoLectures.Net. In particular, there are several approaches that could significantly improve recommendation quality of new lectures at the site, with modest consumption of additional computational resources. Using lecture co-viewing frequency information instead of original preferences information in the form of click-streams should be studied in more detail, in order to understand the implications of this transformation on the personalized recommendation quality from the user's perspective.

Acknowledgements

The Discovery Challenge 2011 has been supported by the EU collaborative project e-LICO (GA 231519). The organizers of the Challenge are grateful to the Center for Knowledge Transfer in Information Technologies of the Jožef Stefan Institute and Viidea Ltd for the data of the VideoLectures.Net site, and TunedIT for the professional support in conducting the competition. Finally, we want to thank all the active participants of the challenge for their effort in the challenge and willingness to share their solutions and experience through the contributions in this workshop.

References

1. S.Rendle, K.Tso-Sutter, W.Huijsen, C.Freudenthaler, Z.Gantner, C.Wartena, R.Brussee and M. Wibbels: Report on State of the Art Recommender Algorithms (Update). My-Media public deliverable D4.1.2., (2011).
2. G.Adomavicius and A.Tuzhilin Towards 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)** (2005).
3. M.Montaner, B.Lopez and J.L. de la Rosa: A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review, **19**, (2003), 285-330.
4. G.Salton: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison Wesley, (1989).
5. R.Baeza-Yates, B.Ribeiro-Neto: Modern Information Retrieval. Addison Wesley, (1999).
6. W.Hill, L. Stead, M. Rosenstein and G.Furnas: Recommending and Evaluating Choices in Virtual Community of Use. Proc. Conf. Human Factors in Computing Systems, (1995).
7. P.Resnick, N.Iakovou, M.Sushak, P.Bergstrom and J.Riedl: GroupLens: An Open Architecture for Collaborative Filtering of Netnews Proc. Computer Supported Cooperative Work Conf., (1994).
8. U.Shardanand and P.Maes: Social Information Filtering: Algorithms for Automating 'word of Mouth' Proc. Conf. Human Factors in Computing Systems, (1995).
9. C.Boutilier , R.S.Zemel and B.Marlin: Active Collaborative Filtering. In Proc. of the Nineteenth Annual Conference on Uncertainty in Artificial Intelligence, (2003).
10. A.Schein, A.Popescul, L.Ungar, and D.Pennock: Generative models for coldstart recommendations. In Proceedings of the 2001 SIGIR Workshop on Recommender Systems, (2001).
11. M.Balabanovic and Y.Shoham: Fab: Content-based, collaborative recommendation. Communications of the ACM, **40(3)**, (1997).
12. J. Basilico and T. Hofmann: Unifying collaborative and content-based filtering. In Proceedings of the Twenty-First International Conference on Machine Learning, pages 65-72, New York, NY, USA, ACM Press, (2004).
13. R.Burke: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, **12(4)**, pp 331-370, (2002).
14. O. Chapelle, Y. Chang: Yahoo! Learning to Rank Challenge Overview, JMLR: Workshop and Conference Proceedings 14, pp 1-24, (2011).

15. Internet Mathematics 2009 contest: Limited Liability Company, http://imat2009.yandex.ru/academic/mathematic/2009/en/.
16. K. Jarvelin, J. Kekalainen: Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems 20(4), pp 422-446 (2002).
17. B. Croft, D. Metzler, and T. Strohman: Search Engines: Information Retrieval in Practice. Addison Wesley, (2009).
18. A. Turpin, W. Hersh: Why batch and user evaluations do not give the same results, In Proceedings of the 24th Annual ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, pp 17-24, (2001).
19. A. D'yakonov: Two Recommendation Algorithms Based on Deformed Linear Combinations. In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 21-27 (2011).
20. E. Spyromitros-Xioufis, E. Stachtiari, G. Tsoumakas, and I. Vlahavas: A Hybrid Approach for Cold-start Recommendations of Videolectures. In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 29-39, (2011).
21. M. Možina, A. Sadikov, and I. Bratko: Recommending VideoLectures with Linear Regression. In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 41-49, (2011).
22. J. A. Kreiner and E. Abraham: Recommender system based on purely probabilistic model from pooled sequence statistics. In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 51-57, (2011).
23. V. Nikulin: OpenStudy: Recommendations of the Following Ten Lectures After Viewing a Set of Three Given Lectures. In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 59-69, (2011).
24. H. Liu, S. Das, D. Lee, P. Mitra, C. Lee Giles: Using Co-views Information to Learn Lecture Recommendations. In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 71-82, (2011).
25. M. Chevalier, T. Dkaki, D. Dudognon, J. Mothe: IRIT at VLNetChallenge. In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 83-93, (2011).
26. L. Iaquinta and G. Semeraro: Lightweight Approach to the Cold Start Problem in the Video Lecture Recommendation. In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 95-101, (2011).
27. G. Capan, O. Yilmazel: Joint Features Regression for Cold-Start Recommendation on VideoLectures.Net In Proc. of ECML-PKDD 2011 Discovery Challenge Workshop, pp 103-109, (2011).
28. N. Antulov-Fantulin, M. Bošnjak, T. Šmuc, M. Jermol, M. Žnidaršič, M. Grčar, P. Keše, N. Lavrač: ECML/PKDD 2011 - Discovery challenge: VideoLectures.Net Recommender System Challenge, `http://lis.irb.hr/challenge/`