Proceedings of the First International Workshop on ADVANCED ANALYTICS AND LEARNING ON TEMPORAL DATA AALTD 2015

Workshop co-located with The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2015)

September 11, 2015. Porto, Portugal

Edited by

Ahlame Douzal-Chouakria
José A. Vilar
Pierre-François Marteau
Ann E. Maharaj
Andrés M. Alonso
Edoardo Otranto
Irina Nicolae

CEUR Workshop Proceedings
Volume 1425, 2015

Proceedings of AALTD 2015

First International Workshop on "Advanced Analytics and Learning on Temporal Data"

Porto, Portugal
September 11, 2015

Volume Editors

Ahlame Douzal-Chouakria
LIG-AMA, Université Joseph Fourier
Bâtiment Centre Equation 4, Allé de la Palestine à Gìres
UFR IM2AG, BP 53, F-38041 Grenoble Cedex 9, France
E:mail: Ahlame.Douzal@imag.fr

José A. Vilar Fernández
MODES, Departamento de Matemáticas, Universidade da Coruña
Facultade de Informática, Campus de Elviña, s/n, 15071 A Coruña, Spain
E:mail: jose.vilarf@udc.es

Pierre-François Marteau
IRISA, ENSIBS, Université de Bretagne Sud
Campus de Tohannic, BP 573, 56017 Vannes cedex, France
E:mail: pierre-francois.marteau@univ-ubs.fr

Ann E. Maharaj
Department of Econometrics and Business Statistics, Monash University
Caulfield Campus Building H, Level 5, Room 86
900 Dandenong Road, Caulfield East, Victoria 3145, Australia
E:mail: ann.maharaj@monash.edu

Andrés M. Alonso Fernández
Departamento de Estadística, Universidad Carlos III de Madrid
C/ Madrid, 126, 28903 Getafe (Madrid) Spain
E:mail: andres.alonso@uc3m.es

Edoardo Otranto
Department of Cognitive Sciences, Educational and Cultural Studies
University of Messina
Via Concezione, n.6, 98121 Messina, Italy
E:mail: eotranto@unime.it

Maria-Irina Nicolae
Jean Monnet University, Hubert Curien Lab
E105, 18 rue du Professeur Benoît Lauras, Saint-Etienne, France
E:mail: irina.nicolae@imag.fr

Published by the editors on CEUR-WS.ORG

ISSN 1613-0073 Volume 1425 http://CEUR-WS.org/Vol-1425

September, 2015

Preface

We are honored to welcome you to the 1st International Workshop on Advanced Analytics and Learning on Temporal Data (AALTD), which is held in Porto, Portugal, on September 11th, 2015, co-located with The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2015).

The aim of this workshop is to bring together researchers and experts in machine learning, data mining, pattern analysis and statistics to share their challenging issues and advance researches on temporal data analysis. Analysis and learning from temporal data cover a wide scope of tasks including learning metrics, learning representations, unsupervised feature extraction, clustering and classification.

This volume contains the conference program, an abstract of the invited keynote and the set of regular papers accepted to be presented at the conference. Each of the submitted papers was reviewed by at least two independent reviewers, leading to the selection of seventeen papers accepted for presentation and inclusion into the program and these proceedings. The contributions are given by the alphabetical order, by surname. An index of authors can be also found at the end of this book.

The keynote given by Gustavo Camps-Valls on "Capturing Time-Structures in Earth Observation Data with Gaussian Processes" focuses on machine learning models based on Gaussian processes which help to monitor land, oceans, and atmosphere through the estimation of climate and biophysical variables.

The accepted papers spanned from innovative ideas on analytic of temporal data, including promising new approaches and covering both practical and theoretical issues. Classification of time series, estimation of graphical models for temporal data, extraction of patterns from audio streams, searching causal models from longitudinal data and symbolic representation of time series are only a sample of the analyzed topics. To introduce the reader, a brief presentation of the problems addressed at each of papers is given below.

A novel approach to analyze the evolution of a disease incidence is presented by Andrade-Pacheco *et al.* The method is based on Gaussian processes and allows to study the effect of the time series components individually and hence to isolate the relevant components and explore short term variations of the disease. Bailly *et al* introduce a series classification procedure based on extracting local features using the Scale-Invariant Feature Transform (SIFT) framework and then building a global representation of the series using the Bag-of-Words (BoW) approach. Billard *et al* propose to highlight the main structure of multiple sets of multivariate time series by using principal component analysis where the standard correlation structure is replaced by lagged cross-autocorrelation. The symbolic representation of time series SAXO is formalized as a hierarchical coclustering approach by Bondu *et al*, evaluating also its compactness in terms of coding length. A framework to learn an efficient temporal metric by combining

several basic metrics for a robust kNN is introduced by Do *et al.* Dupont and Marteau introduce a sparse version of Dynamic Time Warping (DTW), called coarse-DTW, and develop an efficient algorithm (Bubble) to sparse regular time series. By coupling both mechanisms, the nearest-neighbor classification of time series can be performed much faster.

Gallicchio *et al* study the balance assessment of elderly people with time series acquired with a Wii Balance Board. A novel technique to estimate the well-known Berg Balance Scale is proposed by using a Reservoir Computing network. Gibberd and Nelson address the estimation of graphical models when data change over time. Specifically, two extensions of the Gaussian graphical model (GGM) are introduced and empirically examined. Extraction of patterns from audio data streams is investigated by Hardy *et al* considering a symbolization procedure combined with the use of different pattern mining methods. Jain and Spiegel propose a strategy to classify time series consisting of transforming the series into a dissimilarity representation and then applying PCA followed by an SVM. Krempl addresses the problem of forecasting the density at spatio-temporal coordinates in the future from a sample of pre-fixed instances observed at different positions in the feature space and at different times in the past. Two different approaches using spatio-temporal kernel density estimation are proposed. A fuzzy C-medoids algorithm to cluster time series based on comparing estimated quantile autocovariance functions is presented by Lafuente-Rego and Vilar.

A new algorithm for discovering causal models from longitudinal data is developed by Rahmadi *et al.* The method performs structure search over Structural Equation Models (SEMs) by maximizing model scores in terms of data fit and complexity, showing robustness for finite samples. Salperwyck *et al* introduce a clustering technique for time series based on maximizing an inter-inertia criterion inside parallelized decision trees. An anomaly detection approach for temporal graph data based on an iterative tensor decomposition and masking procedure is presented by Sapienza *et al.* Soheily-Khah *et al* perform an experimental comparison of several progressive and iterative methods for averaging time series under dynamic time warping. Finally, Sorokin extends the factored gated restricted Boltzmann machine model by adding discriminative component, thus enabling it to be used as a classifier and specifically to extract translational motion from two related images.

In sum, we think that all these contributions will provide valuable feedback and motivation to advance research on analysis and learning from temporal data. It is planned that extended versions of the accepted papers will be published in a special volume of Lecture Notes of Artificial Intelligence (LNAI).

We wish to thank the ECML PKDD council members for giving us the opportunity to hold the AALTD workshop within the framework of the ECML PKDD Conference and the members of the local organizing committee for their support. Also our gratitude to several colleagues that helped us with the organi-

zation of the workshop, particularly Saeid Soheily (Université Grenoble Alpes, France).

The organizers of the AALTD conference gratefully thank the financial support of the "Programme d'Investissements d'Avenir" of the French government through the IKATS Project as well as the support received from LIG-AMA, IRISA, MODES, Université Joseph Fourier and Universidade da Coruña.

Last but not least, we wish to thank the contributing authors for the high quality works and all members of the Reviewing Committee for their invaluable assistance in the selection process. All of them have significantly contributed to the success of AALTD 2105.

We sincerely hope that the workshop participants have a great and fruitful time at the conference.

September, 2015

Ahlame Douzal-Chouakria
José A. Vilar
Pierre-François Marteau
Ann E. Maharaj
Andrés M. Alonso
Edoardo Otranto
Irina Nicolae

Program Committee

Ahlame Douzal-Chouakria, Université Grenoble Alpes, France
José Antonio Vilar Fernández, University of A Coruña, Spain
Pierre-François Marteau, IRISA, Université de Bretagne-Sud, France
Ann Maharaj, Monash University, Australia
Andrés M. Alonso, Universidad Carlos III de Madrid, Spain
Edoardo Otranto, University of Messina, Italy

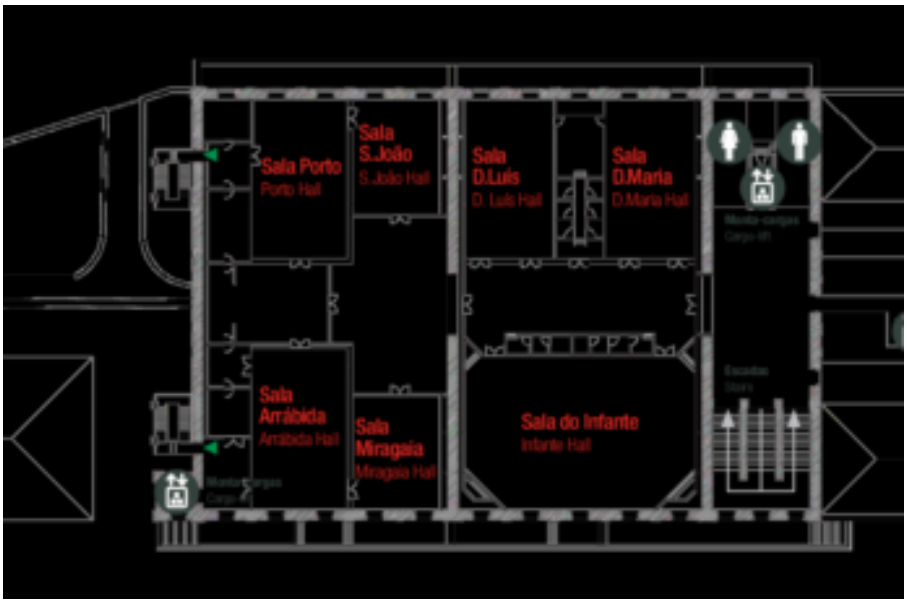
Reviewing Committee

Massih-Reza Amini, Université Grenoble Alpes, France
Manuele Bicego, University of Verona, Italy
Gianluca Bontempi, MLG, ULB University, Belgium
Antoine Cornuéjols, LRI, AgroParisTech, France
Pierpaolo D'Urso, University La Sapienza, Italy
Patrick Gallinari, LIp. 43, UPMC, France
Eric Gaussier, Université Grenoble Alpes, France
Christian Hennig, Department of Statistical Science, London's Global Univ, UK
Frank Höeppner, Ostfalia University of Applied Sciences, Germany
Paul Honeine, ICD, Université de Troyes, France
Vincent Lemaire, Orange Lab, France
Manuel García Magariños, University of A Coruña, Spain
Mohamed Nadif, LIPADE, Université Paris Descartes, France
François Petitjean, Monash University, Australia
Fabrice Rossi, SAMM, Université Paris 1, France
Allan Tucker, Brunel University, UK

Conference programme schedule

Conference venue and some instructions

Within the framework of the ECML PKDD 2015 Conference, the AALTD Workshop will take place from 15:00 to 18:00 on Friday, September 11, at the Alfândega Congress Centre, Rua Nova de Alfândega, 4050-430 Porto. The invited talk and the oral communications will take place at the room *Porto*, on the second floor of the Congress Centre (see partial site plan below).



The lecture room *Porto* will be equipped with a PC and a computer projector, which will be used for presentations. Before the session starts, presenters must provide to the session chair with the files for the presentation in PDF (Acrobat) or PPT (Powerpoint) format on a USB memory stick. Alternatively, the talks can be submitted by e-mail to José A. Vilar (jose.vilarf@udc.es) prior to the start of the conference. Time planned for each presentation is fifteen minutes with five additional minutes for discussion.

With regard to the poster session, the authors will be responsible for placing the posters in the poster panel, which should be carried out well in advance. The maximum size of the poster is A0.

Schedule

Invited talk	Chair: Ahlame Douzal
15:00 - 15:30	Capturing Time-Structures in Earth Observation Data with Gaussian Processes *Gustavo Camps-Valls*
Oral communication	Chair: José A. Vilar
15:30 - 15:50	Time Series Classification in Dissimilarity Spaces *Brijnesh J. Jain, Stephan Spiegel*
Poster session	Chairs: Maria-Irina Nicolae, Saeid Soheily
15:50 - 16:15	See table on next page.
16:00 - 16:15	COFFEE BREAK
Oral communications	Chair: Pierre-François Marteau
16:15 - 16:35	Fuzzy Clustering of Series Using Quantile Autocovariances *Borja R. Lafuente-Rego, José A. Vilar*
16:35 - 16:55	Temporal Density Extrapolation *Georg Krempl*
16:55 - 17:15	Coarse-DTW: Exploiting Sparsity in Gesture Time Series *Marc Dupont, Pierre-François Marteau*
17:15 - 17:35	Symbolic Representation of Time Series: A Hierarchical Coclustering Formalization *Alexis Bondu, Marc Boullé, Antoine Cornuéjols*
17:35 - 17:55	Monitoring Short Term Changes of Malaria Incidence in Uganda with Gaussian Processes *Ricardo Andrade Pacheco, Martin Mubangizi, John Quinn, Neil Lawrence*

Communications in poster session

P1 Bag-of-Temporal-SIFT-Words for Time Series Classification
Adeline Bailly, Simon Malinowski, Romain Tavenard, Thomas Guyet, Lætitia Chapel

P2 An Exploratory Analysis of Multiple Multivariate Time Series
Lynne Billard, Ahlame Douzal-Chouakria, Seyed Yaser Samadi

P3 Temporal and Frequential Metric Learning for Time Series kNN Classification
Cao-Tri Do, Ahlame Douzal-Chouakria, Sylvain Marie, Michele Rombaut

P4 Preliminary Experimental Analysis of Reservoir Computing Approach for Balance Assessment
Claudio Gallicchio, Alessio Micheli, Luca Pedrelli, Federico Vozzi, Oberdan Parodi

P5 Estimating Dynamic Graphical Models from Multivariate Time-series Data
Alexander J. Gibberd, James D.B. Nelson

P6 Sequential Pattern Mining on Multimedia Data
Corentin Hardy, Laurent Amsaleg, Guillaume Gravier, Simon Malinowski, René Quiniou

P7 Causality on Longitudinal Data: Stable Specification Search in Constrained Structural Equation Modeling
Ridho Rahmadi, Perry Groot, Marianne Heins, Hans Knoop, Tom Heskes

P8 CourboSpark: Decision Tree for Time-series on Spark
Christophe Salperwyck, Simon Maby, Jérôme Cubillé, Matthieu Lagacherie

P9 Anomaly Detection in Temporal Graph Data: An Iterative Tensor Decomposition and Masking Approach
Anna Sapienza, André Panisson, Joseph Wu, Lætitia Gauvin, Ciro Cattuto

P10 Progressive and Iterative Approaches for Time Series Averaging
Saeid Soheily-Khah, Ahlame Douzal-Chouakria, Eric Gaussier

P11 Classification Factored Gated Restricted Boltzmann Machine
Ivan Sorokin

Table of Contents

Capturing Time-structures in Earth Observation Data with Gaussian Processes
G. Camps-Valls 1
Monitoring Short Term Changes of Malaria Incidence in Uganda with Gaussian Processes
R. Andrade-Pacheco, M. Mubangizi, J. Quinn, N. Lawrence 3
Bag-of-Temporal-SIFT-Words for Time Series Classification
A. Bailly, S. Malinowski, R. Tavenard, T. Guyet, L. Chapel 11
An Exploratory Analysis of Multiple Multivariate Time Series
L. Billard, A. Douzal-Chouakria, S. Yaser Samadi 19
Symbolic Representation of Time Series: a Hierarchical Coclustering Formalization
A. Bondu, M. Boullé, A. Cornuéjols 27
Temporal and Frequential Metric Learning for Time Series kNN classification
C.-T. Do, A. Douzal-Chouakria, S. Marié, M. Rombaut 39
Coarse-DTW: Exploiting Sparsity in Gesture Time Series
M. Dupont, P.-F. Marteau 47
Preliminary Experimental Analysis of Reservoir Computing Approach for Balance Assessment
C. Gallicchio, A. Micheli, L. Pedrelli, F. Vozzi, O. Parodi 57
Estimating Dynamic Graphical Models from Multivariate Time-Series Data
A.J. Gibberd, J.D.B. Nelson 63
Sequential Pattern Mining on Multimedia Data
C. Hardy, L. Amsaleg, G. Gravier, S. Malinowski, R. Quiniou 71
Time Series Classification in Dissimilarity Spaces
B.J. Jain, S. Spiegel 79
Temporal Density Extrapolation
G. Krempl 85
Fuzzy Clustering of Series Using Quantile Autocovariances
B. Lafuente-Rego, J.A. Vilar 93
Causality on Longitudinal Data: Stable Specification Search in Constrained Structural Equation Modeling
R. Rahmadi, P. Groot, M. Heins, H. Knoop, T. Heskes 101
CourboSpark: Decision Tree for Time-series on Spark
C Salperwyck, S. Maby, J. Cubillé, M. Lagacherie 109
Anomaly Detection in Temporal Graph Data: An Iterative Tensor Decomposition and Masking Approach
A. Sapienza, A. Panisson, J. Wu, L. Gauvin, C. Cattuto 117

Progressive and Iterative Approaches for Time Series Averaging
S. Soheily-Khah, A. Douzal-Chouakria, E. Gaussier 123
Classification Factored Gated Restricted Boltzmann Machine
I. Sorokin 131

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Capturing Time-structures in Earth Observation Data with Gaussian Processes

Gustavo Camps-Valls
Department of Electrical Engineering, Universitat de València, Spain

Abstract. In this talk I will summarize our experience in the last years on developing algorithms in the interplay between Physics and Statistical Inference to analyze Earth Observation satellite data. Some of them are currently adopted by ESA and EUMETSAT. I will pay attention to machine learning models that help to monitor land, oceans, and atmosphere through the estimation of climate and biophysical variables. In particular, I will focus on Gaussian Processes, which provide an adequate framework to design models with high prediction accuracy and able to cope with uncertainties, deal with heteroscedastic noise and particular time-structures, to encode physical knowledge about the problem, and to attain self-explanatory models. The theoretical developments will be guided by the challenging problems of estimating biophysical parameters at both local and global planetary scales.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Monitoring Short Term Changes of Malaria Incidence in Uganda with Gaussian Processes

Ricardo Andrade-Pacheco[1], Martin Mubangizi[2], John Quinn[2,3], and Neil Lawrence[1]

[1] University of Sheffield, Department of Computer Science, UK
[2] Makerere University, College of Computing and Information Science, Uganda
[3] UN Global Pulse, Pulse Lab Kampala, Uganda

Abstract. A method to monitor communicable diseases based on health records is proposed. The method is applied to health facility records of malaria incidence in Uganda. This disease represents a threat for approximately 3.3 billion people around the globe. We use Gaussian processes with vector-valued kernels to analyze time series components individually. This method allows not only removing the effect of specific components, but studying the components of interest with more detail. The short term variations of an infection are divided into four cyclical phases. Under this novel approach, the evolution of a disease incidence can be easily analyzed and compared between different districts. The graphical tool provided can help quick response planning and resources allocation.

Keywords: Gaussian processes, malaria, kernel functions, time series.

1 Introduction

More than a century after discovering its transmission mechanism, malaria has been successfully eradicated from different regions of world [15]. However, it is still endemic in 100 countries and represents a threat for 3.3 billion people approximately [20]. In Uganda, malaria is among the leading causes of morbidity and mortality [19]. Different types of interventions can be carried on to prevent and treat malaria [20]. Their success depend on how well the disease can be anticipated and how fast the population reacts to it. In this regard, mathematical modelling can be a strong ally for decision-making and health services planning. Spatiotemporal modelling for mapping and prediction of infection dynamics is a challenging problem. First of all, because of the costs and difficulties of gathering data. Second, because of the challenges of developing a sound theoretical model that agrees with the data observed.

The Health Management Information System (HMIS) operated by the Uganda Ministry of Health provides weekly records of the number of patients treated for malaria in different hospitals across the country. Unfortunately, the number of reporting hospitals is not consistent across time. This variation is prone to create artificial trends in the observed data. Hence, the underreporting effect has to be estimated to be removed.

A common approach for time series analysis is to decompose the observed variation into specific patterns such as *trends*, *cyclic effects* or *irregular fluctuations* [4, 3, 7]. Gaussian process (GP) models are a natural approach for analyzing

functions that represent time series. GPs provide a robust framework for non-parametric probabilistic modelling [18]. The use of covariance kernels enable to analyse non-linear patterns by embedding an inference problem into an abstract space with a *convenient structure*[14]. By combining different covariance kernels (via additions, multiplications or convolutions) into a single one, a GP is able to describe more complex functions. Each of the individual kernels contributes by encoding a specific set of properties or pattern of the resulting function [5].

We propose a monitoring system for communicable diseases based on Gaussian processes. This methodology is able to isolate the relevant components of the time series and study the short term variations of the disease. The output of this system is a graphical tool that discretizes the disease progress into four phases of simple interpretation.

2 Background

Say we are interested in learning the functional relation, between inputs and output, based on a set of observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. GP models introduce an additional *latent variable* $f_{\mathbf{x}}$, whose covariance kernel K is a function of the input values. Usually, y_i is considered a distorted version of the latent variable.

To deal with multiple outputs, GP models resort to generalizations of kernel functions to the vector-valued case [1]. In time series literature, vector-valued functions are commonly treated in the family of VAR models [12], while in geostatistics literature *co-Kriging* generalizations are used [8, 11]. These approaches are equivalent. Let $h_{\mathbf{x}} = (f_{\mathbf{x}}^1, \dots, f_{\mathbf{x}}^d)^\top$ be a vector-valued GP, its corresponding covariance matrix is given by

$$\big[\operatorname{cov}(h_{\mathbf{x}}, h_{\mathbf{z}})_{ij}\big] = \big[\operatorname{cov}(f_{\mathbf{x}}^i, f_{\mathbf{z}}^j)\big]. \tag{1}$$

The diagonal elements of the correlation matrix $\big[\operatorname{cov}(h_{\mathbf{x}}, h_{\mathbf{z}})_{ii}\big]$ are just the covariance functions of the real-valued GP elements. The non-diagonal elements represent the *cross-covariance functions* between components [9, 10, 2].

3 Method Used

Suppose we have data generated from the combination of two independent signals (see Figure 1a). Usually, not only we are not able to observe the signals separately, but the combined signal they yield is corrupted by noise in the data collected (see Figure 1b). For the sake of this example, suppose that the two signals of the example represent a long term trend (the smooth signal) and a seasonal component (the sinusoidal signal). For an observer, the oscillations of the seasonal component masks the behaviour of the long term trend. At some point, however, the observer might want to know whether the trend is increasing or decreasing. Similarly, there might be interest in studying only the seasonal

component isolated from the trend. For example, in economics and finance, business recession and expansion periods are determined by studying the cyclic component of a set of indicators [16]. The cyclic component tells if an indicator is above or below the trend, and its differences tell if it is increasing or decreasing.

We propose a similar approach for monitoring disease incidence time series, but in our case, we will use a non-parametric approach. To extract the original signals, the observed data can be modelled using a GP with a combination of kernels, say exponentiated quadratics, one having a shorter lengthscale than the other. Figures 1c and 1d shows a model of the combined and independent signals. We also use a vector-valued GP to model directly the derivative of the time series, rather than using simple differences of the observed trend. As a result, we are able to provide uncertainty estimates about the speed of the changes around the trend. Our approach is based on modelling linear functionals of an underlying GP [13]. If $h_{\mathbf{x}} = (f_{\mathbf{x}}, \partial f_{\mathbf{x}}/\partial x_i)^\top$, its corresponding kernel is defined as

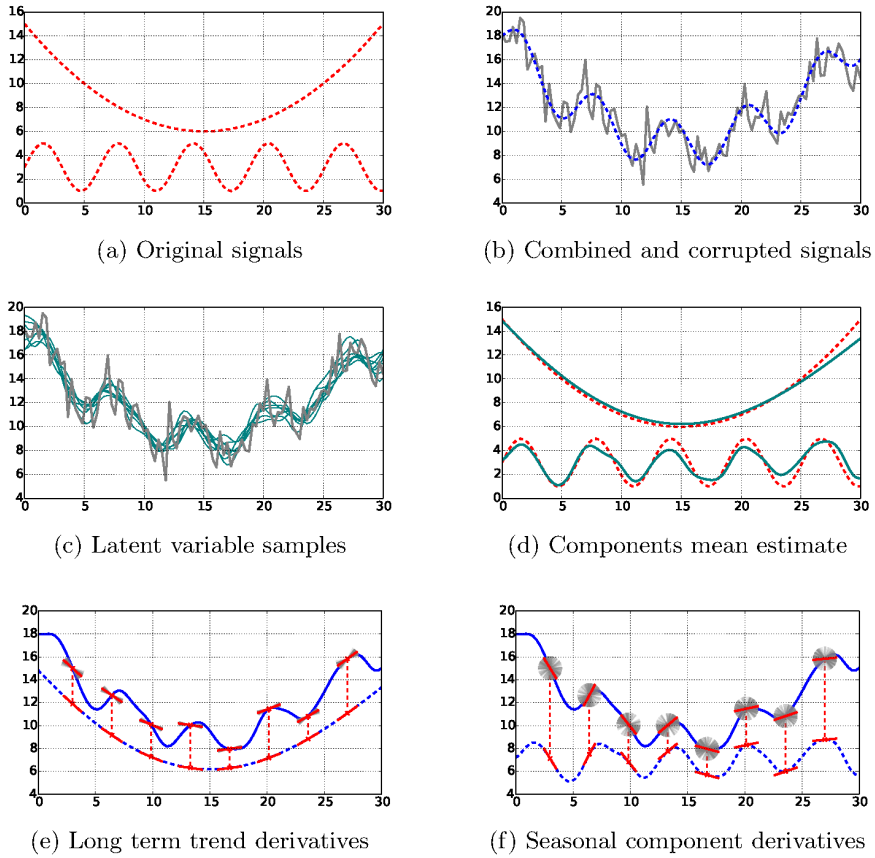
$$\Gamma(\mathbf{x}_i, \mathbf{x}_j) = \begin{bmatrix} K(\mathbf{x}_i, \mathbf{x}_j) & \frac{\partial}{\partial x_j} K(\mathbf{x}_i, \mathbf{x}_j) \\ \frac{\partial}{\partial x_i} K(\mathbf{x}_i, \mathbf{x}_j) & \frac{\partial^2}{\partial x_i x_j} K(\mathbf{x}_i, \mathbf{x}_j) \end{bmatrix}. \tag{2}$$

In most multi-output problems, observations of the different outputs are needed to learn their relation. Here, the relation between $f_{\mathbf{x}}$ and its derivative is known beforehand through the derivative of K. Thus $\partial f_{\mathbf{x}}/\partial x_i$ can be learnt by relying entirely on $f_{\mathbf{x}}$. For the signals described above, Figures 1e and 1f show the corresponding derivatives computed using a kernel of the form of (2). The derivatives of the long term trend are computed with high confidence, while the derivatives of the seasonal component have more uncertainty. The last is due to the magnitude of the seasonal component relative to the noise magnitude.

4 Uganda Case

In this exposition we focus on Kabarole district, but provide a snapshot of the monitoring system for all the country. Our base assumption about the infection process of malaria is that it evolves with some degree of smoothness across time. Smooth functions can be represented by a kernel such that the closer the observations in the input space, the more similar values of the output. The Matérn kernel family satisfies this condition, as it defines dependence through the distance between points with some exponential decay [18]. Different members of this family encode different degrees of smoothness, being the limit case the exponentiated quadratic kernel or RBF, which is infinitely differentiable. To illustrate our method we will use an RBF kernel. Results with (rougher) Matérn kernels do not differ much when used instead.

Despite malaria is a disease influenced by environmental factors like temperature or water availability, we could not observe a seasonal effect in HMIS data [6]. If that was the case, the model could be improved incorporating a periodic kernel in the covariance structure. Yet, the model fit can be improved if a second RBF kernel is added. In this case, one kernel has a short lengthscale and



(a) Original signals

(b) Combined and corrupted signals

(c) Latent variable samples

(d) Components mean estimate

(e) Long term trend derivatives

(f) Seasonal component derivatives

Fig. 1: Series decomposition. Panel (a) shows two independent signals. Panel (b) shows the combination of both signals (dashed line) and a distorted signal after adding some noise (solid line). Panel (c) shows latent variable samples representing the combined signal (thin lines). Panel (d) compares the mean estimate of each component (solid line) with the original signals (dashed line). Panels (e) and (f) show the components derivatives. Tangent lines to the individual components are shown in red. The solid blue lines represent the mean estimate of the composed signal. The gray lines are random realizations of process derivative. For comparison, the estimates of the individual signals (dashed lines) are shown below the composed signal.

therefore represents short term variations, while the other represents long term changes.

An important factor to consider about HMIS data is that the number of health facilities is highly variable. See Figure 2a. This variation is prone to create artificial trends in the incidence of malaria reported. Such trends can be removed by incorporating a linear kernel that describes the relation between reporting facilities and malaria cases. Unlike the RBF kernels mentioned above,

which take time as input, the linear kernel takes the number of health facilities as input. In Table 1, we present a comparison of the model predictive performance, when using different kernels, based on the leave-one-out predictive probabilities [17]. The best predictive performance is achieved when considering short and long term changes and a correction for misreporting facilities.

Table 1: Comparison of LOO-CV log predictive probabilities, when using different kernels. The subindex ℓ refers to the lengthscale of the kernel (measured in years).

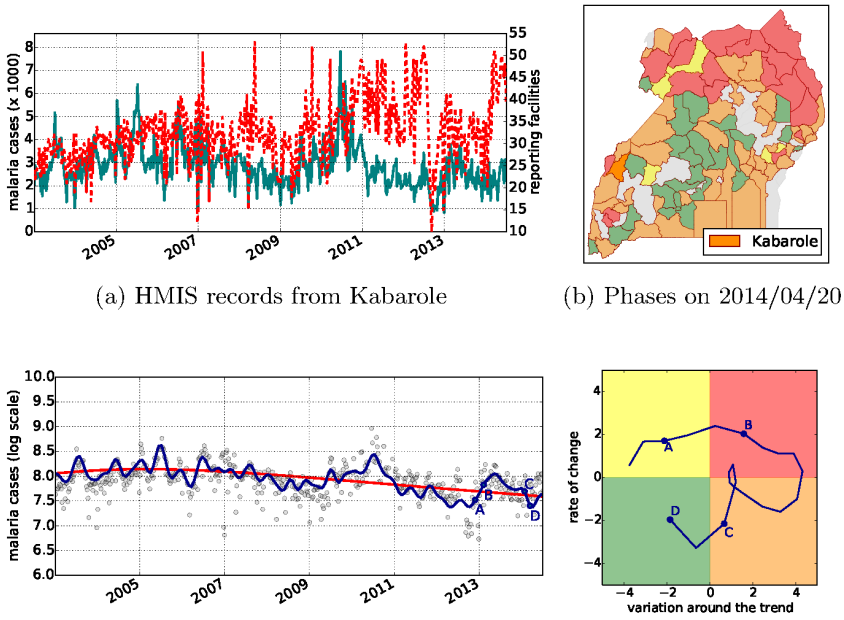
Kernel	LOO-CV (log)
$RBF_{\ell=0.64}$	-40.54
$RBF_{\ell=0.14} + RBF_{\ell=10}$	-16.26
$RBF_{\ell=0.12} + RBF_{\ell=10} + Linear$	41.21

Figure 2c shows the trend and short term component of the number of malaria cases. Variations of a disease incidence around its trend represent short term changes in the population health. Outbreak detection and control of non-endemic diseases take place in this time frame. For some endemic diseases, this variation can be associated to seasonal factors [6]. Quick response actions, such as distribution of medicine and allocation of patients to health centres, have to take place in this time regime to be effective. The short term variations can be classified in four phases as shown in Figure 2d (values are standardized). The upper left quadrant represents an incidence below the trend, but increasing; the upper right quadrant represents an incidence above the trend and expanding; the bottom right quadrant represents an incidence above the trend, but decreasing; and the bottom left quadrant represents an incidence below the trend and decreasing.

This tracking system of short term variations is independent of the order of the disease counts, and can be used to monitor the infection progress in different districts. It is easy to identify districts where the disease is being controlled or where the infection is progressing at an unusual rate. Figure 2b shows the monitoring system on the whole country. Those districts where the variation coefficient of both the process and its derivative are less than 1 (meaning a weak signal vs noise) were left in gray color.

5 Final Remarks

We have proposed a disease monitor based on vector-valued Gaussian processes. Our approach is able to account for uncertainty in both the level of each component and the direction of change. The simplicity for doing inference with this



(a) HMIS records from Kabarole

(b) Phases on 2014/04/20

(c) Trend and short term variations

(d) Cycle phases

Fig. 2: Malaria incidence tracker in Uganda. Panel (a) compares the number of malaria cases (solid line) and the number of reporting health facilities (dashed line). Panel (b) shows the disease phase in each district. The colors are assigned according to the quadrants in panel (d). Panel (c) shows the long term trend (dashed line) and the short term variations (solid line). Gray bullets represent the observed records. Panel (d) shows a tracking system of the short term variations. The bullets A-D in panels (c) and (d) correspond to the same time points.

model is not compromised by the use of a vector-valued approach. The model can be benefited if spatial information is available and encoded in the kernel function. Further research is needed to explore the benefits of this model in practice. We expect that an analysis from this perspective can add situational awareness and contribute to interventions planning and resources allocation when facing infectious diseases.

Acknowledgements

Ricardo Andrade-Pacheco is supported by CONACYT and SEP scholarships.

References

1. M. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

2. L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. Multi-output learning via spectral filtering. *Machine Learning*, 87(3):259–301, 2012.
3. M. Baxter and R. G. King. Measuring business cycles: approximate band-pass filters for economic time series. *Review of economics and statistics*, 81(4):575–593, 1999.
4. W. P. Cleveland and G. C. Tiao. Decomposition of seasonal time series: A model for the census X-11 program. *Journal of the American statistical Association*, 71(355):581–587, 1976.
5. N. Durrande, J. Hensman, M. Rattray, and N. D. Lawrence. Gaussian process models for periodicity detection. *arXiv preprint arXiv:1303.7090*, 2013.
6. S. I. Hay, R. W. Snow, and D. J. Rogers. From predicting mosquito habitat to malaria seasons using remotely sensed data: practice, problems and perspectives. *Parasitology Today*, 14(8):306–313, 1998.
7. A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.
8. G. Matheron. Pour une analyse krigeante de donnés régionalisées. Technical report, École des Mines de Paris, Fontainebleau, France, 1982.
9. C. A. Micchelli and M. Pontil. Kernels for multi-task learning. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2004.
10. C. A. Micchelli and M. Pontil. On learning vector–valued functions. *Neural Computation*, 17:177–204, 2005.
11. D. E. Myers. Matrix formulation of co-Kriging. *Journal of the International Association for Mathematical Geology*, 14(3):249–257, 1982.
12. H. Quenouille. *The analysis of multiple time-series*. Griffin's statistical monographs & courses. Griffin, 1957.
13. S. Särkkä. Linear operators and stochastic partial differential equations in Gaussian process regression. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 151–158. Springer, 2011.
14. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, U.K., 2004.
15. P. I. Trigg and A. V. Kondrachine. Commentary: malaria control in the 1990s. *Bulletin of the World Health Organization*, 76(1):11, 1998.
16. F. van Ruth, B. Schouten, and R. Wekker. The statistics Netherlands business cycle tracer. Methodological aspects; concept, cycle computation and indicator selection. Technical report, Statistics Netherlands, 2005.
17. A. Vehtari, V. Tolvanen, T. Mononen, and O. Winther. Bayesian leave-one-out cross-validation approximations for Gaussian latent variable models. *arXiv preprint arXiv:1412.7461*, 2014.
18. C. K. I. Williams and C. E. Rasmussen. *Gaussian processes for Machine Learning*. MIT Press, 2006.
19. World Health Organization. World health statistics 2015. Technical report, WHO Press, Geneva, 2015.
20. World Health Organization and others. World malaria report 2014. Technical report, WHO Press, Geneva, 2014.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Bag-of-Temporal-SIFT-Words for Time Series Classification

Adeline Bailly[1], Simon Malinowski[2], Romain Tavenard[1],
Thomas Guyet[3], and Lætitia Chapel[4]
[1] Université de Rennes 2, IRISA, LETG-Rennes COSTEL, Rennes, France
[2] Université de Rennes 1, IRISA, Rennes, France
[3] Agrocampus Ouest, IRISA, Rennes, France
[4] Université de Bretagne Sud, Vannes ; IRISA, Rennes, France

Abstract. Time series classification is an application of particular interest with the increase of data to monitor. Classical techniques for time series classification rely on point-to-point distances. Recently, Bag-of-Words approaches have been used in this context. Words are quantized versions of simple features extracted from sliding windows. The SIFT framework has proved efficient for image classification. In this paper, we design a time series classification scheme that builds on the SIFT framework adapted to time series to feed a Bag-of-Words. Experimental results show competitive performance with respect to classical techniques.

Keywords: time series classification, Bag-of-Words, SIFT, BoTSW

1 Introduction

Classification of time series has received an important amount of interest over the past years due to many real-life applications, such as environmental modeling, speech recognition. A wide range of algorithms have been proposed to solve this problem. One simple classifier is the k-nearest-neighbor (kNN), which is usually combined with Euclidean Distance (ED) or Dynamic Time Warping (DTW) [11]. Such techniques compute similarity between time series based on point-to-point comparisons, which is often not appropriate. Classification techniques based on higher level structures are most of the time faster, while being at least as accurate as DTW-based classifiers. Hence, various works have investigated the extraction of local and global features in time series. Among these works, the Bag-of-Words (BoW) approach (also called bag-of-features) has been considered for time series classification. BoW is a very common technique in text mining, information retrieval and content-based image retrieval because of its simplicity and performance. For these reasons, it has been adapted to time series data in some recent works [1, 2, 9, 12, 14]. Different kinds of features based on simple statistics have been used to create the words.

In the context of image retrieval and classification, scale-invariant descriptors have proved their efficiency. Particularly, the Scale-Invariant Feature Transform (SIFT) framework has led to widely used descriptors [10]. These descriptors are scale and rotation invariant while being robust to noise. We build on this framework to design a BoW approach for time series classification where the

words correspond to the description of local gradients around keypoints, that are first extracted from the time series. This approach can be seen as an adaptation of the SIFT framework to time series.

This paper is organized as follows. Section 2 summarizes related work, Section 3 describes the proposed Bag-of-Temporal-SIFT-Words (BoTSW) method, and Section 4 reports experimental results. Finally, Section 5 concludes and discusses future work.

2 Related work

Our approach for time series classification builds on two well-known methods in computer vision: local features are extracted from time series using a SIFT-based approach and a global representation of time series is built using Bag-of-Words. This section first introduces state-of-the-art methods in time series classification, then presents standard approaches for extracting features in the image classification context and finally lists previous works that make use of such approaches for time series classification.

Data mining community has, for long, investigated the field of time series classification. Early works focus on the use of dedicated metrics to assess similarity between time series. In [11], Ratanamahatana and Keogh compare Dynamic Time Warping to Euclidean Distance when used with a simple kNN classifier. While the former benefits from its robustness to temporal distortions to achieve high efficiency, ED is known to have much lower computational cost. Cuturi [4] shows that DTW fails at precisely quantifying dissimilarity between non-matching sequences. He introduces Global Alignment Kernel that takes into account all possible alignments to produce a reliable dissimilarity metric to be used with kernel methods such as Support Vector Machines (SVM). Douzal and Amblard [5] investigate the use of time series metrics for classification trees.

So as to efficiently classify images, those first have to be described accurately. Both local and global descriptions have been proposed by the computer vision community. For long, the most powerful local feature for images was SIFT [10] that describes detected keypoints in the image using the gradients in the regions surrounding those points. Building on this, Sivic and Zisserman [13] suggested to compare video frames using standard text mining approaches in which documents are represented by word histograms, known as Bag-of-Words (BoW). To do so, authors map the 128-dimensional space of SIFT features to a codebook of few thousand words using vector quantization. VLAD (Vector of Locally Aggregated Descriptors) [6] are global features that build upon local ones in the same spirit as BoW. Instead of storing counts for each word in the dictionary, VLAD preserves residuals to build a fine-grain global representation.

Inspired by text mining, information retrieval and computer vision communities, recent works have investigated the use of Bag-of-Words for time series classification [1, 2, 9, 12, 14]. These works are based on two main operations: converting time series into Bag-of-Words (a histogram representing the occurrence of words), and building a classifier upon this BoW representation. Usually, clas-

sical techniques are used for the classification step: random forests, SVM, neural networks, kNN. In the following, we focus on explaining how the conversion of time series into BoW is performed in the literature. In [2], local features such as mean, variance, extremum values are computed on sliding windows. These features are then quantized into words using a codebook learned by a class probability estimate distribution. In [14], discrete wavelet coefficients are extracted on sliding windows and then quantized into words using k-means. In [9, 12], words are constructed using the SAX representation [8] of time series. SAX symbols are extracted from time series and histograms of n-grams of these symbols are computed. In [1], multivariate time series are transformed into a feature matrix, whose rows are feature vectors containing a time index, the values and the gradient of time series at this time index (on all dimensions). Random samples of this matrix are given to decision trees whose leaves are seen as words. A histogram of words is output when the different trees are learned. Rather than computing features on sliding windows, authors of [15] first extract keypoints from time series. These keypoints are selected using the Differences-of-Gaussians (DoG) framework, well-known in the image community, that can be adapted to one-dimensional signals. Keypoints are then described by scale-invariant features that describe the shapes of the extremum surrounding keypoints. In [3], extraction and description of time series keypoints in a SIFT-like framework is used to reduce the complexity of Dynamic Time Warping: features are used to match anchor points from two different time series and prune the search space when finding the optimal path in the DTW computation.

In this paper, we design a time series classification technique based on the extraction and the description of keypoints using a SIFT framework adapted to time series. The description of keypoints is quantized using a k-means algorithm to create a codebook of words and classification of time series is performed with a linear SVM fed with normalized histograms of words.

3 Bag-of-Temporal-SIFT-Words (BoTSW) method

The proposed method is adapted from the SIFT framework [10] widely used for image classification. It is based on three main steps : (i) detection of keypoints (scale-space extrema) in time series, (ii) description of these keypoints by gradient magnitude at a specific scale, and (iii) representation of time series by a BoW, words corresponding to quantized version of the description of keypoints. These steps are depicted in Fig. 1 and detailed below.

Following the SIFT framework, keypoints in time series correspond to local extrema both in terms of scale and location. These scale-space extrema are identified using a DoG function, which establishes a list of scale-invariant keypoints. Let $L(t, \sigma)$ be the convolution ($*$) of a Gaussian function $G(t, \sigma)$ of width σ with a time series $S(t)$:

$$L(t, \sigma) = G(t, \sigma) * S(t).$$

DoG is obtained by subtracting two time series filtered at consecutive scales:

$$D(t, \sigma) = L(t, k_{sc}\sigma) - L(t, \sigma),$$

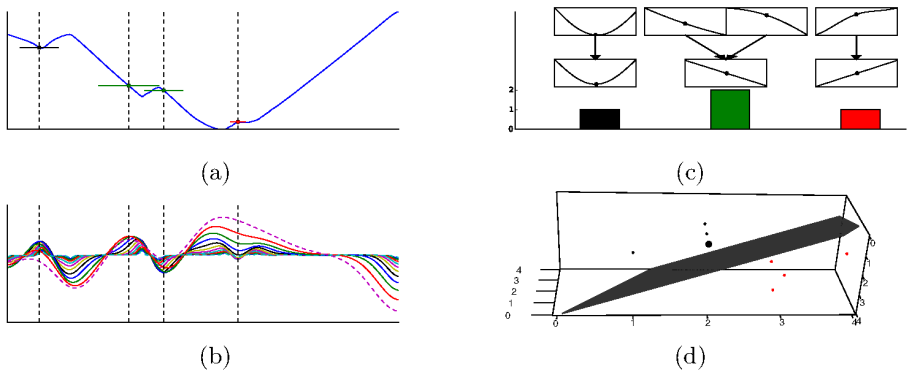


Fig. 1: Approach overview : (a) A time series and its extracted keypoints (the length of the horizontal lines for each point is proportional to the keypoint scale), (b) The Difference-of-Gaussians, computed at different scales, on which the keypoint extraction is built, (c) Keypoint description is based on the time series filtered at the scale at which the keypoint is extracted. Descriptors are quantized into words, and time series are represented by a histogram of words occurrence. For the sake of readability, neighborhoods are shown here instead of features. (d) These histograms are given to a classifier (linear SVM here) that learns boundaries between the different classes. The bigger dot here represents the description of the time series in (a), whose coordinates are $(1, 2, 1)$. Best viewed in color.

where k_{sc} controls the scale ratio between two consecutive scales. A keypoint is detected at time index t and scale j if it corresponds to an extremum of $D(t, k_{sc}^j \sigma)$ in both time and scale (8 neighbors : 2 at the same scale, and 6 in adjacent scales) If a point is higher (or lower) than all of its neighbors, it is considered as an extremum in the scale-space domain and hence a keypoint of S.

Next step in our process is the description of keypoints. A keypoint at (t, j) is described by gradient magnitudes of $L(\cdot, k_{sc}^j \sigma)$ around t. n_b blocks of size a are selected around the keypoint. Gradients are computed at each point of each block and weighted using a Gaussian window of standard deviation $\frac{a \times n_b}{2}$ so that points that are farther in time from the detected keypoint have lower influence. Then, each block is described by storing separately the sums of magnitude of positive and negative gradients. Resulting feature vector is of dimension $2 \times n_b$.

Features are then quantized using a k-means algorithm to obtain a codebook of k words. Words represent different kinds of local behavior in the time series. For a given time series, each feature vector is assigned to the closest word of the codebook. The number of occurrences of each word in a time series is computed. The BoTSW representation of a time series is the normalized histogram (*i.e.*

Dataset	BoTSW + linear SVM			BoTSW + 1NN			ED + 1NN	DTW + 1NN
	k	n_b	ER	k	n_b	ER	ER	ER
50words	512	16	0.363	1024	16	0.400	0.369	**0.310**
Adiac	512	16	0.614	128	16	0.642	**0.389**	0.396
Beef	128	10	0.400	128	16	**0.300**	0.467	0.500
CBF	64	6	0.058	64	14	0.049	0.148	**0.003**
Coffee	256	4	**0.000**	64	12	**0.000**	0.250	0.179
ECG200	256	16	**0.110**	64	12	0.160	0.120	0.230
Face (all)	1024	8	0.218	512	16	0.239	0.286	**0.192**
Face (four)	128	12	**0.000**	128	6	0.046	0.216	0.170
Fish	512	16	**0.069**	512	14	0.149	0.217	0.167
Gun-Point	256	4	0.080	256	10	**0.067**	0.087	0.093
Lightning-2	16	16	0.361	512	16	0.410	0.246	**0.131**
Lightning-7	512	14	0.384	512	14	0.480	0.425	**0.274**
Olive Oil	256	4	**0.100**	512	2	**0.100**	0.133	0.133
OSU Leaf	1024	10	**0.182**	1024	16	0.248	0.483	0.409
Swedish Leaf	1024	16	**0.152**	512	10	0.229	0.213	0.210
Synthetic Control	512	14	0.043	64	8	0.093	0.120	**0.007**
Trace	128	10	0.010	64	12	**0.000**	0.240	**0.000**
Two Patterns	1024	16	0.002	1024	16	0.009	0.090	**0.000**
Wafer	512	12	**0.001**	512	12	**0.001**	0.005	0.020
Yoga	1024	16	**0.150**	512	6	0.230	0.170	0.164

Table 1: Classification error rates (best performance is written as bold text).

frequency vector) of word occurrences. These histograms are then passed to a classifier to learn how to discriminate classes from this BoTSW description.

4 Experiments and results

In this section, we investigate the impact of both the number of blocks n_b and the number of words k in the codebook (defined in Section 3) on classification error rates. Experiments are conducted on 20 datasets from the UCR repository [7]. We set all parameters of BoTSW but n_b and k as follows : $\sigma = 1.6$, $k_{sc} = 2^{1/3}$, $a = 8$. These values have shown to produce stable results. Parameters n_b and k vary inside the following sets : $\{2, 4, 6, 8, 10, 12, 14, 16\}$ and $\left\{2^i, \forall\ i \in \{2..10\}\right\}$ respectively. Codebooks are obtained *via* k-means quantization. Two classifiers are used to classify times series represented as BoTSW : a linear SVM or a 1NN classifier. Each dataset is composed of a train and a test set. For our approach, the best set of (k, n_b) parameters is selected by performing a leave-one-out cross-validation on the train set. This best set of parameters is then used to build the classifier on the train set and evaluate it on the test set. Experimental error rates (ER) are reported in Table 1, together with baseline scores publicly available at [7].

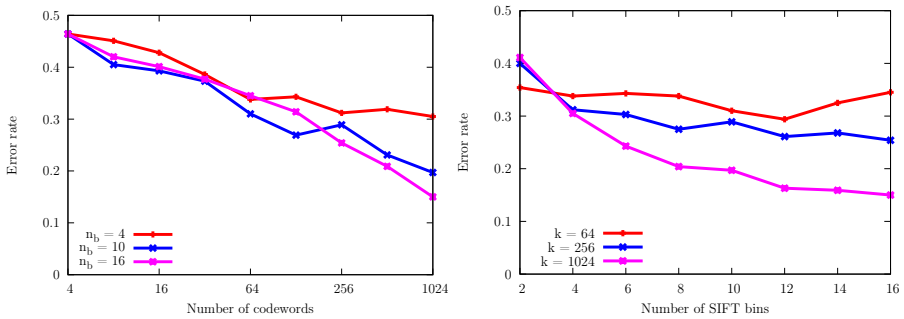


Fig. 2: Classification accuracy on dataset Yoga as a function of k and n_b.

	ED+ 1NN			DTW+ 1NN			TSBF[2]			SAX-VSM[12]			SMTS[1]			BoP[9]		
	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L	W	T	L
BoTSW+lin. SVM	18	0	2	11	0	9	8	0	12	9	2	9	7	0	13	14	0	6
BoTSW + 1NN	13	0	7	9	1	10	5	0	15	4	3	13	4	1	15	7	1	12

Table 2: Win-Tie-Lose (WTL) scores comparing BoTSW to state-of-the-art methods. For instance, BoTSW+linear SVM reaches better performance than ED+1NN on 18 datasets, and worse performance on 2 datasets.

BoTSW coupled with a linear SVM is better than both ED and DTW on 11 datasets. It is also better than BoTSW coupled with a 1NN classifier on 13 datasets. We also compared our approach with classical techniques for time series classification. We varied number of codewords k between 4 and 1024. Not surprisingly, cross-validation tends to select large codebooks that lead to more precise representation of time series by BoTSW. Fig. 2 shows undoubtedly that, for Yoga dataset, (left) the larger the codebook, the better the results and (right) the choice of the number n_b of blocks is less crucial as a wide range of values yield competitive classification performance.

Win-Tie-Lose scores (see Table 2) show that coupling BoTSW with a linear SVM reaches competitive performance with respect to the literature.

As it can be seen in Table 1, BoTSW is (by far) less efficient than both ED and DTW for dataset Adiac. As BoW representation maps keypoint descriptions into words, details are lost during this quantization step. Knowing that only very few keypoints are detected for these Adiac time series, we believe a more precise representation would help.

5 Conclusion

BoTSW transforms time series into histograms of quantized local features. Distinctiveness of the SIFT keypoints used with Bag-of-Words enables to efficiently and accurately classify time series, despite the fact that BoW representation

ignores temporal order. We believe classification performance could be further improved by taking time information into account and/or reducing the impact of quantization losses in our representation.

Acknowledgments

This work has been partly funded by ANR project ASTERIX (ANR-13-JS02-0005-01), Région Bretagne and CNES-TOSCA project VEGIDAR.

References

1. M. G. Baydogan and G. Runger. Learning a symbolic representation for multivariate time series classification. *DMKD*, 29(2):400–422, 2015.
2. M. G. Baydogan, G. Runger, and E. Tuv. A Bag-of-Features Framework to Classify Time Series. *IEEE PAMI*, 35(11):2796–2802, 2013.
3. K. S. Candan, R. Rossini, and M. L. Sapino. sDTW: Computing DTW Distances using Locally Relevant Constraints based on Salient Feature Alignments. *Proc. VLDB*, 5(11):1519–1530, 2012.
4. M. Cuturi. Fast global alignment kernels. In *Proc. ICML*, pages 929–936, 2011.
5. A. Douzal-Chouakria and C. Amblard. Classification trees for time series. *Elsevier Pattern Recognition*, 45(3):1076–1091, 2012.
6. H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, pages 3304–3311, 2010.
7. E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR Time Series Classification/Clustering Homepage, 2011. `www.cs.ucr.edu/~eamonn/time_series_data/`.
8. J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. ACM SIGMOD Workshop on Research Issues in DMKD*, pages 2–11, 2003.
9. J. Lin, R. Khade, and Y. Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *IJIS*, 39:287–315, 2012.
10. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
11. C. A. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. In *Proc. ACM SIGKDD Workshop on Mining Temporal and Sequential Data*, pages 22–25, 2004.
12. P. Senin and S. Malinchik. SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model. *Proc. ICDM*, pages 1175–1180, 2013.
13. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, pages 1470–1477, 2003.
14. J. Wang, P. Liu, M. F.H. She, S. Nahavandi, and A. Kouzani. Bag-of-words Representation for Biomedical Time Series Classification. *BSPC*, 8(6):634–644, 2013.
15. J. Xie and M. Beigi. A Scale-Invariant Local Descriptor for Event Recognition in 1D Sensor Signals. In *Proc. ICME*, pages 1226–1229, 2009.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

An Exploratory Analysis of Multiple Multivariate Time Series

Lynne Billard[1], Ahlame Douzal-Chouakria[2], and Seyed Yaser Samadi[3]
[1] Department of Statistics, University of Georgia
[2] Université Grenoble Alpes, CNRS - LIG/AMA, France
[3] Department of Mathematics, Southern Illinois University

Abstract. Our aim is to extend standard principal component analysis for non-time series data to explore and highlight the main structure of multiple sets of multivariate time series. To this end, standard variance-covariance matrices are generalized to lagged cross-autocorrelation matrices. The methodology produces principal component time series, which can be analysed in the usual way on a principal component plot, except that the plot also includes time as an additional dimension.

1 Introduction

Time series data are ubiquitous, arising throughout economics, meteorology, medicine, the basic sciences, even in some genetic microarrays, to name a few of the myriad fields of application. Multivariate time series are likewise prevalent. Our aim is to use principal components methods as an exploratory technique to find clusters of time series in a set of S multivariate time series. For example, in a collection of stock market time series, interest may center on whether some stocks, such as mining stocks, behave alike but differently from other stocks, such as pharmaceutical stocks.

A seminal paper in univariate time series clustering is that of Košmelj and Batagelj (1990), based on a dissimilarity measure. Since then several researchers have proposed other approaches (e.g. Caiado et al (2015), D'Urso and Maharaj (2009)). A comprehensive summary of clustering for univariate time series is in Liao (2005). Liao (2007) introduced a two-step procedure for multivariate series which transformed the observations into a single multivariate series. Most of these methods use dissimilarity functions or variations thereof. A summary of Liao (2005, 2007) along with more recent proposed methods is in Billard et al. (2015). Though a few authors specify a particular model structure, by and large, the dependence information inherent to time series observations is not used.

Dependencies in time series are measured through the autocorrelation (or, equivalently, the autocovariance) functions. In this work, we illustrate how these

dependencies can be used in a principal component analysis. This produces principal component time series, which in turn allows the projection of the original time series observations onto three dimensional principal component by time space. The basic methodology is outlined in Section2, and illustrated in Section 3.

2 Methodology

2.1 Cross-Autocorrelation functions for $S > 1$ series and $p > 1$ dimensions

Let $\mathbf{X}_{st} = \{(X_{stj}), j = 1, \ldots, p\}$, $t = 1, \ldots, N_s$, $s = 1, \ldots, S$, be a p-dimensional time series of length N_s, for each series s. For notational simplicity, assume $N_s = N$ for all s. Let us also assume the observations have been suitably differenced/transformed so that the data are stationary.

For a standard single univariate series time series where $S = 1$ and $p = 1$, it is well-known that the sample autocovariance function at lag k is (dropping the $s = S = 1$ and $j = p = 1$ subscripts)

$$\hat{\gamma}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (X_t - \bar{X})(X_{t+k} - \bar{X}), \quad k = 0, 1, \ldots, \quad \bar{X} = \frac{1}{N} \sum_{t=1}^{N} X_t, \quad (2.1)$$

and the sample autocorrelation function at lag k is $\hat{\rho}(k) = \hat{\gamma}(k)/\hat{\gamma}(0)$, $k = 0, 1, \ldots$.

These autocorrelation functions provide a measure of the time dependence between observations changes as their distance apart, lag k. They are used to identify the type of model and also to estimate model parameters. See, many of the basic texts on time series, e.g., Box et al. (2011); Brockwell and Davis (1991); Cryer and Chan (2008). Note that the divisor in Eq.(2.1) is N, rather than $N - k$. This ensures that the sample autocovariance matrix is non-negative definite.

For a single multivariate time series where $S = 1$ and $p \geq 1$, the cross-autocovariance function between variables (j, j') at lag k is the $p \times p$ matrix $\boldsymbol{\Gamma}(k)$ with elements estimated by

$$\hat{\gamma}_{jj'}(k) = \frac{1}{T} \sum_{t=1}^{T-k} (X_{tj} - \bar{X}_j)(X_{t+k,j'} - \bar{X}_{j'}), \quad k = 0, 1, \text{with } \bar{X}_j = \frac{1}{N} \sum_{t=1}^{N} X_{tj}, \quad (2.2)$$

and the cross-autocorrelation function between variables (j, j') at lag k is the $p \times p$ matrix, $\boldsymbol{\rho}(k)$, with elements $\{\rho_{jj'}(k), j, j' = 1, \ldots, p\}$ estimated by

$$\hat{\rho}_{jj'}(k) = \hat{\gamma}_{jj'}(k)/\{\hat{\gamma}_{jj}(0)\hat{\gamma}_{j'j'}(0)\}^{1/2}, \quad k = 0, 1, \ldots. \tag{2.3}$$

Unlike the autocorrelation function obtained from Eq.(2.1) with its single value at each lag k, Eq.(2.3) produces a $p \times p$ matrix at each lag k. The function Eq.(2.2) was first given by Whittle (1963) and shown to be nonsymmetric by Jones (1964). In general, $\rho_{jj'}(k) \neq \rho_{j'j}(k)$ for variables $j \neq j'$, except for $k = 0$, but $\boldsymbol{\rho}(k) = \boldsymbol{\rho}'(-k)$; see, e.g., Brockwell and Davis (1991).

When there are $S \geq 1$ series and $p \geq 1$ variables, the definition of Eqs.(2.2)-(2.3) can be extended to give a $p \times p$ sample cross-autocovariance function matrix between variables (j, j') at lag k, $\hat{\boldsymbol{\Gamma}}(k)$, with elements given by, for $j, j' = 1, \ldots, p$,

$$\hat{\gamma}_{jj'}(k) = \frac{1}{NS} \sum_{s=1}^{S} \sum_{t=1}^{N-k} (X_{stj} - \bar{X}_j)(X_{s,t+k,j'} - \bar{X}_{j'}), \; k = 0, 1, \tag{2.4}$$

$$\text{with } \bar{X}_j = \frac{1}{NS} \sum_{s=1}^{S} \sum_{t=1}^{N} X_{stj};$$

and the $p \times p$ sample cross-autocorrelation matrix at lag k, $\hat{\boldsymbol{\rho}}^{(1)}(k)$, has elements $\hat{\rho}_{jj'}(k)$, $j, j' = 1, \ldots, p$, obtained by substituting Eq.(2.4) into Eq.(2.3). This cross-autocovariance function in Eq.(2.4) is a measure of time dependence between observations k units apart for a given variable pair (j, j'), calculated across all S series. Notice, the sample means $\bar{X}_j$ in Eq.(2.4) are calculated across all NS observations.

An alternative approach is to calculate these sample means by series. In this case, the cross-autocovariance matrix $\hat{\boldsymbol{\Gamma}}(k)$ has elements estimated by, for $j, j' = 1, \ldots, p$, $s = 1, \ldots, S$,

$$\hat{\gamma}_{jj'}(k) = \frac{1}{NS} \sum_{s=1}^{S} \sum_{t=1}^{N-k} (X_{stj} - \bar{X}_{sj})(X_{s,t+k,j} - \bar{X}_{sj'}), \; k = 0, 1, \tag{2.5}$$

$$\text{with } \bar{X}_{sj} = \frac{1}{N} \sum_{t=1}^{N} X_{stj};$$

and the corresponding $p \times p$ cross-autocorrelation function matrix $\hat{\boldsymbol{\rho}}^{(2)}(k)$ has elements $\hat{\rho}_{jj'}(k)$ found by substituting Eq.(2.5) into Eq.(2.3).

Other model structures can be considered, which would provide other options for obtaining the relevant sample means. These include class structures, lag k structures, weighted series and/or weighted variable structures, and the like; see Billard et al. (2015).

2.2 Principal Components for Time Series

In a standard classical principal component analysis on a set of p-dimensional multivariate observations $\mathbf{X} = \{X_{ij}, i = 1, \ldots n,\ j = 1, \ldots, p\}$, each observation is projected into a corresponding ν^{th} order principal component, $PC_\nu(i)$, through the linear combination of the observation's variables,

$$PC_\nu(i) = w_{\nu 1} X_{i1} + \cdots + w_{\nu p} X_{ip}, \quad \nu = 1, \ldots, p, \tag{2.6}$$

where $\mathbf{w}_\nu = (w_{\nu 1}, \ldots, w_{\nu p})$ is the ν^{th} eigenvector of the correlation matrix $\boldsymbol{\rho}$ (or, equivalently for non-standardized observations, the variance-covariance matrix $\boldsymbol{\Sigma}$). The eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$, and $\sum_{\nu=1}^{p} \lambda_\nu = p$ (or, σ^2 for non-standardized data). A detailed description of this methodology for standard data can be found in any of the numerous texts on multivariate analysis, e.g., Joliffe (1986) and Johnson and Wichern (2007) for an applied approach, and Anderson (1984) for theoretical details.

For time series data, the correlation matrix $\boldsymbol{\rho}$ is replaced by the cross-autocorrelation matrix $\boldsymbol{\rho}(k)$, for a specific lag $k = 1, 2, \ldots$, and the ν^{th} order principal component of Eq.(2.6) becomes

$$PC_\nu(s,t) = w_{\nu 1} X_{s1t} + \cdots + w_{\nu p} X_{spt}, \quad \nu = 1, \ldots, p, \quad t = 1, \ldots, N, \ s = 1, \ldots, S. \tag{2.7}$$

The elements of $\boldsymbol{\rho}(k)$ can be estimated by $\hat{\rho}_{jj'}(k)$ from Eq.(2.4) or from Eq.(2.5) (or from other choices of model structure). The problem of non-positive definiteness, for lag $k > 0$, for the cross-autocorrelation matrix has been studied by Rousseeuw and Molenberghs (1993) and Jäckel (2002), with the recommendation that negative eigenvalues be re-set at zero.

3 Illustration

To illustrate, take a data set (<http://dss.ucar.edu/datasets/ds578.5>) where the observations are time series of monthly temperatures at $S = 14$ cities (weather stations) in China over the years 1923-88. In the present analysis, each month is taken to be a single variable corresponding to the twelve months (January, ..., December, respectively); hence, $p = 12$. Clearly, these variables are dependent as reflected in the cross-autocovariances when $j \neq j'$.

Let us limit the discussion to using the cross-autocorrelation functions at lag $k = 1$, evaluated from Eq.(2.4) and Eq.(2.3), and shown in Table 1. We obtain the corresponding eigenvalues and eigenvectors, and hence we calculate the principal components PC_ν, $\nu = 1, \ldots, p$, from Eq.(2.6). A plot of $PC_1 \times PC_2 \times$ time is

displayed in Figure 1, and that for $PC_1 \times PC_3 \times$ time is given in Figure 2. An interesting feature of these data highlighted by the methodology is that it is the $PC_1 \times PC_3$ pair that distinguishes more readily the city groupings. Figure 3 displays the $PC_1 \times PC_3$ values for all series and all times without tracking time (i.e., the 3-dimensional $PC_1 \times PC_3 \times$ time values are projected onto the $PC_1 \times PC_3$ plane). Hence, we are able to discriminate between cities.
Thus, we observe that cities 1-4 (Hailaer, HaErBin, MuDanJiang and ChangChun, respectively), color coded in black (and indicated by the symbol black $\circ$ and full lines ('lty=1')) have similar temperatures and are located in the north-eastern region of China. Cities 5-7 (TaiYuan, BeiJing, TianJin), identified by red ($\triangle$ and lines $- \cdot -$ ('lty=4')), are in the north, and have similar but different temperature trends than do those in the north-eastern region. Two (BeiJing and TianJin) are located close to sea-level, while the third (TaiYuan) is further south (and so might be expected to have higher temperatures) but its elevation is very high so decreasing its temperature patterns to be more in line with BeiJing and TianJin. Cities 8-11 (ChengDu, WuHan, ChangSha, HangZhou), green ($*$) with lines $\cdots$ ('lty=3'), are located in central regions with ChengDu further west but elevated. Finally, cities 12-14 (FuZhou, XiaMen, GuangZhou), blue ($\square$) with lines $- - -$ ('lty=8'), are in the southeast part of the country.

Pearson correlations between the variables X_j, $j = 1, \ldots, 12$, and the principal components PC_ν, $\nu = 1, \ldots, 12$, sand correlation circles (not shown) show that all months have an impact on PC_1 with the months of June, July and August having a slightly negative influence on PC_2. Plots for other $k \neq 1$ values give comparable results. Likewise, analyses using the cross-autocorrelations of Eq.(2.5) also produce similar conclusions.

4 Conclusion

The methodology has successfully identified cities with similar temperature trends, which trends *a priori* could not have been foreshadowed, but which do conform with other geophysical information thus confirming the usefulness of the methodology. The cross-autocorrelation functions for a p-dimensional multivariate time series have been extended to the case where there are $S \geq 1$ multivariate time series. These replaced the standard variance-covariance matrices for use in a principal component analysis, thus retaining measures of the time dependencies inherent to time series data. The methodology produces principal component time series, which can be compared in the usual way on a principal component plot, except that the plot also includes time as an additional plot dimension.

References

Anderson, T.W. (1984): *An Introduction to Multivariate Statistical Analysis* (2nd ed), John Wiley, New York.

Billard, L., Douzal-Chouakria, D. and Samadi, S. Y. (2015). Toward Autocorrelation Functions: A Non-Parametric Approach to Exploratory Analysis of Multiple Multivariate Time Series. Manuscript.

Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (2011): *Time Series Analysis: Forecasting and Control (4th. ed.)*. John Wiley, New York.

Brockwell, P.J. and Davis, R.A. (1991): *Time Series: Theory and Methods.* Springer-Verlag, New York.

Caiado, J., Maharaj, E. A., D'Urso, P. Time series clustering, in Handbook of Cluster Analysis, Chapman & Hall, C. Hennig, M. Meila, F. Murtagh, R. Rocci (eds.), in press.

Cryer, J.D. and Chan, K.-S. (2008): *Time Series Analysis.* Springer-Verlag, New York.

D'Urso, P., Maharaj, E. A. (2009) Autocorrelation-based Fuzzy Clustering of Time Series, Fuzzy Sets and Systems, 160, 35653589. DOI: 10.1016/j.fss.2009.04.013.

Jäckel, P. (2002): *Monte Carlo Methods in Finance.* John Wiley, New York.

Johnson, R.A. and Wichern, D.W. (2007): *Applied Multivariate Statistical Analysis* (7th ed.), Prentice Hall, New Jersey.

Joliffe, I.T. (1986): *Principal Component Analysis*, Springer-Verlag, New York.

Jones, R.H. (1964): Prediction of multivariate time series. *Journal of Applied Meteorology*, 3, 285-289.

Košmelj, K. and Batagelj, V. (1990): Cross-sectional approach for clustering time varying data. *Journal of Classification* 7, 99-109.

Liao, T.W. (2005): Clustering of time series - a survey. *Pattern Recognition* 38, 1857-1874.

Liao, T.W. (2007): A clustering procedure for exploratory mining of vector time series. *Pattern Recognition* 40, 2550-2562.

Rousseeuw, P. and Molenberghs, G. (1993): Transformation of non positive semidefnite correlation matrices. *Communications in Statistics - Theory and Methods* 22, 965-984.

Whittle, P. (1963): On the fitting of multivariate autoregressions, and the approximate canonical factorization of a spectral density matrix. *Biometrika* 50, 129-134.

Table 1 - Sample Cross-Autocorrelations - $\hat{\boldsymbol{\rho}}(k),\ k = 1$

	Sample Cross-Autocorrelations $\hat{\rho}_{jj'}(1)$											
X_j	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
X_1	0.965	0.963	0.947	0.938	0.924	0.883	0.851	0.888	0.942	0.959	0.961	0.964
X_2	0.960	0.959	0.954	0.942	0.926	0.882	0.850	0.887	0.935	0.950	0.958	0.957
X_3	0.952	0.952	0.948	0.937	0.925	0.876	0.840	0.882	0.929	0.940	0.947	0.948
X_4	0.943	0.945	0.941	0.936	0.929	0.883	0.846	0.877	0.923	0.932	0.935	0.940
X_5	0.921	0.923	0.922	0.924	0.926	0.894	0.841	0.870	0.916	0.918	0.915	0.915
X_6	0.886	0.888	0.890	0.891	0.897	0.882	0.852	0.871	0.895	0.889	0.877	0.878
X_7	0.849	0.845	0.849	0.847	0.850	0.855	0.894	0.912	0.887	0.865	0.857	0.848
X_8	0.890	0.883	0.877	0.879	0.877	0.870	0.906	0.927	0.922	0.904	0.899	0.891
X_9	0.943	0.938	0.922	0.921	0.915	0.895	0.892	0.923	0.960	0.958	0.950	0.946
X_{10}	0.960	0.953	0.938	0.931	0.921	0.891	0.869	0.906	0.956	0.964	0.963	0.958
X_{11}	0.970	0.960	0.947	0.936	0.921	0.879	0.862	0.897	0.952	0.961	0.962	0.963
X_{12}	0.969	0.960	0.948	0.933	0.920	0.878	0.849	0.889	0.946	0.959	0.962	0.961

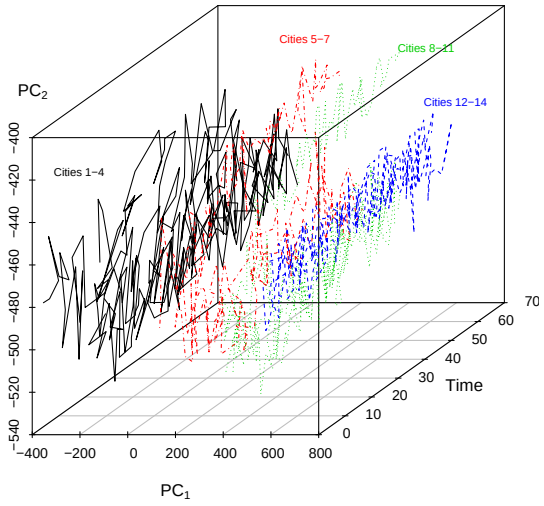


Figure 1 - Temperature Data: $PC_1 \times PC_2$ over Time – All Cities, $k = 1$

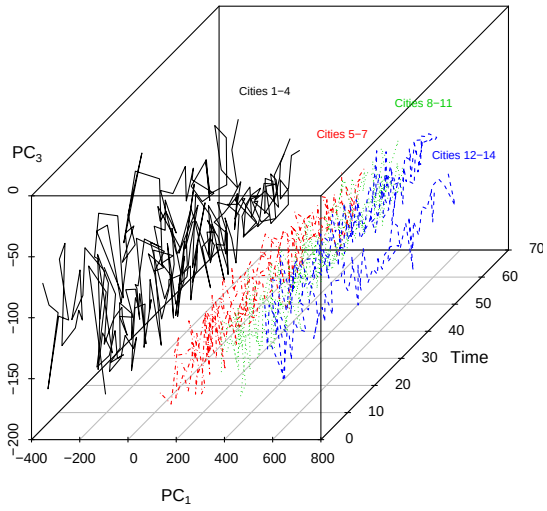


Figure 2 - Temperature Data: $PC_1 \times PC_3$ over Time – All Cities, $k = 1$

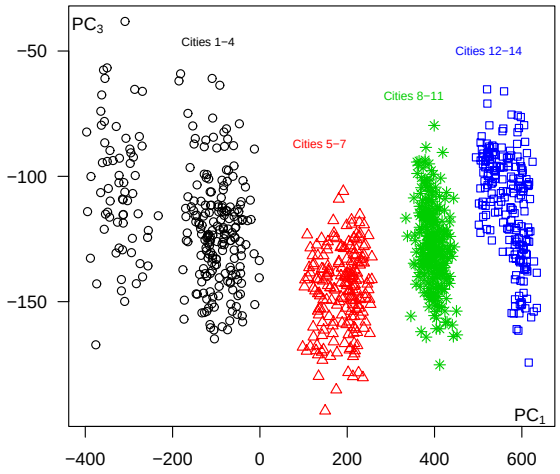


Figure 3 - Temperature Data: $PC_1 \times PC_3$ – All Cities, All Times, $k = 1$

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Symbolic Representation of Time Series: a Hierarchical Coclustering Formalization

Alexis Bondu[1], Marc Boullé[2], Antoine Cornuéjols[3]
[1] EDF R&D, 1 avenue du Général de Gaulle 92140 Clamart, France
[2] Orange Labs, 2 avenue Pierre Marzin 22300 Lannion, France
[3] AgroParisTech, 16 rue Claude Bernard 75005 Paris, France

Abstract. The choice of an appropriate representation remains crucial for mining time series, particularly to reach a good trade-off between the dimensionality reduction and the stored information. Symbolic representations constitute a simple way of reducing the dimensionality by turning time series into sequences of symbols. SAXO is a data-driven symbolic representation of time series which encodes typical distributions of data points. This approach was first introduced as a heuristic algorithm based on a regularized coclustering approach. The main contribution of this article is to formalize SAXO as a hierarchical coclustering approach. The search for the best symbolic representation given the data is turned into a model selection problem. Comparative experiments demonstrate the benefit of the new formalization, which results in representations that drastically improve the compression of data.

Keywords: Time series, symbolic representation, coclustering

1 Introduction

The choice of the representation of time series remains crucial since it impacts the quality of supervised and unsupervised analysis [1]. Time series are particularly difficult to deal with due to their inherently high dimensionality when they are represented in the time-domain [2] [3]. Virtually all data mining and machine learning algorithms scale poorly with the dimensionality. During the last two decades, numerous high level representations of time series have been proposed to overcome this difficulty. The most commonly used approaches are: the Discrete Fourier Transform [4], the Discrete Wavelet Transform [5] [6], the Discrete Cosine Transform [7], the Piecewise Aggregate Approximation (PAA) [8]. Each representation of time series encodes some information derived from the raw data[4]. According to [1], mining time series heavily relies on the choice of a representation and a similarity measure. Our objective is to find a **compact** and **informative** representation which is driven by the data. The symbolic representations constitute a simple way of reducing the dimensionality of the data by turning time series into sequences of symbols [9]. In such representations, each symbol corresponds to a time interval and encodes information which summarize

[4] *"Raw data"* designates a time series represented in the time-domain by a vector of real values.

the related sub-series. Without making hypothesis on the data, such a representation does not allow one to quantify the loss of information. This article focuses on a less prevalent symbolic representation which is called SAXO[5]. This approach optimally discretizes the time dimension and encodes typical distributions[6] of data points with the symbols [10]. SAXO offers interesting properties. Since this representation is based on a **regularized** Bayesian coclustering[7] approach called MODL[8] [11], a good trade-off is naturally reached between the dimensionality reduction and the information loss. SAXO is a parameter-free and data-driven representation of time series. In practice, this symbolic representation proves to be highly **informative** for training classifiers. In [10], SAXO was evaluated on public datasets and favorably compared with the SAX representation.

Originally, SAXO was defined as a heuristic algorithm. The two main contributions of this article are: i) the **formalization** of SAXO as a hierarchical coclustering approach; ii) the evaluation of its **compactness** in terms of coding length. This article is organized as follows. Section 2 briefly introduces the symbolic representations of time series and presents the original SAXO heuristic algorithm. Section 3 formalizes the SAXO approach resulting in a new evaluation criterion which is the main contribution of this article. Experiments are conducted in Section 4 on real datasets in order to compare the SAXO evaluation criterion with that of the MODL coclustering approach. Lastly, perspectives and future works are discussed in Section 5.

2 Related work

Numerous compact representations of time series deal with the curse of dimensionality by discretizing the time and by summarizing the sub-series within each time interval. For instance, the Piecewise Aggregate Approximation (PAA) encodes the mean values of data points within each time interval. The Piecewise Linear Approximation (PLA) [12] is an other example of compact representation which encodes the gradient and the y-intercept of a linear approximation of sub-series. In both cases, the representation consist of numerical values which describe each time interval. In contrast, the symbolic representations characterize the time intervals by categorical variables [9]. For instance, the Shape Definition Language (SDL) [13] encodes the shape of sub-series by symbols. The most commonly used symbolic representation is the SAX[9] approach [9]. In this case, the time dimension is discretized into regular intervals, the symbols encode the mean values per interval.

[5] SAXO *Symbolic Aggregate approXimation Optimized by data.*
[6] The SAXO approach produces clusters of time series within each time interval which correspond to the symbols.
[7] The coclustering problem consist in reordering rows and columns of a matrix in order to satisfy a homogeneity criterion.
[8] *Minimum Optimized Description Length*
[9] *Symbolic Aggregate approXimation.*

The symbolic representations appear to be really helpful for processing large datasets of time series owing to dimensionality reduction. However, these approaches suffer several limitations.

- Most of these representations are lossy compression approaches unable to quantify the loss of information without strong hypothesis on the data.
- The discretization of the time dimension into regular intervals is not data driven.
- The symbols have the same meaning over time irrespectively of their rank *(i.e. the ranks of the symbols may be used to improve the compression)*.
- Most of these representations involve user parameters which affect the stored information *(ex: for the SAX representation, the number of time intervals and the size of the alphabet must be specified)*.

The SAXO approach overcomes these limitations by optimizing the time discretization, and by encoding typical distributions of data points within each time interval [10]. SAXO was first defined as a heuristic which exploits the MODL coclustering approach.

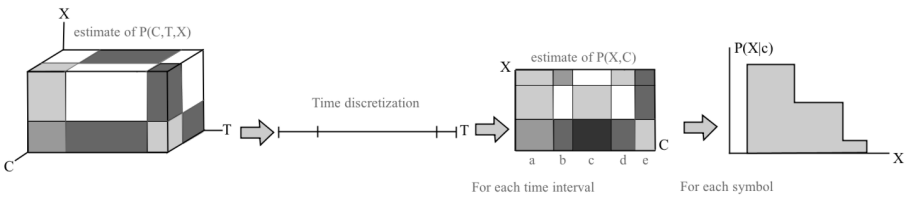
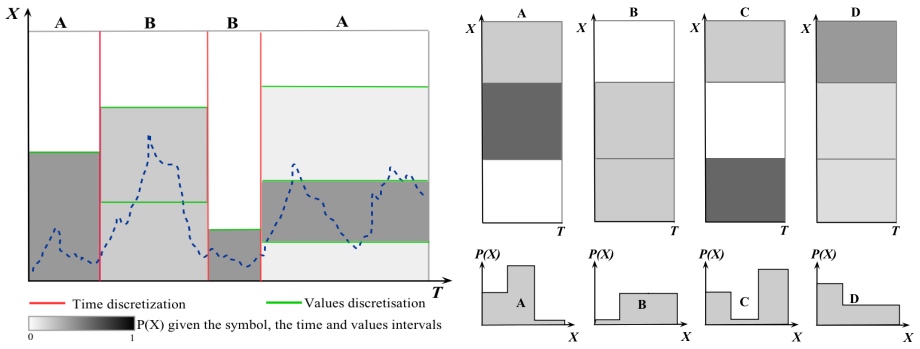


Fig. 1. Main steps of the SAXO learning algorithm.

Figure 1 provides an overview of this approach by illustrating the main steps of the learning algorithm. The joint distribution of the identifiers of the time series C, the values X, and the timestamp T is estimated by a trivariate coclustering model. The time discretization resulting from the first step is retained, and the joint distribution of X and C is estimated within each time interval by using a bivariate coclustering model. The resulting clusters of time series are characterized by piecewise constant distributions of values and correspond to the symbols. A specific representation allows one to re-encode the time series as a sequence of symbols. Then, the typical distribution that best represents the data points of the time series is selected within each time interval. Figure 2(a) plots an example of recoded time series. The original time series *(represented by the blue curve)* is recoded by the **"abba"** SAXO word. The time is discretized into four intervals *(the vertical red lines)* corresponding to each symbol. Within time intervals, the values are discretized *(the horizontal green lines)*: the number of intervals of values and their locations are not necessary the same. The symbols correspond to typical distributions of values: conditional probabilities of X are associated with each cell of the grid *(represented by the gray levels)*; Figure 2(b) gives an example of the alphabet associated with the second time interval. The four available symbols correspond to typical distributions which are both

represented by gray levels and by histograms. By considering Figures 2(a) and 2(b), **b** appears to be the closest typical distribution of the second sub-series.



(a) (b)

Fig. 2. Example of a SAXO representation (a) and the alphabet of the second time interval (b).

As in any heuristic approach, the original algorithm finds a suboptimal solution for selecting the most suitable SAXO representation given the data. Solving this problem in an exact way appears to be intractable, since it is comparable to the coclustering problem which is NP-hard. The main contribution of this paper is to **formalize** the SAXO approach within the MODL framework. We claim this formalization is a first step to improving the quality of the SAXO representations learned from data. In this article, we define a new evaluation criterion denoted by C_{saxo} *(see Section 3)*. The most probable SAXO representation given the data is defined by minimizing C_{saxo}. We expect to reach better representations by optimizing C_{saxo}, instead of exploiting the original heuristic algorithm.

3 Formalization of the SAXO approach

This section presents **the main contribution** of this article: the SAXO approach is formalized as a hierarchical coclustering approach. As illustrated in Figure 3, the originality of the SAXO approach is that the groups of identifiers *(variable C)* and the intervals of values *(variable X)* are allowed to change over time. By contrast, the MODL coclustering approach forces the discretization of C and X to be the same within time intervals. Our objective is to reach better models by removing this constraint.

A SAXO model is hierarchically instantiated by following two successive steps. First, the discretization of time is determined. The bivariate discretization $C \times X$ is then defined within each time interval. Additional notations are required to describe the sequence of bivariate data grids.

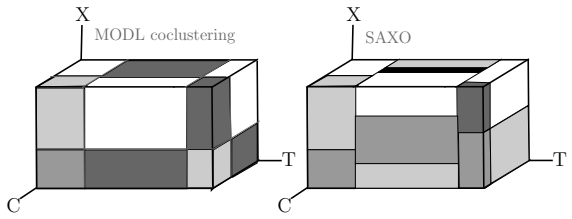


Fig. 3. Examples of a MODL coclustering model *(left part)* and a SAXO model *(right part)*.

Notations for time series: *In this article, the input dataset $\mathcal{D}$ is considered to be a collection of N time series denoted S_i (with $i \in [1, N]$). Each time series consists of m_i data points, which are couples of values X and timestamps T. The total number of data points is denoted by $m = \sum_{i=1}^{N} m_i$.*

Notations for the *t-th* time interval of a SAXO model:

- k_T*: number of time intervals;*
- k_C^t*: number of clusters of time series;*
- k_X^t*: number of intervals of value;*
- $k_C(i,t)$*: index of the cluster that contains the sub-series of S_i;*
- $\{n_{i_C}^t\}$*: number of time series in each cluster i_C^t;*
- m_t*: number of data point;*
- m_i^t*: number of data points of each time series S_i;*
- $m_{i_C}^t$*: number of data points in each cluster i_C^t;*
- $\{m_{j_X}^t\}$*: number of data points in the intervals j_X;*
- $\{m_{i_C j_X}^t\}$*: number of data points belonging to each cell (i_C, j_X).*

Eventually, a SAXO model M' is first defined by a number of time intervals and the location of their bounds. The bivariate data grids $C \times X$ within each time interval are defined by: i) the partition of the time series into clusters; ii) the number of intervals of values; iii) the distribution of the data points on the cells of the data grid; iv) for each cluster, the distribution of the data points on the time series belonging to the same cluster. Section 3.1 presents the prior distribution of the SAXO models. The likelihood of a SAXO model given the data is described in Section 3.2. A new evaluation criterion which defines the most probable model given the data is proposed in Section 3.3.

3.1 Prior distribution of the SAXO models

The proposed prior distribution $P(M')$ exploits the hierarchy of the parameters of the SAXO models and is uniform at each level. The prior distribution of the number of time intervals k_T is given by Equation 1. The parameter k_T belongs to $[1, m]$, with m representing the total number of data points. All possible values

of k_T are considered as equiprobable. By using combinatorics, the number of possible locations of the bounds can be enumerated given a fixed value of k_T. Once again, all possible locations are considered as equiprobable. Equation 2 represents the prior distribution of the parameter $\{m^t\}$ given k_T. Within each time interval t, the number of intervals of values k_X^t is uniformly distributed *(see Equation 3)*. The value of k_X^t belongs to $[1, m^t]$, with m^t representing the number of data points within the *t-th* time interval. All possible values of k_X^t are equiprobable. The same approach is applied to define the prior distribution of the number of clusters within each time interval *(see Equation 4)*. The value of k_C^t belongs to $[1, N]$, with N denoting the total number of time series. Once again, all possible values of k_C^t are equiprobable. The possible ways of partitioning the N time series into k_C^t clusters can be enumerated, given a fixed number of clusters in the *t-th* time interval. The term $B(N, k_C^t)$ in Equation 5 represents the number of possible partitions of N elements into k_C^t possibly empty clusters[10]. Within each time interval, all distributions of the m^t data points on the cells of the bivariate data grid $C \times X$ are considered as equiprobable. Equation 6 enumerates the possible ways of distributing $\{m^t\}$ data points on $k_X^t.k_C^t$ cells. Given a time interval t and a cluster i_C^t, all distributions of the data points on the time series belonging to the same cluster are equiprobable. Equation 7 enumerates the possible ways of distributing m_i^t data points on $n_{i_C}^t$ time series.

$$P(k_T) = \frac{1}{m} \quad (1) \qquad P(\{m^t\}|k_T) = \frac{1}{\binom{m+k_T-1}{k_T-1}} \quad (2) \qquad P(\{k_X^t\}|k_T, \{m^t\}) = \prod_{t=1}^{k_T} \frac{1}{m^t} \quad (3)$$

$$P(\{k_C^t\}|k_T) = \prod_{t=1}^{k_T} \frac{1}{N} \quad (4) \qquad P\left(k_C(i,t)|k_T, \{k_C^t\}\right) = \prod_{t=1}^{k_T} \frac{1}{B(N, k_C^t)} \quad (5)$$

$$P\left(\{m_{j_C,j_X}^t\}|k_T, \{m^t\}, \{k_X^t\}, \{k_C^t\}\right) = \prod_{t=1}^{k_T} \frac{1}{\binom{m^t+k_C^t.k_X^t-1}{k_C^t.k_X^t-1}} \quad (6)$$

$$P\left(\{m_i^t\}|k_T, \{k_C^t\}, k_C(i,t), \{m_{j_C,j_X}^t\}\right) = \prod_{t=1}^{k_T} \prod_{i=1}^{k_C^t} \frac{1}{\binom{m_{i_C}^t+n_{i_C}^t-1}{n_{i_C}^t-1}} \quad (7)$$

In the end, the prior distribution of the SAXO models M' is given by Equation 8.

$$P(M') = \frac{1}{m} \times \frac{1}{\binom{m+k_T-1}{k_T-1}} \times \prod_{t=1}^{k_T} \left[\frac{1}{m^t} \times \frac{1}{N} \times \frac{1}{B(N, k_C^t)} \times \frac{1}{\binom{m^t+k_C^t.k_X^t-1}{k_C^t.k_X^t-1}} \times \prod_{i=1}^{k_C^t} \frac{1}{\binom{m_{i_C}^t+n_{i_C}^t-1}{n_{i_C}^t-1}} \right] \quad (8)$$

[10] The second kind of Stirling numbers $S\left\{{v \atop k}\right\}$ enumerates the possible partitions of v elements into k clusters and $B(N, k_C^t) = \sum_{i=1}^{k_C^t} S\left\{{N \atop i}\right\}$.

3.2 Likelihood of data given a SAXO model

A SAXO model matches with several possible datasets. Intuitively, the likelihood $P(D|M')$ enumerates all the datasets which are compatible with the parameters of the model M'. The first term of the likelihood represents the distribution of the ranks of the values of T. In other words, Equation 9 codes all the possible permutations of the data points within each time interval. The second term enumerates all the possible distributions of the m data points on the k_T time intervals, which are compatible with the parameter $\{m^t\}$ *(see Equation 10)*. In the same way, Equation 11 enumerates the distributions of the m^t data points on the $k_X^t.k_C^t$ cells of the bivariate data grids $C \times X$ within each time interval. The considered distributions are compatible with the parameter $\{m^t_{i_C,j_X}\}$. For each time interval and for each cluster, Equation 12 enumerates all the possible distributions of the data points on the time series belonging to the same cluster. Equation 13 enumerates all the possible permutations of the data points in the intervals of X, within each time interval. This information must also be coded over all the time intervals, which is equivalent to enumerating all the possible fusions of k_T stored lists in order to constitute a global stored list *(see Equation 14)*. In the end, the likelihood of the data given a SAXO models M' is characterized by Equation 15.

$$\frac{1}{\prod_{t=1}^{k_T} m^t!} \quad (9) \qquad \frac{1}{\frac{m!}{\prod_{t=1}^{k_T} m^t!}} \quad (10) \qquad \prod_{t=1}^{k_T} \frac{1}{\frac{m^t!}{\prod_{i_C=1}^{k_C^t}\prod_{j_X=1}^{k_X^t} m^t_{i_C,j_X}!}} \quad (11)$$

$$\prod_{t=1}^{k_T} \frac{1}{\frac{\prod_{i_C=1}^{k_C^t} m^t_{i_C}!}{\prod_{i=1}^{N} m^t_i!}} \quad (12) \qquad \prod_{t=1}^{k_T} \frac{1}{\prod_{j_X=1}^{k_X^t} m^t_{j_X}!} \quad (13) \qquad \frac{1}{\frac{m!}{\prod_{t=1}^{k_T} m^t!}} \quad (14)$$

$$P(D|M') = \frac{1}{m!^2} \times \prod_{t=1}^{k_T} \left[\frac{\prod_{i_C=1}^{k_C^t}\prod_{j_X=1}^{k_X^t} m^t_{i_C,j_X}! \times \prod_{i=1}^{N} m^t_i!}{\prod_{j_X=1}^{k_X^t} m^t_{j_X}! \times \prod_{i_C=1}^{k_C^t} m^t_{i_C}!} \right] \quad (15)$$

3.3 Evaluation criterion

The SAXO evaluation criterion is the negative logarithm of $P(M') \times P(D|M')$ *(see Equation 16)*. The first three lines correspond to the prior term $-log(P(M'))$ and the last two lines represent the likelihood term $-log(P(M'|D))$. The most probable model given the data is found by minimizing $C_{saxo}(M')$ over the set of all possible SAXO models denoted by $\mathbb{M}'$.

$$
\begin{aligned}
C_{saxo}(M') &= log(m) + log\binom{m+k_T-1}{k_T-1} + \sum_{t=1}^{k_T} log(m^t) \\
&+ k_T.log(N) + \sum_{t=1}^{k_T} log\big(B(N,k_C^t)\big) + \sum_{t=1}^{k_T} log\binom{m^t + k_C^t.k_X^t - 1}{k_C^t.k_X^t - 1} \\
&+ \sum_{t=1}^{k_T} \sum_{i_C=1}^{k_C^t} log\binom{m_{i_C}^t + n_{i_C}^t - 1}{n_{i_C}^t - 1} \\
&+ 2.log(m!) - \sum_{t=1}^{k_T} \sum_{i_C=1}^{k_C^t} \sum_{j_X=1}^{k_X^t} log(m_{i_C,j_X}^t!) \\
&+ \sum_{t=1}^{k_T} \left[\sum_{i_C=1}^{k_C^t} log(m_{i_C}^t!) - \sum_{i=1}^{N} log(m_i^t!) + \sum_{j_X=1}^{k_X^t} log(m_{j_X}^t!) \right]
\end{aligned}
\tag{16}
$$

Key ideas to retain: Rather than having a heuristic decomposition of the SAXO approach in a two-step algorithm, we propose a single evaluation criterion based on the MODL framework. Once optimized, this criterion should yield better representations of time series. We compare the ability of both criterion to compress data. We aim at evaluating the interest of optimizing C_{saxo} rather than the original trivariate coclustering criterion [14] *(denoted by C_{modl})*.

4 Comparative experiments on real datasets

According to the information theory and since both criteria are a negative logarithm of a probability, C_{saxo} and C_{modl} represent the coding length of the models. In this section, both approaches are compared in terms of coding length. The 20 processed datasets come from the *UCR Time Series Classification and Clustering repository* [15]. Some datasets are relatively small, we have selected the ones which include at least 800 learning examples. Originally, these datasets are divided into training and test sets which have been merged in our experiments. The objective of this section is to compare C_{saxo} and C_{modl} for each dataset. On the one hand, the criterion C_{modl} is optimized by using the greedy heuristic and a neighborhood exploration mentioned described in [11]. The coding length of the most probable MODL model *(denoted by MAP_{modl})* is then calculated by using C_{modl}. On the other hand, the criterion C_{saxo} is optimized by exploiting the original heuristic algorithm illustrated in Figure 1 [10]. The coding length of best SAXO model *(denoted by MAP_{saxo})* is given by the criterion C_{saxo}. Notice that both algorithms have a $\mathcal{O}(m\sqrt{m}\log m)$ time complexity. The order of magnitude of the coding length depends on the size of the data set and can not be easily compared over all datasets. We choose to exploit the compression gain [16] which consists in comparing the coding length of a model M with the coding

length of the simplest model M_{sim}. This key performance indicator varies in the interval $[0, 1]$. The compression gain is similarly defined for the MODL and the SAXO approaches such that:

$$\mathcal{G}ain_{modl}(M) = 1 - C_{modl}(M)/C_{modl}(M_{sim})$$
$$\mathcal{G}ain_{saxo}(M') = 1 - C_{saxo}(M')/C_{saxo}(M_{sim})$$

Our experiments evaluate the variation of the compression gain between the SAXO and the MODL approaches. This indicator is denoted by Δ_G and represents the relative improvement of the compression gain provided by SAXO. The value of Δ_G can be negative, which means that SAXO provides a worse compression gain than the MODL approach.

$$\Delta_G = \frac{\mathcal{G}ain_{saxo}(MAP_{saxo}) - \mathcal{G}ain_{modl}(MAP_{modl})}{\mathcal{G}ain_{modl}(MAP_{modl})}$$

Dataset	Δ_G	Dataset	Δ_G
Starlight curves	63.86%	CBF	−1.43%
uWaveGestureX	191.41%	AllFace	383.24%
uWaveGestureY	157.79%	Symbols	23.16%
uWaveGestureZ	185.13%	50 Words	400.68%
ECG Five Days	−1.80%	Wafer	37.03%
MoteStrain	627.84%	Yoga	63.40%
CincECGtorso	32.93%	FacesUCR	−18.39%
MedicalImages	191.32%	Cricket Z	290.22%
WordSynonym	264.93%	Cricket X	285.87%
TwoPatterns	missing	Cricket Y	296.40%

Table 1. Coding length evaluation.

Table 1 presents the results of our experiments and includes a particular case with a missing value for the dataset "TwoPatterns". In this case, the first step of the heuristic algorithm which optimizes C_{saxo} *(see Figure 1)* leads to the simplest trivariate coclustering model that includes a single cell. This is a side effect due to the fact that the MODL approach is regularized. A possible explanation is that the temporal representation of time series is not informative for this dataset. Other representations such as the Fourier or the wavelet transforms could be tried. In most cases, Δ_G has a positive value which means SAXO provides a better compression than the MODL approach. This trend emerges clearly, the average compression improvement reaches 183%. We exploit the *Wilcoxon signed-ranks test* to reliably comparing both approaches over all datasets [17]. If the output value *(denoted by z)* is smaller than -1.96, the gap in performance is considered as significant. Our experiments give $z = -3.37$ which is highly significant. In the end, the compression of data provided by SAXO appears to be intrinsically better than the MODL approach. The prior term of C_{saxo} induces an additional cost in terms of coding length. This additional cost is far outweighed by a better encoding of the likelihood.

5 Conclusion and perspectives

SAXO is a data-driven symbolic representation of time series which extends SAX in three ways: i) the discretization of time is optimized by a Bayesian approach rather than considering regular intervals; ii) the symbols within each time interval represents typical distributions of data points rather than average values; iii) the number of symbols may differ per time interval. The parameter settings is automatically optimized given the data. SAXO was first introduced as an heuristic algorithm. This article formalizes this approach within the MODL framework as a hierarchical coclustering approach *(see Section 3)*. A Bayesian approach is applied leading to an analytical evaluation criterion. This criterion must be minimized in order to define the most probable representation given the data. This new criterion is evaluated on real datasets in Section 4. Our experiments compare the SAXO representation with the original MODL coclustering approach. The SAXO representation appears to be significantly better in terms of data compression. In future work, we plan to use the SAXO criterion in order to define a similarity measure. Numerous learning algorithms, such as K-means and K-NN, could use such an improved similarity measure defined over time series. We plan to explore potential gains in areas such as: i) the detection of atypical time series; ii) the query of a database by similarity; iii) the clustering of time series.

References

1. T. Liao, "Clustering of time series data: a survey," *Pattern Recognition*, vol. 38, pp. 1857–1874, 2005.
2. D. Bosq, *Linear Processes in Function Spaces: Theory and Applications (Lecture Notes in Statistics)*. Springer, 2000.
3. J. Ramsay and B. Silverman, *Functional Data Analysis*, ser. Springer Series in Statistics. Springer, 2005.
4. M. Frigo and S. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005, special issue on "Program Generation, Optimization, and Platform Adaptation".
5. R. Polikar, *Physics and Modern Topics in Mechanical and Electrical Engineering*. World Scientific and Eng. Society Press, 1999, ch. The story of wavelets.
6. K. Chan and W. Fu, "Efficient Time Series Matching by Wavelets," in *ICDE '99: Proceedings of the 15th International Conference on Data Engineering*. IEEE Computer Society, 1999.
7. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transfom," *IEEE Trans. Comput.*, vol. 23, no. 1, pp. 90–93, 1974.
8. C. Guo, H. Li, and D. Pan, "An improved piecewise aggregate approximation based on statistical features for time series mining," in *Proceedings of the 4th international conference on Knowledge science, engineering and management*, ser. KSEM'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 234–244.
9. J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," in *8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, San Diego, 2003.

10. A. Bondu, M. Boullé, and B. Grossin, "SAXO : An Optimized Data-driven Symbolic Representation of Time Series," in *IJCNN (International Joint Conference on Neural Networks)*. IEEE, 2013.
11. M. Boullé, *Hands on pattern recognition*. Microtome, 2010, ch. Data grid models for preparation and modeling in supervised learning.
12. H. Shatkay and S. B. Zdonik, "Approximate Queries and Representations for Large Data Sequences," in *12th International Conference on Data Engineering (ICDE)*, 1996, pp. 536–545.
13. R. Agrawal, G. Psaila, E. L. Wimmers, and M. Zait, "Querying Shapes of Histories," in *21th International Conference on Very Large Data Bases (VLDB 95)*, 1995, pp. 502–514.
14. M. Boullé, "Functional data clustering via piecewise constant nonparametric density estimation," *Pattern Recognition*, vol. 45, no. 12, pp. 4389–4401, 2012.
15. E. Keogh, Q. Zhu, B. Hu, H. Y., X. Xi, L. Wei, and C. A. Ratanamahatana, "The UCR Time Series Classification/Clustering Homepage : www.cs.ucr.edu/∼eamonn/time_series_data/," 2011.
16. M. Boullé, "Optimum simultaneous discretization with data grid models in supervised classification: a Bayesian model selection approach," *Advances in Data Analysis and Classification*, vol. 3, no. 1, pp. 39–61, 2009.
17. J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, Dec. 2006.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Temporal and Frequential Metric Learning for Time Series *k*NN Classification

Cao-Tri Do[123], Ahlame Douzal-Chouakria[2], Sylvain Marié[1], and Michèle Rombaut[3]

[1] Schneider Electric, France
[2] LIG, University of Grenoble Alpes, France
[3] GIPSA-Lab, University of Grenoble Alpes, France

Abstract. This work proposes a temporal and frequential metric learning framework for a time series nearest neighbor classification. For that, time series are embedded into a pairwise space where a combination function is learned based on a maximum margin optimization process. A wide range of experiments are conducted to evaluate the ability of the learned metric on time series kNN classification.

Keywords: Metric learning, Time series, kNN, Classification, Spectral metrics.

1 Introduction

Due to their temporal and frequential nature, time series constitute complex data to analyze by standard machine learning approaches [1]. In order to classify such challenging data, distance features must be used to bring closer time series of identical classes and separate those of different classes. Temporal data may be compared on their values. The most frequently used value-based metrics are the Euclidean distance and the Dynamic Time Warping DTW to cope with delays [2, 3]. They can also be compared on their dynamics and frequential characteristics [4, 5]. Promising approaches aims to learn the Mahalanobis distance or kernel function for a specific classifier [6, 7]. Other work investigate the representation paradigm by representating objects in a dissimilarity space where are investigated dissimilarity combinations and metric learning [8, 9]. The idea in this paper is to combine basic metrics into a discriminative one for a kNN classifier. In the metric learning context for a metric learning approach driven by nearest neighbors (Weinberger & Saul [6]), we extend the work of Do & al. in [10] to temporal and frequential characteristics. The main idea is to embed pairs of time series in a space whose dimensions are basic temporal and frequential metrics, where a combination function is learned based on a large margin optimization process.

The main contributions of the paper are a) propose a new temporal and frequential metric learning framework for a time series nearest neighbors classification, b) learn a combination metric involving amplitude, behavior and frequential characteristics and c) conduct large experimentations to study the ability of learned metric. The rest of the paper is organized as follows. Section 2 recalls briefly the major metrics for time series. In Section 3, we present the proposed

metric learning approach. Finally, Section 4 presents the experiments conducted and discusses the results obtained.

2 Time series metrics

Let $\mathbf{x}_i = (x_{i1}, ..., x_{iT})$ and $\mathbf{x}_j = (x_{j1}, ..., x_{jT})$ be two time series of time length T. Time series metrics fall at least within three main categories. The first one concerns value-based metrics, where time series are compared according to their values regardless of their behaviors. Among these metrics are the Euclidean distance (d_E), the Minkowski distance and the Mahalanobis distance [3]. We recall the formula of d_E:

$$d_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{t=1}^{T}(x_{it} - x_{jt})^2} \tag{1}$$

The second category relies on metrics in the spectral representations. In some applications, time series may be similar because they share the same frequency characteristics. For that, time series $\mathbf{x}_i$ are first transformed into their Fourier representation $\tilde{\mathbf{x}}_i = [\tilde{x}_{i1}, ..., \tilde{x}_{iF}]$, where $\tilde{x}_{if}$ is a complex number (i.e. Fourier components), with $F = \frac{2^T}{2} + 1$ [5]. Then, one may use the Euclidean distance (d_{FFT}) between the module of the complex numbers $\tilde{x}_{if}$, noted $|\tilde{x}_{if}|$:

$$d_{FFT}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{f=1}^{F}(|\tilde{x}_{if}| - |\tilde{x}_{jf}|)^2} \tag{2}$$

Note that times series of similar frequential characteristics may have distinctive global behavior. Thus, to compare time series based on their behavior, a third category of metrics is used. Many applications refer to the Pearson correlation or its generalization, the temporal correlation coefficient [4] defined as:

$$Cort_r(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum\limits_{t,t'}(x_{it} - x_{it'})(x_{jt} - x_{jt'})}{\sqrt{\sum\limits_{t,t'}(x_{it} - x_{it'})^2}\sqrt{\sum\limits_{t,t'}(x_{jt} - x_{jt'})^2}} \tag{3}$$

where $|t - t'| \leq r$, $r \in [1, ..., T-1]$ being a parameter that can be learned or fixed *a priori*. The optimal value of r is noisy dependant. For $r = T - 1$, Eq. 3 leads to the Pearson correlation. As $Cort_r$ is a similarity measure, it is transformed into a dissimilarity measure: $d_{Cort_r}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(1 - Cort_r(\mathbf{x}_i, \mathbf{x}_j))$.

3 Temporal and frequential metric learning for a large margin *k*NN

Let $\mathbf{X} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ be a set of N static vector samples, $\mathbf{x}_i \in \mathbb{R}^p$, p being the number of descriptive features and y_i the class labels. Weinberger & Saul proposed in [6] an approach to learn a dissimilarity metric D for a large margin

kNN. It is based on two intuitions: first, each training sample $\mathbf{x}_i$ should have the same label y_i as its k nearest neighbors; second, training samples with different labels should be widely separated. For this, they introduced the concept of *target* for each training sample $\mathbf{x}_i$. *Target* neighbors of $\mathbf{x}_i$, noted $j \rightsquigarrow i$, are the k closest $\mathbf{x}_j$ of the same class ($y_j = y_i$). The *target* neighborhood is defined with respect to an initial metric. The aim is to learn a metric D that pulls the *targets* and pushes the ones of different class.

Let $d_1, ..., d_h..., d_p$ be p given dissimilarity metrics that allow to compare samples. The computation of a metric always takes into account a pair of samples. Therefore, we used the pairwise representation introduced in Do & al. [10]. In this space, a vector $\mathbf{x}_{ij}$ represents a pair of samples $(\mathbf{x}_i, \mathbf{x}_j)$ described by the p basics metrics d_h: $\mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), ..., d_p(\mathbf{x}_i, \mathbf{x}_j)]^T$. If $\mathbf{x}_{ij} = \mathbf{0}$ then $\mathbf{x}_j$ is identical to $\mathbf{x}_i$ according to all metrics d_h. A combination function D of the metrics d_h can be seen as a function in this space. We propose in the following to use a linear combination of d_h: $D_w(\mathbf{x}_i, \mathbf{x}_j) = \sum_h w_h.d_h(\mathbf{x}_i, \mathbf{x}_j)$. Its pairwise notation is $D_w(\mathbf{x}_{ij}) = \mathbf{w}^T.\mathbf{x}_{ij}$. To ensure that D_w is a valid metric, we set $w_h \geq 0$ for all $h = 1...p$. The main steps of the proposed approach to learn the metric, detailed hereafter, can be summarized as follows:

1. Embed each pair $(\mathbf{x}_i, \mathbf{x}_j)$ into the pairwise space $\mathbb{R}^p$.
2. Scale the data within the pairwise space.
3. Define for each $\mathbf{x}_i$ its *targets*.
4. Scale the neighborhood of each $\mathbf{x}_i$.
5. Learn the combined metric D_w.

Data scaling. This operation is performed to scale the data within the pairwise space and ensure comparable ranges for the p basic metrics d_h. In our experiment, we use dissimilarity measures with values in $[0; +\infty[$. Therefore, we propose to Z-normalize their log distributions.

Target set. For each $\mathbf{x}_i$, we define its *target* neighbors as the k nearest neighbors $\mathbf{x}_j$ ($j \rightsquigarrow i$) of the same class according to an initial metric. In this paper, we choose a L2-norm of the pairwise space as the initial metric ($\sqrt{\sum_h d_h^2}$). Other metrics could be chosen. We emphasize that *target* neighbors are fixed *a priori* (at the first step) and do not change during the learning process.

Neighborhood scaling. In real datasets, local neighborhoods can have very different scales. To make the target neighborhood spreads comparable, we propose for each $\mathbf{x}_i$ to scale its neighborhood vectors $\mathbf{x}_{ij}$ such that the L2-norm of the farthest *target* is 1.

Learning the combined metric D_w. Let $\{\mathbf{x}_{ij}, y_{ij}\}_{i,j=1}^N$ be the training set with $y_{ij} = -1$ if $y_j = y_i$ and $+1$ otherwise. Learning D_w for a large margin kNN classifier can be formalized as the following optimization problem:

$$\begin{aligned}
&\min_{w,\xi} \underbrace{\sum_{i,j\rightsquigarrow i} D_w(\mathbf{x}_{ij})}_{pull} + \underbrace{C \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2}.\xi_{ijl}}_{push} \\
&\text{s.t. } \forall j \rightsquigarrow i, y_l \neq y_i, \\
&D_w(\mathbf{x}_{il}) - D_w(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\
&\xi_{ijl} \geq 0 \\
&w_h > 0 \ \forall h = 1...p
\end{aligned} \tag{4}$$

Note that the "pull" term $\sum_{j\rightsquigarrow i} D_w(\mathbf{x}_{ij}) = \sum_{j\rightsquigarrow i} \mathbf{w}^T.\mathbf{x}_{ij} = N.k.\mathbf{w}^T.\bar{\mathbf{x}}_{ij}$ is a L1-Mahalanobis norm weighted by the average target sample. Therefore, it behaves like a L1-norm in the optimization problem. The problem is very similar to a C-SVM classification problem. When C is infinite, we have a "strict" problem: the solver will try to find a direction in the pairwise space for which only targets are in the close neighborhood of each $\mathbf{x}_i$, and a maximum margin $\frac{1}{||\mathbf{w}||_2}$.

Let $\mathbf{x}_{test}$ be a new sample to classify and $\mathbf{x}_{test,i}$ $(i = 1, ..., N)$ the corresponding vectors into the pairwise embedding space. After $\mathbf{x}_{test,i}$ normalization according to the *Data Scaling* step, $\mathbf{x}_{test}$ is classified based on a standard kNN and D_w.

4 Experiments

In this section, we compare kNN classifier performances for several metrics on reference time series datasets [11–14] described in Table 1. To compare with the reference results in [3, 11], the experiments are conducted with the same protocols as in Do &. al. [10]: k is set to 1; train and test set are given a priori. Due to the current format to store the data, small datasets with short time series were retained and the experiments are conducted on one runtime.

In this experimentation, we consider basic metrics d_E, d_{FFT} and d_{Cort_r} then, we learn a combined metric D_w according to the procedure described in Section 3. First, two basic temporal metrics are considered in D_2 (d_E and d_{Cort_r}) as in Do & al. [10]. Second, we consider a combination between temporal and frequential metrics in D_3 (d_E, d_{Cort_r} and d_{FFT}). Cplex library [15] has been used to solve the optimization problem in Eq. 4. We learn the optimal parameter values of these metrics by minimizing a leave-one out cross-validation criterion.

As the training dataset sizes are small, we propose a hierarchical error criterion:

1. Minimize the kNN error rate
2. Minimize $\frac{d_{intra}}{d_{inter}}$ if several parameter values obtain the minimum kNN error.

where d_{intra} and d_{inter} stands respectively to the mean of all intraclass and interclass distances according to the metric at hand. Table 2 gives the range of the grid search considered for the parameters. In the following, we consider only the raw series and don't align them using a DTW algorithm for example. For

all reported results (Table 3), the best one is indexed with a star and the ones significantly similar from the best one (Z-test at 1% risk) are in bold [16].

Dataset	Nb. Class	Nb. Train	Nb. Test	TS length
SonyAIBO	2	20	601	70
MoteStrain	2	20	1252	84
GunPoint	2	50	150	150
PowerCons	2	73	292	144
ECG5Days	2	23	861	136
SonyAIBOII	2	27	953	65
Coffee	2	28	28	286
BME	3	48	102	128
UMD	3	46	92	150
ECG200	2	100	100	96
Beef	5	30	30	470
DiatomSizeReduction	4	16	306	345
FaceFour	4	24	88	350
Lighting-2	2	60	61	637
Lighting-7	7	70	73	319
OliveOil	4	30	30	570

Table 1. Dataset description giving the number of classes (Nb. Class), the number of time series for the training (Nb. Train) and the testing (Nb. Test) sets, and the length of each time series (TS length).

Method	Parameter	Parameter range
d_{Cort_r}	r	$[1, 2, 3, , ..., T]$
D_2, D_3	C	$[10^{-3}, 0.5, 1, 5, 10, 20, 30, ..., 150]$

Table 2. Parameter ranges

From Table 3, we can see that temporal metrics d_E and d_{Cort_r} alone performs better one from the other depending on the dataset. Using a frequential metric alone such as d_{FFT} brings significant improvements for some datasets (SonyAIBO, GunPoint, PowerCons, ECG5Days). It can be observed that one basic metric is sufficient on some databases (MoteStrain, GunPoint, PowerCons, ECG5Days). In other cases, learning a combination of these basic metrics reach the same performances on most datasets or achieve better results (UMD). The new approach allows to extend combination functions to many metrics without having to cope with additional parameters in grid search and without to test every basic metrics alone to retained the best one. It also extends the work done in [6] for single distance to multiple distances. Adding metrics such as d_{FFT} improves the performances on some datasets (SonyAIBO, GunPoint, UMD, FaceFour, Lighting-2, Lighting-7) than considering only temporal metrics

Dataset	Metrics				
	Basic			Learned combined	
	d_E	d_{Cort_r}	d_{FFT}	D_2	D_3
SonyAIBO	0.305	0.308	**0.258***	0.308	**0.259**
MoteStrain	**0.121***	0.264	0.278	0.210	0.277
GunPoint	0.087	0.113	**0.027***	0.113	0.073
PowerCons	**0.370**	0.445	**0.315***	0.384	0.410
ECG5Days	0.203	0.153	**0.006***	0.153	0.156
SonyAIBOII	**0.141**	**0.142**	**0.128***	**0.142**	**0.142**
Coffee	0.250	**0***	0.357	**0***	**0***
BME	0.128	**0.059***	0.412	**0.059***	**0.078**
UMD	**0.185***	0.207	0.315	0.207	**0.185***
ECG200	0.120	**0.070***	0.166	**0.070***	**0.070***
Beef	**0.467**	**0.300***	**0.500**	**0.300***	**0.367**
DiatomSizeReduction	**0.065***	**0.075**	**0.069**	**0.075**	**0.075**
FaceFour	**0.216**	**0.216**	**0.239**	**0.216**	**0.205***
Lighting-2	**0.246**	**0.246**	**0.148***	**0.246**	**0.213**
Lighting-7	**0.425**	**0.411**	**0.315**	**0.411**	**0.288***
OliveOil	**0.133***	**0.133***	**0.200**	**0.133***	**0.133***

Table 3. Error rate of 1NN classifier for different metrics. D_2 is computed using d_E and d_{Cort_r}; D_3 uses the 3 basic metrics. The metric with the best performance for each dataset is indicated by a star (*) and the ones with equivalent performances are in bold.

(d_E, d_{Cort_r}). However, it does not always improve the results (GunPoint, PowerCons, ECG5Days). This might be caused by the fact that our framework is sensitive to the choice of the initial metric (L2-norm) or maybe, some steps in the algorithm should be improved to make the combination better.

5 Conclusion

For nearest neighbor time series classification, we propose to learn a metric as a combination of temporal and frequential metrics based on a large margin optimization process. The learned metric shows good performances on the conducted experimentations. For future work, we are looking for some improvements. **First**, the choice of the initial metric is crucial. It has been set here as the L2-norm of the pairwise space but a different metric could provide better *target* sets. Otherwise, using an iterative procedure (reusing D_w to generate new *target* sets and learn D_w again) could be another solution. **Second**, we note that the L1-norm on the "pull" term leads to sparcity. Changing it into a L2-norm could allow for non-sparse solutions and also extend the approach to non-linear metric combination functions thanks to the Kernel trick. **Finally**, we could extend this framework to multivariate, regression or clustering problems.

References

1. T.C. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, 2011.
2. H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE transactions on acoustics, speech, and signal processing*, 1978.
3. H. Ding, G. Trajcevski, and P. Scheuermann, "Querying and Mining of Time Series Data : Experimental Comparison of Representations and Distance Measures," in *VLDB*, 2008.
4. A. Douzal-Chouakria and C. Amblard, "Classification trees for time series," *Pattern Recognition journal*, 2011.
5. S. Lhermitte, J. Verbesselt, W.W. Verstraeten, and P. Coppin, "A comparison of time series similarity measures for classification and change detection of ecosystem dynamics," 2011.
6. K. Weinberger and L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, 2009.
7. M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.
8. R. Duin, M. Bicego, M. Orozco-alzate, S. Kim, and M. Loog, "Metric Learning in Dissimilarity Space for Improved Nearest Neighbor Performance," pp. 183–192.
9. A. Ibba, R. Duin, and W. Lee, "A study on combining sets of differently measured dissimilarities," in *Proceedings - International Conference on Pattern Recognition*, 2010, pp. 3360–3363.
10. C. Do, A. Douzal-Chouakria, S. Marié, and M. Rombaut, "Multiple Metric Learning for large margin kNN Classification of time series," *EUSIPCO*, 2015.
11. E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C.A. Ratanamahatana, "The UCR Time Series Classification/Clustering Homepage (www.cs.ucr.edu/~eamonn/time_series_data/)," 2011.
12. K. Bache and M. Lichman, "UCI Machine Learning Repository (http://archive.ics.uci.edu/ml)," 2013.
13. "LIG-AMA Machine Learning Datasets Repository (http://ama.liglab.fr/resourcestools/datasets/)," 2014.
14. C. Frambourg, A. Douzal-Chouakria, and E. Gaussier, "Learning Multiple Temporal Matching for Time Series Classification," *Advances in Intelligent Data Analysis XII*, 2013.
15. "IBM Cplex (http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/)," .
16. T. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," 1997.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Coarse-DTW: Exploiting Sparsity in Gesture Time Series

Marc Dupont[1,2] and Pierre-François Marteau[1]
[1] IRISA, Université de Bretagne Sud, Campus de Tohannic, Vannes, France
[2] Thales Optronique, 2 Avenue Gay Lussac, Elancourt, France

Abstract. Dynamic Time Warping (DTW) is considered as a robust measure to compare numerical time series when some *time elasticity* is required. Even though its initial formulation can be slow, extensive research has been conducted to speed up the calculations. However, those optimizations are not always available for multidimensional time series. In this paper, we focus on time series describing gesture movement, all of which are multidimensional. Our approch propose to speed up the processing by 1. adaptively downsampling the time series into *sparse* time series and 2. generalizing DTW into a version exploiting sparsity. Furthermore, the downsampling algorithm doesn't need to know the whole timeseries to function, making it a good candidate for streaming applications such as real-time gesture recognition.

1 Introduction

Among other measures, Dynamic Time Warping (DTW) has been widely popularized during the seventies with the advent of speech recognition systems [18], [14]. However, one of the main drawbacks of such a *time-elastic* measure is its quadratic computational complexity which, as is, prevents processing a very large amount of lengthy temporal data. Recent research has thus mainly focused on circumventing this complexity barrier. The original approach proposed in this paper is to cope directly and explicitly with the potential sparsity of the time series during their *time-elastic* alignment.

2 Previous work

DTW has seen speed enhancements in several forms; [14] and [5] reduce the search space by using a band or parallelogram; [1] introduced the concept of a sparse alignment matrix to dynamically reduce the search space without optimality loss. The dimensionality of the data can be reduced, such as in [21] and [7] who propose Piecewise Aggregate Approximation (PAA) to downsample the time series into segments of constant size, then handled by a DTW modification, PDTW [9]; further compressing can be obtained with Adaptive Piecewise Constant Approximation (APCA) [2]; or compression via symbolic representation of scalar points can be obtained with SAX [13]. Early abandoning strategies avoid useless calculation by computing cheap lower bounds: such as [20], [8] and [10], but the most powerful [8] is not readily available in a form available

for multidimensional time series. ID-DTW (Iterative Deepening DTW) [4] and FastDTW [16] use multi-resolution approximations, possibly with an early abandoning strategy; [15] and [17] have also proposed approaches mixing APCA with a lower bounding strategy. Additionally, some alternative elastic distance variants have been proposed, such as ERP [3] or TWED [12] with some gain in classification accuracy, but with no speed-up strategy designed so far.

Our method differs from the previous work as follows: first, it gives a novel way of producing a piecewise constant time series, especially interesting because of its simplicity and its potential use in streaming scenarios (the downsampled time series is produced as fast as the original one arrives); second, DTW is enhanced with a new weighting strategy to accept such downsampled time series and achieve the desired speed enhancement.

3 Presentation of Coarse-DTW

3.1 Sparse time series

The usual notion of a time series will be called here a *dense time series*. It represents a sequence (v_i) of points in $\mathbb{R}^d$, where d is the dimension. Such a time series is usually sampled at a regular interval.

By contrast, let a *sparse time series* be a pair of sequences (s_i) and (v_i) with the same length n:

$$\begin{aligned} s &: \{1, \ldots, n\} \to \mathbb{R}_+ \\ v &: \{1, \ldots, n\} \to \mathbb{R}^d \end{aligned} \tag{1}$$

Each v_i represents a multidimensional point (of dimension d) and each s_i is a number describing *how long the value v_i lasts*. We call this number s_i the *stay* of v_i. In the following, we will also denote a sparse time series as $\{(s_1, v_1), \ldots, (s_n, v_n)\}$.

For example, the 2D dense time series $\{(0.5, 1.2), (0.5, 1.2), (0.3, 1.5)\}$ is equivalent to the 2D sparse time series $\{(2, (0.5, 1.2)), (1, (0.3, 1.5))\}$. As another example, a dense time series (v_i), is exactly represented by the sparse time series with the same values v_i and all stays $s_i = 1$.

3.2 Coarse-DTW

The Coarse-DTW algorithm accepts two sparse time series: (s_i, v_i) of length n, and (t_j, w_j) of length m.

Algorithm 1 Coarse-DTW

```
1: procedure COARSE-DTW((s, v), (t, w))
2:     A = new matrix [0..n, 0..m]
3:     A[0, .] = A[., 0] = ∞ and A[0, 0] = 0
4:     for i = 1 to n do
5:         for j = 1 to m do
6:             A[i, j] = min( s_i.δ(v_i, w_j) + A[i−1, j],
7:                            t_j.δ(v_i, w_j) + A[i, j−1],
8:                            max(s_i, t_j).δ(v_i, w_j) + A[i−1, j−1] )
9:     return A[n, m]
```

The symbol δ represents any distance on $\mathbb{R}^d$. A common choice is $\delta(x, y) = \|x - y\|_2^2 = \sum_{k=1}^{d}(x_k - y_k)^2$.

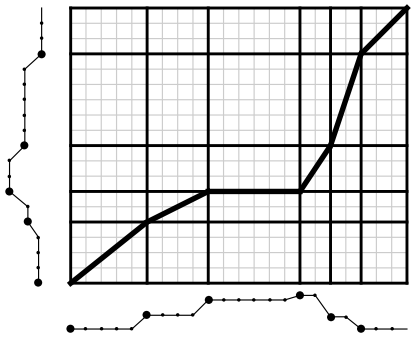


Fig. 1. A warping path in Coarse-DTW. We superimposed the sparse timeseries (bigger points) on top of their equivalent dense timeseries (smaller points). The coarse, thick grid is the Coarse-DTW matrix, whereas the underlying thin grid is the classical DTW cost matrix.

Coarse-DTW takes advantage of the sparsity in the time series to calculate costs efficiently. However, because the points last for different amount of time, we must adapt the classical DTW formulation in order to account for the stays s_i and t_j of each point into the aggregate cost calculation.

Obviously, when a point lasts for a long time, it should cost more than a point which lasts for a brief amount of time. For this reason, the pure cost $\delta(v_i, w_j)$ is multiplied by some quantity, called *weight*, linked to how long the points last, as in lines 6–8 of the algorithm. The goal of this subsection is to explain why we set those weights to s_i, t_j, and $\max(s_i, t_j)$ respectively.

The choice of weights s_i and t_j in lines 6 and 7 is motivated as follows: when we advance one time series without advancing the other, we want a lengthy point to cost more than a brief point. In the DTW constant-cost sub-rectangle, advancing the first time series is like following an horizontal subpath, whose aggregated cost would be $\delta(v_i, w_j)$ on each of its s_i cells. This sums up to $s_i.\delta(v_i, w_j)$, which

is why the weight is chosen to be s_i in line 6. An analog interpretation holds for a vertical subpath of t_j cells.

In a constant-cost sub-rectangle (of size $s_i \times t_j$), minimizing the aggregated cost of a path is equivalent with minimizing its number of cells, because all cells have the same cost. Furthermore, the minimal number of cells is exactly $\max(s_i, t_j)$. This would be the path followed by classical DTW. Hence, the weight is set to $\max(s_i, t_j)$ in line 8.

4 Downsampling

In this section, we seek to transform a dense time series (u_i) into a sparse time series (s_i, v_i); the goal is to detect when series "move a lot" and "are rather static", adjusting the number of emitted points accordingly.

Bubble downsampling can be described in a simple form as follows:

Algorithm 2 Bubble Downsampling

```
1:  procedure BUBBLE(v, ρ)                                  ▷ ρ ≥ 0
2:      i_center = 1                                        ▷ initialize bubble center
3:      v_center = v_1
4:      v_mean = v_1
5:      for i = 2 to n do
6:          Δv = δ(v_i, v_center)                           ▷ distance to center
7:          Δi = i − i_center                               ▷ find the stay
8:          if Δv ≥ ρ then                                  ▷ does the bubble "burst"?
9:              yield (Δi, v_mean)                          ▷ emit stay + point
10:             i_center = i                                ▷ update bubble center
11:             v_center = v_i
12:             v_mean = v_i
13:         else
14:             v_mean = (Δi × v_mean + v_i)/(Δi + 1)       ▷ update mean
15:     Δi = n − i_center + 1                               ▷ force bursting last bubble
16:     yield (Δi, v_mean)
```

The idea behind Bubble downsampling is based on the following approximation: consecutive values can be considered equal as long as they stay within a given radius ρ for the distance δ. We can picture a curve which makes bubbles along its path (see Fig. 3), hence the name. Concretely, the algorithm emits a sparse time series, where each stay is the number of consecutive points contained in a given bubble, and each value is the mean of the points in this bubble.

The parameter ρ represents the tradeoff between information loss and density. A large ρ emits few points, thus yielding a very sparse time series, but less accurate; a smaller ρ preserves more information at the expense of a lower downsampling ratio. The degenerate case $\rho = 0$ will output a clone of the original time series with no downsampling (all stays equal to 1). Because speed is a direct consequence of sparsity in Coarse-DTW, a good middle value for ρ must

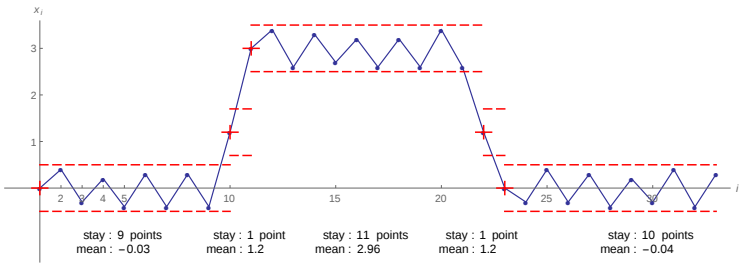


Fig. 2. Bubble downsampling applied on a 1D time series (blue, solid) with $\rho = 0.5$. The 1-bubbles are represented by their 1-centers (red crosses) and their 1-boundaries (red, dashed lines). The sparse time series emitted is $\{(9, -0.03), (1, 1.2), (11, 2.96), (1, 1.2), (10, -0.04)\}$.

be found, so that time series are as sparse as possible while retaining just the right amount of information.

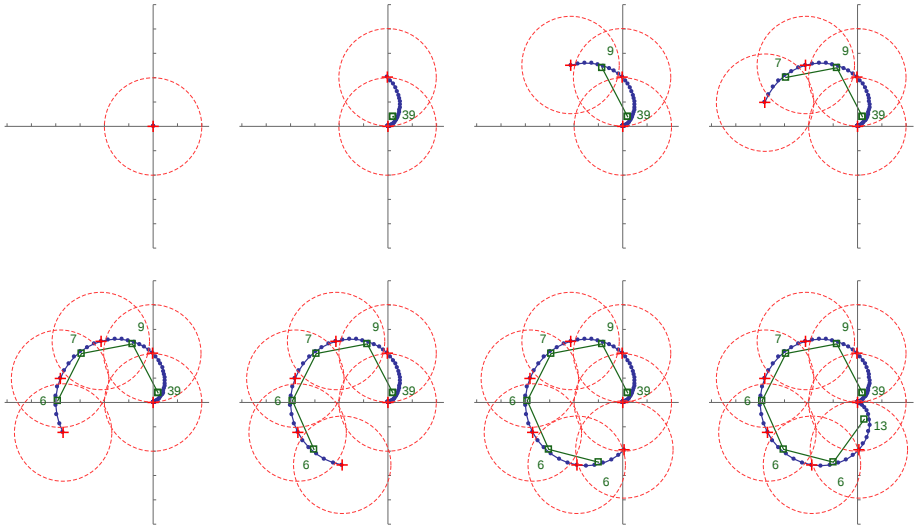


Fig. 3. Bubble downsampling progressively applied on a 2D time series (outer blue line with dots) with $\rho = 2.0$, along with the sparse time series emitted (inner green line with squares). Again, the 2-bubbles are represented by their 2-centers and their 2-boundaries (red crosses and dashed circles). Numbers indicate the stays. Notice how stays take into account the slowness at the beginning of the signal.

5 Optimizations on Coarse-DTW

DTW suffers from a slow computation time if not implemented wisely. For this reason, several optimizations have been designed [8]. The next optimizations we

considered are called *lower bounds*, designed to early-abandon computations in a k-Nearest Neighbor scenario.

The first lower bound $\mathrm{LB}_{\mathrm{Kim}}$ [20] is transposable to Coarse-DTW: as with 1D time series, the first and last pairs of points will always be matched together as long as the timeseries have each at least two points. First, the cost of matching the first points is: $\max(s_1, t_1).\delta(v_1, w_1)$ because the first matching is done diagonally ($A[0,1] = A[1,0] = \infty$). Then, the cost of matching the last points is $\min(s_n, t_m, \max(s_n, t_m)) \,.\, \delta(v_n, w_m)$. Hence, the lower bound is written:

$$\text{Coarse-DTW}(v, w) \geq \max(s_1, t_1).\delta(v_1, w_1) + \min(s_n, t_m, \max(s_n, t_m)).\delta(v_n, w_m) \tag{2}$$

The second lower bound can be evaluated several times as DTW progresses: for any row i, the minimum of all cells $A[i,.]$ is a lower bound to the DTW result. Indeed, this result is the last cell of the last row, and the sequence mapping a row i to $\min_j A[i,j]$ is increasing, because the costs are positive. Hence, during each outer loop iteration (i.e., on index i), we can store the minimum of the current row and compare it to the best-so-far for possibly early abandoning. This can be transposed directly to Coarse-DTW without additional modifications.

Finally, probably the most powerful lower bound for unidimensional time-series, known as $\mathrm{LB}_{\mathrm{Keogh}}$ [8], is based upon the calculation of an envelope; however this calculation is not trivially transferable to the case of multidimensional time series simply by generalizing the uni-dimensional equations. Thus, we will unfortunately not consider it in our study.

6 Results

6.1 DTW vs. Coarse-DTW in 1-NN classification

We considered the classification accuracy and speed of three multidimensional labeled time series datasets describing gesture movement. The classifier is 1-NN and we enabled all optimizations described earlier that apply to multidimensional time series, namely: $\mathrm{LB}_{\mathrm{Kim}}$ and early abandoning on the minima of rows. We report only the classification time; learning time is zero because no processing is required. Each dataset is run once with DTW and several times with Coarse-DTW, each time with a different value for the downsampling radius ρ.

MSRAction3D [11] time series have 60 dimensions (twenty 3D joints) which we classified by cross-validating all 252 combinations of 5 actors in training and 5 in test. uWaveGestureLibrary_[XYZ] comes from the UCR time series database [6]; it can be considered as three independent uni-dimensional datasets, but we rather used it here as a single set of 3-dimensional time series, which makes 1-NN DTW classification fall from 1D errors of respectively 27.3 %, 36.6 % and 34.2 % down to only 2.8 % as a 3D time series. Character Trajectories [19] comes from the UCI database and describes trajectories of character handwriting; they were first resampled to have all the same size (204 data points, size of the longest sequence in the dataset).

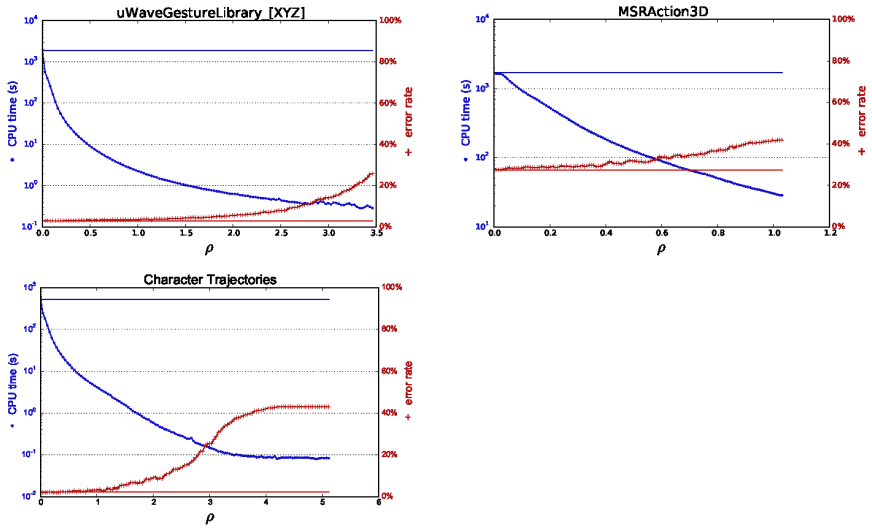


Fig. 4. 1-NN classification time and error rate of Coarse-DTW as ρ increases. For comparison, DTW results are shown as horizontal bars (independent of ρ). Speedups from 10x to 1000x are obtained without sensible accuracy degradation.

7 Conclusions

Not only have we transposed DTW into Coarse-DTW, a version accepting sparse time series, but we have also developed Bubble, an extremely efficient algorithm to generate such sparse time series from regular ones. By coupling those two mechanisms, we found out that time series can be classified much faster in nearest-neighbor classification; the user can reach the desired tradeoff between speed and accuracy, by tuning the parameter ρ in the downsampling algorithm. Gesture timeseries produce smooth time series which present a considerable ability to be downsampled, producing good results in classification speedup.

In order to learn ρ from the data, experiments above suggest a simple method: first, set an acceptable threshold on the error (e.g. +2% w.r.t. regular DTW error) ; then, select the ρ whose error is under this threshold and classification speed is fastest.

Although we didn't cover it in our test scenarios, it is worth highlighting that Coarse-DTW and Bubble are directly applicable to a streaming scenario: indeed, Bubble doesn't need to know the whole timeseries before emitting sparse points. As a consequence, it could be a great way to save CPU time and battery life in an embedded gesture recognition setup.

8 Acknowledgements

This study was co-funded by the ANRT agency and Thales Optronique SAS, under the CIFRE convention 2013/0932.

References

1. Ghazi Al-Naymat, Sanjay Chawla, and Javid Taheri. Sparsedtw: A novel approach to speed up dynamic time warping. In *Proceedings of the Eighth Australasian Data Mining Conference - Volume 101*, AusDM '09, pages 117–127, Darlinghurst, Australia, Australia, 2009. Australian Computer Society, Inc.
2. Kaushik Chakrabarti, Eamonn Keogh, Sharad Mehrotra, and Michael Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Trans. Database Syst.*, 27(2):188–228, June 2002.
3. L. Chen and R. Ng. On the marriage of lp-norm and edit distance. In *Proceedings of the 30th International Conference on Very Large Data Bases*, pages 792–801, 2004.
4. Selina Chu, Eamonn J. Keogh, David M. Hart, and Michael J. Pazzani. Iterative deepening dynamic time warping for time series. In Robert L. Grossman, Jiawei Han, Vipin Kumar, Heikki Mannila, and Rajeev Motwani, editors, *Proceedings of the Second SIAM International Conference on Data Mining, Arlington, VA, USA, April 11-13, 2002*, pages 195–212. SIAM, 2002.
5. F. Itakura. Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):67–72, Feb 1975.
6. E. J. Keogh, X. Xi, L. Wei, and C.A. Ratanamahatana. The UCR time series classification-clustering datasets, 2006. http://wwwcs.ucr.edu/ eamonn/time_series_data/.
7. Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *JOURNAL OF KNOWLEDGE AND INFORMATION SYSTEMS*, 3:263–286, 2000.
8. Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, March 2005.
9. Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 285–289, New York, NY, USA, 2000. ACM.
10. Daniel Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition*, 42(9):2169 – 2180, 2009.
11. W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In IEEE CS Press, editor, *Proc. IEEE Int'l Workshop on CVPR for Hum. Comm. Behav. Analysis*, pages 9–14, 2010.
12. P. F. Marteau. Time warp edit distance with stiffness adjustment for time series matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):306–318, 2008.
13. Pranav Patel, Eamonn Keogh, Jessica Lin, and Stefano Lonardi. Mining motifs in massive time series databases. In *In Proceedings of IEEE International Conference on Data Mining (ICDM'02*, pages 370–377, 2002.
14. H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *Proceedings of the 7th International Congress of Acoustic*, pages 65–68, 1971.
15. Yasushi Sakurai, Masatoshi Yoshikawa, and Christos Faloutsos. Ftw: Fast similarity search under the time warping distance. In *Proceedings of the Twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '05, pages 326–337, New York, NY, USA, 2005. ACM.

16. Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, October 2007.
17. Yutao Shou, Nikos Mamoulis, and David W. Cheung. Fast and exact warping of time series using adaptive segmental approximations. *Mach. Learn.*, 58(2-3):231–267, February 2005.
18. V. M. Velichko and N. G. Zagoruyko. Automatic recognition of 200 words. *International Journal of Man-Machine Studies*, 2:223–234, 1970.
19. Ben H Williams, Marc Toussaint, and Amos J Storkey. *Extracting motion primitives from natural handwriting data*. Springer, 2006.
20. Sang wook Kim, Sanghyun Park, and Wesley W. Chu. An index-based approach for similarity search supporting time warping in large sequence databases. In *In ICDE*, pages 607–614, 2001.
21. Byoung-Kee Yi and Christos Faloutsos. Fast time sequence indexing for arbitrary lp norms. In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB '00, pages 385–394, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Preliminary Experimental Analysis of Reservoir Computing Approach for Balance Assessment

Claudio Gallicchio[1], Alessio Micheli[1], Luca Pedrelli[1], Federico Vozzi[2], and Oberdan Parodi[2]

[1] Department of Computer Science, University of Pisa,
Largo B. Pontecorvo 3, Pisa, Italy

[2] IFC-CNR Pisa, Via Moruzzi 1, Pisa, Italy

Abstract. Evaluation of balance stability in elderly people is of prominent relevance in the field of health monitoring. Recently, the use of Wii Balance Board has been proposed as valid alternative to clinical balance tests, such as the widely used Berg Balance Scale (BBS) test, allowing to measure and analyze static features such as the duration or the speed of assessment of patients' center of pressure. In an innovative way, in this paper we propose to take into consideration the whole temporal information generated by the balance board, analyzing it by means of dynamical neural networks. In particular, using Recurrent Neural Networks implemented according to the Reservoir Computing paradigm, we propose to estimate the BBS score from the temporal data generated by the execution of one simple exercise on the balance board. Preliminary experimental assessments of the proposed approach on a real-world dataset show promising results.

Keywords: Reservoir Computing, Learning with Temporal Data, Balance Assessment

1 Introduction

A sedentary lifestyle is a risk factor for the development of many chronic illnesses. The common physiological aging causes a decrease of global functional abilities: one of the most important is balance disorder [16]. The control of balance is complex, with a strong integration and coordination of multiple body elements including visual, auditor and motor systems [9]. A comprehensive clinical assessment of balance is important for both diagnostic and therapeutic reasons in clinical practice [4, 17]. The Berg Balance Scale (BBS) test is considered the gold standard assessment of balance with small intra-inter rater feasibility and good internal validity. The work in [3] assessed the validity of the BBS by examining how scale scores are related to clinical judgments, laboratory measures of postural sway and external criteria reflecting balancing ability. Furthermore, scores could predict falls in the elderly, and how they are related to motor and functional performance in stroke patients. The Berg's utility includes grading different patients' balance abilities, monitoring functional balance over time and evaluating patients responses to different protocols of treatment [18]. Based on a test of 14 exercises/items, BBS is performance-based and has a scale of 0-4

(clinician assigned) score for each item, with a maximum overall score of 56. Within the scopes of the DOREMI European project (GA 611650), a technological platform to support and motivate older people to perform physical activity is under development, aiming at reducing sedentariness, cognitive decline and malnutrition, promoting an improvement of quality of life and social inclusion. One of the element of DOREMI platform is a smart carpet, based on the use of Nintendo Wii Balance Board (WBB), able to gather information pertaining to users' weight distribution at the four corners of the board. Such tool allows to design an automatic system for balance assessment through the daily repetition of one simple BBS exercise. This type of analysis, done by users at medical facilities or, remotely, at their own houses, can help clinicians in the evaluation of older people equilibrium and in control of its evolution.
The use of the WBB is motivated by the fact that it represents a low-cost and portable tool, recently successfully adopted for problems related to standing posture correction [14] and for training standing balance in the elderly [19]. Interestingly, the WBB has been validated in comparison with gold standard force platforms [13] in its reliability to track users' balance parameters, such as the center of pressure path length and velocity [5]. However, it is worth to observe that the whole signal time-series generated by the WBB potentially contains a richer information than such static parameters. Thereby, in this paper we propose to analyze the data generated by WBB using Recurrent Neural Networks (RNNs), which are learning models suitable for catching and processing dynamic knowledge from noisy temporal information. In particular, we considered the problem of estimating the BSS score of a patient using in input the temporal information generated by the execution of one simple BBS exercise on the WBB. This approach potentially allows to avoid the need to repeat all the 14 BBS exercises for new patients. An alternative approach in [15] tries to estimate the BBS score of a patient using information extracted from a tri-axial accelerometer placed on the lower back during the execution of some items of the BBS. Such approach, however, adopts a solution which is more intrusive for the patient. At the best of our knowledge, our work represents the first attempt at estimating the BBS score directly from the temporal data generated while the patient performs a simple balance exercise in an non-intrusive way using an external device.

2 Balance Assessment with RC

A measurement campaign has been conducted on 21 volunteers, aged between 65 and 80 years. We measured the weight signal produced by the WBB at the 4 corners of the board sampled at 5 Hz during the execution of the exercise # 10 in the BBS test, i.e. *turn to look behind*, selected for its simple execution and short duration ($\approx$ 10 seconds). To take into account for possible variations in the exercise executions, for each patient we recorded data from a number of maximum 10 repetitions of the exercise. We therefore obtained a *Balance dataset* for a regression task on sequences, containing couples of the type $(\mathbf{s}, y_{tg})$, where $\mathbf{s}$ is the 4-dimensional input sequence of users' weight values recorded by the WBB

during the exercise and y_{tg} is the target BBS score (over all the 14 exercises) of the corresponding patient, representing the ground-truth evaluated by a clinician during the campaign. For performance assessment we adopted the Mean Absolute Error (MAE) of the BBS score estimation provided by the learning models. It is worth noticing that the Balance dataset contains an outlier patient with BBS score of 24, which has been discarded for performance evaluation.
We model the dynamics of the temporal data involved by the balance evaluation task by dynamical neural networks models within the class of RNNs. In particular, we adopt the Reservoir Computing (RC) approach [12] for RNN modeling, and take into consideration the Leaky Integration Echo State Network (LI-ESN) [11, 10], a state-of-the-art model for efficient learning in sequential/temporal domains, which has proved to be particularly suitable in dealing with the nature of the input data originated from sensors [1, 2]. LI-ESNs implement discrete time dynamical systems, and consist of two main components, a dynamical reservoir, which realizes a recurrent encoding of the input history and provides the system with a memory of the past [6], and a static readout which computes the output. A LI-ESN is composed of an input layer with N_U units, a recurrent non-linear reservoir layer with N_R sparsely connected units, and a linear readout layer with N_Y units. At each time step t, the reservoir computes a state $\mathbf{x}(t) \in \mathbb{R}^{N_R}$ according to a state transition function $\mathbf{x}(t) = (1-a)\mathbf{x}(t-1)+a\tanh(\mathbf{W}_{in}\mathbf{u}(t)+\hat{\mathbf{W}}\mathbf{x}(t-1))$, where $\mathbf{u}(t) \in \mathbb{R}^{N_U}$ is the input at time step t, $\mathbf{W}_{in} \in \mathbb{R}^{N_R \times N_U}$ is the input-to-reservoir weight matrix, $\hat{\mathbf{W}} \in \mathbb{R}^{N_R \times N_R}$ is the recurrent reservoir weight matrix, and $a \in [0, 1]$ is the leaking rate parameter that controls the speed of the reservoir dynamics [11, 12]. For sequence-to-element regression tasks in which an output value is required in correspondence of an entire input sequence, the use of a mean state mapping function has proved to be effective [7, 8]. Accordingly, given an input sequence of length n, $\mathbf{s} = [\mathbf{u}(1), \ldots, \mathbf{u}(n)]$, we average the state activation over the steps of the input sequence, i.e. $\chi(\mathbf{s}) = \frac{1}{n}\sum_{t=1}^{n}\mathbf{x}(t)$. Then, the readout is applied to compute the output of the model $\mathbf{y}(\mathbf{s}) \in \mathbb{R}^{N_Y}$ by a linear combination of the elements in $\chi(\mathbf{s})$, i.e. $\mathbf{y}(\mathbf{s}) = \mathbf{W}_{out}\chi(\mathbf{s})$, where $\mathbf{W}_{out} \in \mathbb{R}^{N_Y \times N_R}$ is the readout-to-reservoir weight matrix. The readout is the only LI-ESN component that is trained, typically by efficient linear methods, e.g. pseudo-inversion and ridge regression [12]. The reservoir is left untrained after initialization under the constraints of the *echo state property* (ESP) [10, 12, 6]. A reservoir initialization condition related to the spectral radius of $\hat{\mathbf{W}}$ is often used in literature and is adopted in this paper, i.e. $\rho((1-a)\mathbf{I} + a\hat{\mathbf{W}})) < 1$ (see e.g. [12, 6] for details).

3 Experimental Results

The experimental analysis presented in this paper aimed at preliminarily assessing the generalization performance of the proposed RC approach. At the same time, in order to reduce the patients' effort for future data gathering campaigns, we were also interested in empirically analyzing the trade-off between the number of exercise repetitions for each patient required for training and the predictive performance that can be achieved. Accordingly, we took into consideration two

experimental settings. In the first experimental setting, the Balance dataset was split in a training set, containing data from 17 patients ($\approx 80\%$ of the total), and an external test set for performance assessment, containing data from 4 patients ($\approx 20\%$ of the total, chosen in order to represent a uniform sampling in the range of possible BBS target values). We considered LI-ESNs with reservoir dimension in $N_R \in \{100, 200, 500\}$, 10% of reservoir units connectivity, leaky parameter $a \in \{0.1, 0.3, 0.5, 0.7, 1\}$ and spectral radius $\rho = 0.99$. For each reservoir hyper-parametrization, we independently generated 5 reservoir guesses, averaging the results over such guesses. For readout training we used pseudo-inversion and ridge regression with regularization $\lambda_r \in \{10, 1, 0.7, 0.5, 0.3, 0.1, 0.01, 0.001\}$. The values of the reservoir hyper-parameters and readout regularization were chosen by model selection, adopting a 4-fold cross validation scheme over the training set. The selected LI-ESN resulted in a very good predictive performance, with a test MAE of 4.25 ± 0.39, which outperforms the results in [15] for patients within a corresponding age range. Such results appear promising, also considering the tolerance in the ground-truth data due to human observations. Moreover, we observed that the test error is higher for patients with lower BBS target scores, which correspond to a less sampled region in the input space.
We also conducted a preliminary empirical investigation in order to evaluate how the performance of the proposed LI-ESN approach scales with the number of available training data for each patient. Accordingly, we uniformly split the Balance dataset into groups containing sequences pertaining to 3 patients each, according to a 7-fold cross validation scheme, progressively reducing the number of training sequences for each patient. For this second experimental setting, we restricted to the case of LI-ESNs with $N_R = 100$ reservoir units, whereas all the others reservoir hyper-parameters and readout regularization values were selected (for each fold) on the validation set, considering the same range of values as in the case of the first experimental setting. Fig. 1 shows the MAE achieved by LI-ESNs on the validation set, for decreasing number of available training sequences. Results show that the validation performance is approximately stable for a number of training sequences per patient in the range of 10-4, while it gets rapidly worse as less than 4 training sequences per patient are used.

4 Conclusions

We have proposed an approach for assessing the balance abilities of elderly people based on RC networks, used for temporal processing of data recorded by a WBB during the execution of a simple BBS exercise, with the major advantage of automatically evaluating the BBS score using only 1 of the 14 exercises. The preliminary experimental analysis on a real-world dataset showed that our approach is able to achieve a very good predictive performance, up to ≈ 4 points of discrepancy in the BBS score with respect to the gold standard, which is good also considering both the use of a single BBS item and the tolerance typical of any subjective assessment scale. Overall, the possibility to infer the BBS scores with a good performance starting from the signal of a single BBS exercise shows

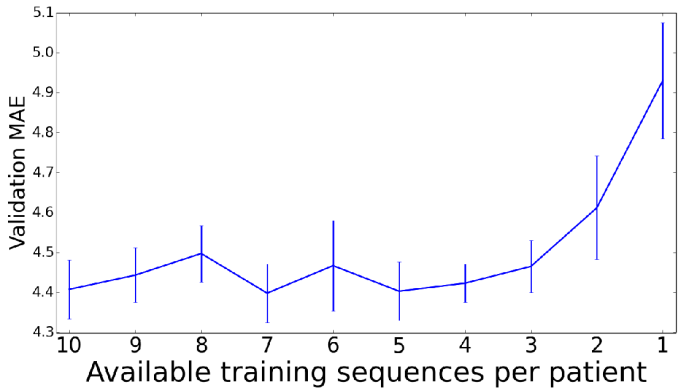


Fig. 1. Validation MAE (and standard deviation) achieved by LI-ESNs on the Balance dataset for a decreasing number of training sequences available for each patient.

the potentiality of our idea of exploiting the entire curve of the signal stream as a rich source of information for the evaluation of balance assessment. We also addressed the problem of evaluating the trade-off between the predictive performance and the number of exercise repetitions required for training the RC networks. A moderate number of repetitions in the training set turned out to be already sufficient for achieving a good performance. This aspect is of particular interest in view of minimizing the effort required for the collection of an adequate and sufficiently sampled dataset for balance estimation, as the repeated execution of BBS exercises by elderly people could be onerous. The results illustrated in this preliminary study have a potential utility by themselves, for the development of a balance estimation tool, and they will be eventually exploited within the purposes of the DOREMI project as a part of a larger health monitoring system aiming at improving elderly quality of life and active aging.

Acknowledgments. The work was funded by a grant from DOREMI project (FP7-ICT-2013, GA no. 611650). We would like to acknowledge Dr. Sara Lanzisera, Dr. Cristina Laddaga (ASL5, Pisa), Dr. Andrea Bemi (Istituto Superiore di Istruzione C. Piaggia, Viareggio) and Dr. Franca Giugni (CNR-IFC) for their valuable inputs, support and effort during the preparation and execution of tests, and Dr. Luigi Fortunati, Dr. Filippo Palumbo and Dr. Erina Ferro (CNR-ISTI) also for the realization of the middleware. We would also like to acknowledge all the test participants for their support and active participation in these activities.

References

1. Amato, G., Bacciu, D., Broxvall, M., Chessa, S., Coleman, S., Rocco, M.D., Dragone, M., Gallicchio, C., Gennaro, C., ant T.M. McGinnity, H.L., Micheli, A., Ray, A., Renteira, A., Saffiotti, A., Swords, D., Vairo, C., Vance, P.: Robotic ubiquitous

cognitive ecology for smart homes. Journal of Intelligent & Robotic Systems pp. 1–25 (2015)
2. Bacciu, D., Barsocchi, P., Chessa, S., Gallicchio, C., Micheli, A.: An experimental characterization of reservoir computing in ambient assisted living applications. Neural Computing and Applications 24 (6), 1451–1464 (2014)
3. Berg, K.O., Wood-Dauphinee, S.L., Williams, J.I., Maki, B.: Measuring balance in the elderly: validation of an instrument. Canadian journal of public health= Revue canadienne de sante publique 83, S7–11 (1991)
4. Bloem, B.R., Visser, J.E., Allum, J.H.: Movement disorders - handbook of clinical neurophysiology. Elsevier (2009)
5. Clark, R.A., Bryant, A.L., Pua, Y., McCrory, P., Bennell, K., Hunt, M.: Validity and reliability of the nintendo wii balance board for assessment of standing balance. Gait & posture 31(3), 307–310 (2010)
6. Gallicchio, C., Micheli, A.: Architectural and markovian factors of echo state networks. Neural Networks 24(5), 440–456 (2011)
7. Gallicchio, C., Micheli, A.: Tree echo state networks. Neurocomputing 101, 319–337 (2013)
8. Gallicchio, C., Micheli, A.: A preliminary application of echo state networks to emotion recognition. In: Proceedings of EVALITA 2014. pp. 116–119 (2014)
9. Horak, F.B., Wrisley, D.M., Frank, J.: The balance evaluation systems test (bestest) to differentiate balance deficits. Physical therapy 89(5), 484–498 (2003)
10. Jaeger, H., Haas, H.: Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science 304(5667), 78–80 (2004)
11. Jaeger, H., Lukoševičius, M., Popovici, D., Siewert, U.: Optimization and applications of echo state networks with leaky-integrator neurons. Neural Networks 20(3), 335–352 (2007)
12. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review 3(3), 127–149 (2009)
13. Maki, B.E., Holliday, P.J., Topper, A.K.: A prospective study of postural balance and risk of falling in an ambulatory and independent elderly population. Journal of gerontology 49(2), M72–M84 (1994)
14. Shih, C.H., Shih, C.T., Chu, C.L.: Assisting people with multiple disabilities actively correct abnormal standing posture with a nintendo wii balance board through controlling environmental stimulation. Research in developmental disabilities 31(4), 936–942 (2010)
15. Simila, H., Mantyjarvi, J., Merilahti, J., Lindholm, M., Ermes, M.: Accelerometry-based berg balance scale score estimation. Biomedical and Health Informatics, IEEE Journal of 18(4), 1114–1121 (2014)
16. Tinetti, M.E.: Performance-oriented assessment of mobility problems in elderly patients. Journal of the American Geriatrics Society 34(2), 119–126 (1986)
17. Visser, J.E., Carpenter, M.G., van der Kooij, H., Bloem, B.R.: The clinical utility of posturography. Clinical Neurophysiology 119(11), 2424–2436 (2008)
18. Wood-Dauphinee, S.L., Berg, K.O., Bravo, G.: The balance scale: Responding to clinically meaningful changes. Canadian journal of rehabilitation 10, 35–50 (1997)
19. Young, W., Ferguson, S., Brault, S., Craig, C.: Assessing and training standing balance in older adults: a novel approach using the nintendo wiibalance board. Gait & posture 33(2), 303–305 (2011)

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Estimating Dynamic Graphical Models from Multivariate Time-Series Data

Alexander J. Gibberd and James D.B. Nelson
Department of Statistical Science, University College London, Gower Street, London, WC1E 6BT

Abstract. We consider the problem of estimating dynamic graphical models that describe the time-evolving conditional dependency structure between a set of data-streams. The bulk of work in such graphical structure learning problems has focused in the stationary i.i.d setting. However, when one introduces dynamics to such models we are forced to make additional assumptions about how the estimated distributions may vary over time. In order to examine the effect of such assumptions we introduce two regularisation schemes that encourage piecewise constant structure within Gaussian graphical models. This article reviews previous work in the field and gives an introduction to our current research.

1 Introduction

As the current data explosion continues, governments, business, and academia are now not only harvesting more data points but also measuring an ever-increasing number of variables. The complex systems represented by such datasets arise in many socio-scientific domains, such as: cyber-security, neurology, genetics and economics. In order to understand such systems, we must focus our analytic and experimental resources on investigating the most important relationships. However, searching for significant relationships between variables is a complex task. The number of possible graphs that encode such dependencies between variables becomes exponentially large as the number of variables increase. Such computational issues are only compounded when such graphs vary over time.

From a statistical estimation viewpoint, the significance of a model component can often be viewed in terms of a model selection problem. Generally, one may construct an estimate of model fit (a lower score implies better fit) $L(M, \boldsymbol{\theta}, \boldsymbol{Y})$, relating a given model $M \in \mathcal{M}$ and parameters $\boldsymbol{\theta} \in \Theta(M)$ to some observed data $\boldsymbol{Y} \in \Omega$. Additionally, to account for differences in perceived model complexity one should penalise this by a measure of complexity $R(M, \boldsymbol{\theta})$ (larger is more *complex*). An optimal model and identification of parameters can be found through balancing the two terms, i.e:

$$(\hat{M}, \hat{\boldsymbol{\theta}}) = \underset{M \in \mathcal{M}, \boldsymbol{\theta} \in \Theta(M)}{\arg\min} \left[L(M, \boldsymbol{\theta}, \boldsymbol{Y}) + R(M, \boldsymbol{\theta}) \right] . \tag{1}$$

In statistics such a formulation is referred to as an M-estimator [15], however such frameworks are popular across all walks of science [2], for example, maximum-likelihood (ML), least-squares, robust (Huber loss), penalised ML estimators can

all be discussed in this context. The principle idea is to suggest a mathematical (and therefore can be communicated objectively) statement to the effect of Occam's Razor, whereby given similar model-fit, one should prefer the simpler model. Depending on the specification of the functions $L(\cdot)$ and $R(\cdot)$ and associated model/parameter spaces, the problem in (1) can be either very easy or difficult (for example, are the functions smooth, convex, etc).

In the next section we introduce the canonical *Gaussian graphical model (GGM)*, and study the estimation of such models within the M-estimation framework. This lays the foundations for our proposed dynamical extensions. We conclude with an example of an estimated dynamic GGM, some recovery properties of our estimators and discuss future research directions.

2 Gaussian graphical models

A Gaussian graphical model is a generative model which encodes the conditional dependency structure between a set of P variables $(Y_{1,\dots}Y_P) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ as a graph $G(V, E)$. For now we will discuss the traditional i.i.d setting, in Section (4) we will demonstrate ways in which we may relax the assumption of the distribution being identical over time.

In the standard case, the vertex set $V = \{1, \dots, P\}$ identifies variables and the edge set $E = \{(i, j), \dots (l, m)\}$ contains an edge if variables are conditionally dependent, specifically if $(i, j) \notin E$ we can decompose a joint distribution as $P(Y_i, Y_j | Y_{V\setminus\{i,j\}}) = P(Y_i | Y_{V\setminus\{i,j\}}) P(Y_j | Y_{V\setminus\{i,j\}})$. The aim of our work is to estimate an edge-set that appropriately represents a given data-set. Within the GGM setting, learning such representations does not only provide insight by suggesting key dependencies, but also specifies a robust probabilistic model which we can use for tasks such as anomaly detection.

It is well known that the edges in a GGM are encoded by non-zero off-diagonal entries within the precision matrix $\boldsymbol{\Theta} := \boldsymbol{\Sigma}^{-1}$, specifically $(i, j) \in E \iff \Theta_{i,j} \neq 0$ (see [12] for details). Learning the structure within the GGM can then be linked with the general framework of (1) through a ML or Maximum a-posteriori (MAP) paradigm. Assuming T observations $\boldsymbol{Y} \in \mathbb{R}^{P \times T}$ drawn as i.i.d samples the model fit function $L(\cdot)$ can be related to the likelihood specified by the multivariate normal. Typically, one prefers to work with the log-likelihood, which if we assume $\boldsymbol{\mu} = \mathbf{0}$ (we assume this throughout) is given by:

$$\log(P(\boldsymbol{Y}|\boldsymbol{\Theta}))/T = \frac{1}{2}\log\det(\boldsymbol{\Theta}) - \frac{1}{2}\mathrm{trace}(\hat{\boldsymbol{S}}\boldsymbol{\Theta}) - \frac{P}{2}\ln(\pi)\ ,$$

where $\hat{\boldsymbol{S}} = \boldsymbol{Y}\boldsymbol{Y}^\top/T$. Taking $L(\cdot) = -\log\det(\boldsymbol{\Theta}) + \mathrm{trace}(\hat{\boldsymbol{S}}\boldsymbol{\Theta})$ gives (in the setting where $T > P$) a well-behaved smooth, convex function describing how well a given parameterisation $\boldsymbol{\Sigma}$ represents the data $\boldsymbol{Y}$.

3 Penalising complexity

If one considers Eq. (1) with the function $R(\cdot) = 0$, i.e. no complexity penalty, then the precision matrix estimator $\hat{\boldsymbol{\Theta}} := \arg\min_{\{\boldsymbol{\Theta} \succeq 0\} \in \mathbb{R}^{P \times P}} \big[-\log\det(\boldsymbol{\Theta}) + \mathrm{trace}(\hat{\boldsymbol{S}}\boldsymbol{\Theta}) \big]$ demonstrates some undesirable properties indicative of over-fitting:

- The estimator exhibits large variance when $T \approx P$ and is very sensitive to changes in observations leading to poor generalisation performance.
- In the high-dimensional setting ($P > T$), the sample estimator is rank deficient ($\mathrm{rank}(\hat{\boldsymbol{S}}) < P$) and there is no unique estimator $\hat{\boldsymbol{\Theta}}$.

In order to avoid estimating a complete GGM graph (where all vertices's are connected to each other), one must actively select edges according to some criteria. In the asymptotic setting where $T \gg P$ we can test for the significance of edges by considering the asymptotic distribution of the empirical partial correlation coefficients ($\rho_{ij} = -\Theta_{ij}/\Theta_{ii}^{1/2}\Theta_{jj}^{1/2}$) [4]. However, such a procedure cannot be performed in the high-dimensional setting (this is important for the dynamical extensions, see Sec. 4) as we require that the empirical estimate be positive semi-definite.

An alternative approach to testing is to consider prior knowledge about the number of edges in the graph. If we assume a flat prior on the model $\mathcal{M}$ and parameters $\boldsymbol{\Theta}(\mathcal{M})$, maximising the approximate posterior probability over models $P(\mathcal{M}|\boldsymbol{Y})$, then leads to the Bayesian information criterion for GGM [5]: $BIC(\hat{\boldsymbol{\Theta}}_{ML}) = N(-\log\det(\hat{\boldsymbol{\Theta}}_{ML}) + \mathrm{trace}(\hat{\boldsymbol{S}}\hat{\boldsymbol{\Theta}}_{ML})) + \hat{p}\log(N)$, where $\hat{p}$ is given by the number of unique non-zeros within the ML estimated precision matrix $\hat{\boldsymbol{\Theta}}_{ML}$. Unfortunately, interpreting BIC under the framework in Eq. (1), we find the complexity penalty $R() = \hat{p}\log(N)$ is non-convex ($\hat{p} \propto \|\boldsymbol{\Theta}\|_0$ it basically counts the number of estimated edges). In order to arrive at a global minima an exhaustive search over the model space (all possible graphs $\mathcal{O}(2^{P^2})$) is required.

Alternatively, one can place an informative prior on the parameterisation and model (i.e. the GGM sparsity pattern) to encourage a parsimonious representation. One popular approach [6,11,17,20] is to place a Laplace type prior on the precision matrix in an effort to directly shrink off-diagonal values. Whilst one could choose to perform full Bayesian inference for the posterior $P(\boldsymbol{\Theta}|\boldsymbol{Y}, \gamma)$ (as demonstrated in [17]), a computationally less demanding approach is to perform MAP estimation resulting in the *graphical lasso* problem [6]:

$$\hat{\boldsymbol{\Theta}}_{GL} := \underset{\boldsymbol{\Theta} \succ 0}{\arg\min} \big[-\log\det(\boldsymbol{\Theta}) + \mathrm{trace}(\hat{\boldsymbol{S}}\boldsymbol{\Theta}) + (\gamma/N)\|\boldsymbol{\Theta}\|_1 \big] \,, \tag{2}$$

where $\|\boldsymbol{\Theta}\|_1 = \sum_{1 \leq i,j \leq P} |\Theta_{i,j}|$ is the ℓ_1 norm of $\boldsymbol{\Theta}$. The graphical lasso problem can yet again be interpreted within the general framework, except this time with $R(\cdot) = (\gamma/N)\|\boldsymbol{\Theta}\|_1$. Unlike BIC this complexity penalty is convex thus we can quickly find a global minima.

4 Introducing dynamics

In this section we extend the basic GGM model to a dynamic setting whereby the estimated graph is permitted to change as a function of time. Consider the P-variate time-series data $\boldsymbol{Y} \in \mathbb{R}^{P \times T}$ as before, however, we now permit the generative distribution to be a function of time, i.e:

$$(Y_1^t, \dots Y_P^t) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}^t) \, , \tag{3}$$

the challenge is now to learn a GGM via $(\boldsymbol{\Sigma}^t)^{-1}$ for each time point $t = 1, \dots, T$. Clearly such a model is far more flexible than the identically distributed version, instead of $\mathcal{O}(P^2)$ parameters we now have $\mathcal{O}(P^2 T)$. In such a semi-parametric model the potential complexity can scale with the amount of data we have available. Our aim is to harness this additional flexibility to identify potential changes within the graphical models which may shed insight onto dynamics of the data-generating system.

Local kernel/window estimation

Zhou et. al. [20] consider the dynamic GGM model in a continuous setting such that the underlying graphs are assumed to vary smoothly as a function of time. To provide a local estimate of the covariance they suggest the estimator $\hat{\boldsymbol{S}}(t) = \sum_s w_{st} \boldsymbol{y}_s \boldsymbol{y}_s^\top / \sum_s w_{st}$, where $w_{st} = K(|s-t|/h_T)$ are weights derived from a symmetric non-negative kernel (typically one may use a box-car/Gaussian function) with bandwidth h_T. The idea is that by replacing $\hat{\boldsymbol{S}}$ with $\hat{\boldsymbol{S}}(t)$ in the graphical lasso problem (Eq. 2) it is possible to obtain a temporally localized estimate of the graph $\hat{\boldsymbol{\Theta}}(t)_{GL}$. Given some smoothness conditions on the true covariance matrices one can demonstrate [20] that the estimator is consistent (estimator risk converges in probability $R(\hat{\boldsymbol{\Sigma}}(t)) - R(\boldsymbol{\Sigma}^*(t)) \xrightarrow{P} 0$) even in the dynamic (non-identically distributed) case.

Piecewise constant GGM

The seminal work by Zhou et al. [20] focused in the setting where graphs continuously and smoothly evolve over time. However, there are many situations where we might expect the smoothness assumptions to be broken. Our research [7,8,9] focuses on how we can incorporate different smoothness assumptions when estimating dynamic GGM. In particular we wish to study piecewise constant GGM where the generative distribution is strictly stationary within regions separated by a set of changepoints $\mathcal{T} = \{\tau_1, \dots, \tau_K\}$, $\tau_i \in \{1, \dots, T\}$, such that:

$$P(Y^t) = P(Y^{t+i}) \; \forall \, t, (t+i) \in \{\tau_k, \dots \tau_{k+1}\} \text{ for } k = 0, \dots, K-1 \, .$$

If we keep the Gaussian assumption of Eq. (3), then estimation relates to finding a set of $K-1$ GGM describing the distribution between changepoints. Such a definition extends the usual definition of a changepoint [13] to multivariate distributions, it is expected that the number of changepoints should be small relative to the total period of measurement, i.e. $K \ll T$ and that such points may lead to insight about changes within observed systems.

5 Structure learning with dynamic GGM

Our approach to searching for changepoints falls naturally into the M-estimation framework of Eq. (1). As has already been discussed, appropriate complexity penalties $R(\cdot)$ may act to induce sparsity in a given set of parameters. We propose two sparsity aware estimators that use such properties not only to estimate the graphical model, but also jointly extract a sparse set of changepoints.

Independent Fusing

Our first approach (see [7,9], also related to [14,3,18]) constructs a model fit function $L(\boldsymbol{\Theta}, \boldsymbol{Y}) = \sum_{t=1}^{T} \big(-\operatorname{logdet}(\boldsymbol{\Theta}^t) + \operatorname{tr}(\hat{\boldsymbol{S}}^t \boldsymbol{\Theta}^t) \big)$, where $\hat{\boldsymbol{S}}^t = \boldsymbol{y}^t (\boldsymbol{y}^t)^\top / 2$ is an estimate of the covariance for a specific time t. Clearly, there is not enough information within $\hat{\boldsymbol{S}}^t$ to recover a graph, as we are effectively trying to estimate with only one data point. To solve this problem we introduce an explicit prior on the smoothness of the graph via a complexity function

$$R_{IFGL}(\boldsymbol{\Theta}) = \lambda_1 \sum_{t=1}^{T} \|\boldsymbol{\Theta}^t\|_1 + \lambda_2 \sum_{t=2}^{T} \|\boldsymbol{\Theta}^t - \boldsymbol{\Theta}^{t-1}\|_1 \ , \tag{4}$$

where λ_1, λ_2 control the level of sparsity and number of changepoints in the model. Unlike in the work of Zhou et al. our prior encodes an assumption that the model has a piecewise constant parameterisation (this is similar to the fused lasso, see [16,10]). We refer to the problem $\{\hat{\boldsymbol{\Theta}}\}_{t=1}^{T} = \arg\min_{\boldsymbol{\Theta} \succeq 0} \big[L(\cdot) + R_{IFGL}(\cdot) \big]$ as defined above, as the *independently fused graphical lasso (IFGL)*, it estimates changepoints at an individual edge level such that changepoints do not necessarily coincide between edges.

Group Fusing

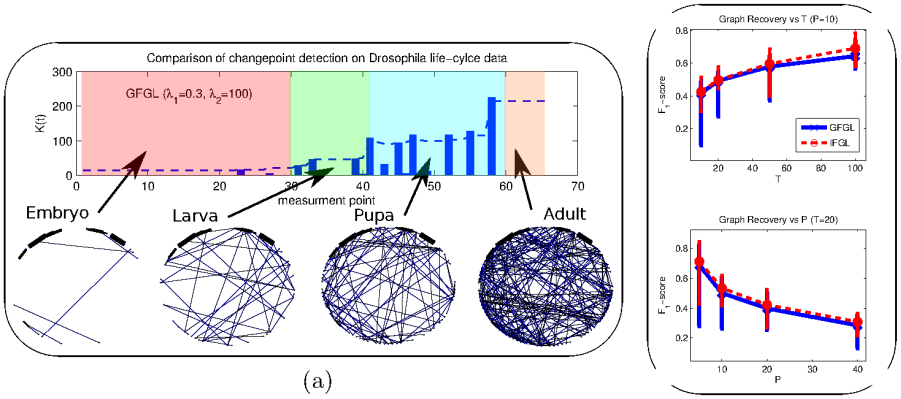
Sometimes we have a-priori knowledge that particular variables may change in a grouped manner, that is changepoints across the edges which connect variables may coincide. Examples, might include genes associated with a specific biological function (see the example in Fig. 1), or stocks within a given asset class. In order to encode such prior structure for changepoints one can adapt the smoothing prior to act over a group of edges, for such cases we suggest the *group-fused graphical lasso (GFGL)* penalty[9]:

$$R_{GFGL}(\boldsymbol{\Theta}) = \lambda_1 \sum_{t=1}^{T} \|\boldsymbol{\Theta}^t\|_1 + \lambda_2 \sum_{t=2}^{T} \|\boldsymbol{\Theta}^t - \boldsymbol{\Theta}^{t-1}\|_2 \ . \tag{5}$$

Optimisation

Both IFGL and GFGL form non-smooth convex optimisation problems which can be tackled within a variety of optimisation schemes. We have developed

an alternating direction method of multipliers (ADMM) algorithm that efficiently solves both the above problems by taking advantage of subtle separability properties of the estimators. Typically, one can solve for several changepoints $K = 1, \ldots, \sim 10$ on problems of size $T \approx 100 - 1000$, $P \approx 10 - 100$ in a few minutes. Due to the convex formulation scaling is linear in time $\mathcal{O}(TP^3K^2)$ for GFGL, which is a considerable advantage when compared to the quadratic time complexity of dynamic programming approaches [1].



(a) (b)

Fig. 1: a) Example of estimated graphical models (using GFGL) describing gene dependency through the life-cycle of *Drosophila melanogaster* (the common fruit fly). The generative distribution is assumed to be stationary between the blue bars which indicate changepoints. The dashed line indicates the number of active edges in the recovered graph. b) Results from an empirical study [9] considering how recovery of the graph changes with problem size.

6 Conclusion

To date, we have examined some properties of the IFGL and GFGL estimators in an empirical setting (see Fig. 1). Through the use of a wavelet framework our work [8] has also considered how one could allow for trends and changes in the mean parameter for dynamic GGM. Empirical results suggest some desirable properties for the proposed estimators (graph recovery improves when one increases the size and amount of data available within the stationary segments, see Fig. (1b), however, we have yet to examine the theoretical consistency properties. Theoretical analysis is complicated by the fact we regularise in multiple directions (the graph and over time), it is possible some insight in this direction can be gained from results in the regression setting [19].

References

1. D. Angelosante and G. B. Giannakis. Sparse graphical modeling of piecewise-stationary time series. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.
2. S. Boyd and L. Vandenberghe. *Convex Optimization*. 2004.
3. P. Danaher, P. Wang, and D. M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2013.
4. M. Drton and M. D. Perlman. Model selection for Gaussian concentration graphs. *Biometrika*, 2004.
5. R. Foygel and M. Drton. Extended Bayesian information criteria for gaussian graphical models. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*. 2010.
6. J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics (Oxford, England)*, 2008.
7. A. J. Gibberd and J. D. B. Nelson. High dimensional changepoint detection with a dynamic graphical lasso. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.
8. A. J. Gibberd and J. D. B. Nelson. Estimating multi-resolution dependency graphs within a locally stationary wavelet framework. *In review*, 2015.
9. A. J. Gibberd and J. D. B. Nelson. Regularized Estimation of Piecewise Constant Gaussian Graphical Models: The Group-Fused Graphical Lasso. *In review*, 2015.
10. Z. Harchaoui and C. Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 2010.
11. J. Lafferty, H. Liu, and L. Wasserman. Sparse nonparametric graphical models. *Statistical Science*, 2012.
12. S. L. Lauritzen. *Graphical models*. Oxford, 1996.
13. M. A. Little and N. S. Jones. Generalized methods and solvers for noise removal from piecewise constant signals. II. New methods. *Proceedings. Mathematical, physical, and engineering sciences / the Royal Society*, 2011.
14. R. P. Monti, P. Hellyer, D. Sharp, R. Leech, C. Anagnostopoulos, and G. Montana. Estimating time-varying brain connectivity networks from functional MRI time series. *NeuroImage*, 2014.
15. S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 2012.
16. R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2005.
17. H. Wang. Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis*, 2012.
18. S. Yang, Z. Pan, X. Shen, P. Wonka, and J. Ye. Fused multiple graphical lasso. *Arxiv*, 2012.
19. B. Zhang, J. Geng, and L. Lai. Multiple change-points estimation in linear regression models via sparse group lasso. *IEEE Trans. Signal Processing*, 2015.
20. S. Zhou, J. Lafferty, and L. Wasserman. Time varying undirected graphs. *Machine Learning*, 2010.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Sequential Pattern Mining on Multimedia Data

Corentin Hardy, Laurent Amsaleg, Guillaume Gravier, Simon Malinowski, and Reniniou
IRISA/Inria Rennes, France

Abstract. Analyzing multimedia data is a challenging problem due to the quantity and complexity of such data. Mining for frequently recurring patterns is a task often ran to help discovering the underlying structure hidden in the data. In this article, we propose audio data symbolization and sequential pattern mining methods to extract patterns from audio streams. Experiments show that this task is hard and that the symbolization is a critical step for extracting relevant audio patterns.

1 Introduction

The amount of multimedia data grows from day to day with ever increasing acquisition and storage capabilities. In turn, analyzing such complex data to extract knowledge is a challenging problem. For instance, analysts are looking for methods that could help to discover the underlying structure of multimedia documents such as video or audio streams. Unsupervised extraction of recurrent patterns and finding their occurrences in the data could provide such a segmentation and could achieve a first step towards the automatic understanding of multimedia data. In an audio stream, a word, a jingle, or an advertisement could typically represent a pattern. However, the variability of audio motifs makes pattern mining difficult, especially audio motifs related to words, since the variability due to different speakers and channels is high.

Overall, the extraction of repeated motifs in time series is a very active domain. Two kinds of approaches have been proposed: the first one consists in working directly with the time series and in finding close sub-sequences based on a distance measure such as the Euclidean or the Dynamic Time Warping (DTW) [1] distances. The second one consists in transforming the time series into sequences of symbols to then use sequential motif discovery algorithms [2]. Very few works have investigated the second approach; this preliminary work thus explores how to use sequential pattern mining algorithms on audio data.

This paper is organized as follows. In Section 2, we review the related work about motif discovery in audio data. In Section 3, we explain our proposed approach. Section 4 presents preliminary results and section 5 concludes and discusses future issues for this work.

2 Related work

Motif discovery relies either on raw time series processing or on mining a symbolic version [3,4,5]. In the first kind of approaches, algorithms are mostly built on

the DTW distance which can deal with temporal distortions that often occurs in audio signals [6]. Muscariello et al. [7] have proposed an extended version of the DTW for finding the best occurrence of a seed in a longer subsequence. This kind of approaches is efficient in terms of accuracy as the signal is completely exploited but the computational cost of the DTW distance prevents its use on very large databases.

Other approaches working with a symbolized version of the audio signal mostly use algorithms from bioinformatics to extract motifs. In [8], the MEME algorithm [9] is used to estimate a statistical model for each discovered motif. In [10], the SNAP algorithm [11] is used to search by query near-duplicate video sequences.

Some algorithms coming from bioinformatics are very efficient, but have been optimized to work with alphabets of very small size (from 4 to 20). In this paper, we consider the use of sequential pattern mining algorithms for discovering motifs in audio data.

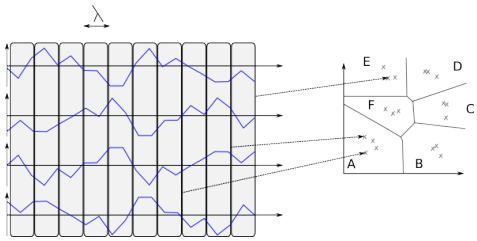
3 Pattern mining on audio data

In this section, we explain how we used sequential pattern mining algorithms to discover repeating patterns in audio data. As pattern mining algorithms deal with symbolic sequences, we present first how to transform time series related to audio data into symbolic sequences. Then we show how to use sequential pattern algorithms on symbolic sequences.

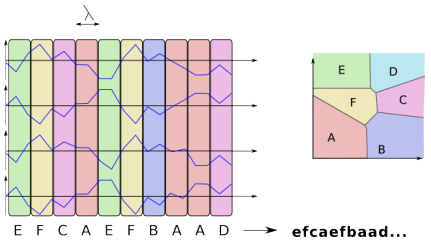
MFCC (Mel-frequency cepstral coefficients) is a popular method for representing audio signals. First, MFCC coefficients are extracted from the raw audio signal (with a sliding window) yielding a 13-dimensional time series. Then, this multivariate time series is transformed into a sequence of symbols. Many methods have been proposed for transforming time series into a sequence of symbols. Here, we have chosen to use a method proposed by Wang et al. [12]. We have also tried the very popular SAX approach [2]. SAX symbols contain very few information about the original signal (only the average value on a window). This symbolisation technique is less adapted to our problem and produced worse results.

To this end, each dimension of the 13-dimensional time series is divided into consecutive non-overlapping windows of length λ. The 13 sub-series related to the same window are then concatenated (respecting the order of the MFCC data). The resulting vectors of size $13 \times \lambda$ are then clustered by a k-means algorithm for building a codebook, each word in the codebook corresponding to a cluster. Finally, the original multivariate time series is coded into a sequence of symbols by assigning to each window the symbol in the codebook corresponding to the closest cluster centroid. This symbolization process is sketched in Figures 1a and 1b.

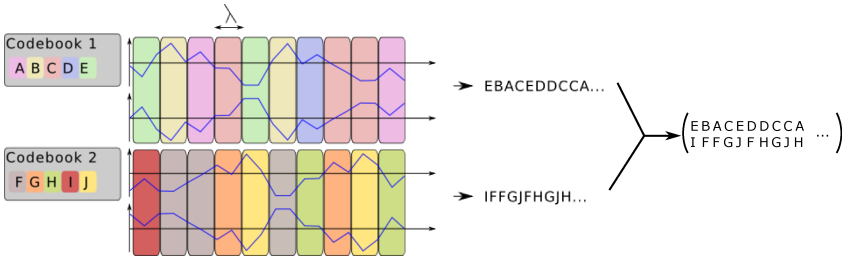
The representation above could be too imprecise as it mixes coefficients of very different order. To cope with this problem we propose to divide the 13 dimensions into 2 or more sub-bands of consecutive dimensions that represent



(a) K-means clustering is performed on the set of windows to build a codebook (of size 6, here).



(b) Every window is labeled by the symbol associated with the closest cluster centroid.



(c) Conversion of a 2 sub-band times series into a sequence of itemsets using 2 codebook of size 5.

Fig. 1: Time series symbolization into a sequence of items (figures 1a and 1b) and a sequence of itemsets (figure 1c).

more closely related dimensions. The same transformation described above operates on sub-bands and yields one codebook per sub-band. There are thus as many symbolic sequences as there are sub-bands. Finally, the sub-band symbols related to the same windows are grouped into itemsets in the Figure 1c.

Once the raw signal is transformed into a symbolic sequence of items or itemsets, classical sequential motif discovery algorithms can be applied. Two kinds of sequential pattern discovery algorithms have been proposed: algorithms that process sequences of items and algorithms that process sequences of itemsets (an itemset is a set of items that occur in a short time period). We have chosen to evaluate one algorithm of each kind in this paper: MaxMotif [13] and CMP-Miner [14] that process respectively sequences of items and sequences of itemsets.

Note that, in the classical setting of sequential pattern mining, a pattern occurrence may skip symbols in the sequence. For instance, ***acccb*** is an occurrence of pattern *ab* in sequence *dacccbe*. Generally, algorithms provide means to put constraints on extracted motifs, such as minimum and maximal motif length and the allowed gaps; gaps are symbols that can be skipped when looking for a pattern occurrence. In our application, it is crucial to allow gaps in motifs since temporal distortions often occurs in audio signals.

MaxMotif enumerates all frequent (with respect to a given minimal support) closed patterns in a database of item sequences. MaxMotif allows gaps in the

temporal domain (represented by the *wildcard* symbol $-$). For instance, pattern $(f - a)$ occurs in sequence (e**fca**e**fba**ab) at positions 2 and 6.

CMP-Miner extracts all frequent closed patterns in a database of itemset sequences. It uses the PrefixSpan projection principle [15] and the BIDE bidirectional checking [16]. CMP-Miner allows gaps both in the temporal domain and inside an itemset. For instance, pattern $\begin{pmatrix} b & - & c & - \\ f & - & g & j \end{pmatrix}$ occurs in sequence $\begin{pmatrix} e & b & a & c & e & b & d & c & c & a \\ i & f & f & g & j & f & h & g & j & h \end{pmatrix}$ at positions 2 and 6.

The parameters of the two methods are described in Table 1.

Table 1: List of parameters

Methods	**Symbolization**	**Parameters for mining**
MaxMotif	α, size of codebook.	*minSupport*, minimal support.
	λ, length of windows.	*maxGap*, maximal gap between 2 consecutive items in a pattern.
		maxLength, maximal pattern length.
		minLength, minimal pattern length.
CMP-Miner	α, size of codebook.	*minSupport*, minimal support.
	λ, length of windows.	*maxGap*, maximal gap between 2 consecutive itemsets in a pattern.
	β, number of bands.	*minItem*, minimal number of items in itemsets.
		maxLength, maximal pattern length.
		minLength, minimal pattern length.

4 Experiments

We present in this section some results from two experiments, one on a synthetic dataset and the other on a real dataset.

4.1 Experiment on a synthetic dataset

In this first experiment, we have created a dataset composed of 30 audio signals corresponding to 10 utterances of the 3 words "affaires", "mondiale" and "cinquante" pronounced by several French speakers. Our goal is to evaluate the impact of the codebook size on the extracted motifs. The two algorithms presented above have been applied on this dataset with the following parameters: $\lambda = 5$, $minSupport = 4$, $maxGap = 1$, $minLength = 4$, $maxLength = 20$. For CMP-Miner we set $\beta = 3$ and $minItem = 2$. These parameter settings were chosen after extensive tests on possible value ranges.

First, sequential patterns are extracted. Then, we associate with each pattern the word in the utterances of which this pattern most often occurs. For each

extracted pattern, a precision/recall score is computed. Figure 2a and 2b depict the precision/recall score versus the codebook size for MaxMotif and CMP-Miner. As can be seen, MaxMotif obtains the best efficiency. This figure also shows that when the codebook size increases, the precision improves slightly but not the recall.

Figure 2c shows the pattern length distribution for different codebook sizes for MaxMotif. For small codebooks, many long patterns are extracted. However, they are not very accurate because, being general, they can occur in many different sequences. For big codebooks, many pattern candidates can be found, reflecting sequence variability. However, many candidates have a low support, often under the minimal threshold, and, so, less patterns are extracted.

The symbolization step is crucial. Figure 2d shows five symbolic representations of the word "cinquante" for a codebook of size 15. These strings highlight the two kinds of variability (spectral and temporal) that makes the task hard for mining algorithms in this example. The same experiment was performed using the SAX symbolization method [2] on each dimension of the multidimensional times series. This representation revealed to be less accurate. Indeed, the results obtained by CMP-Miner using the SAX representation were worse. There is no space to detail these results here.

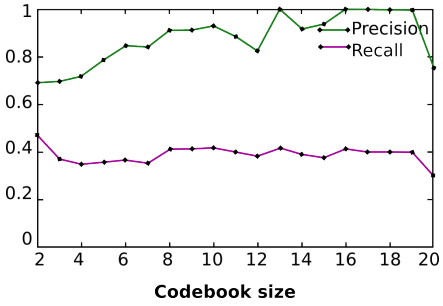
4.2 Experiment on a larger database

Now, we consider a dataset containing 7 hours of audio content. The dataset is divided into 21 audio tracks coming from various radio stations. This experience is closer to a real setting.

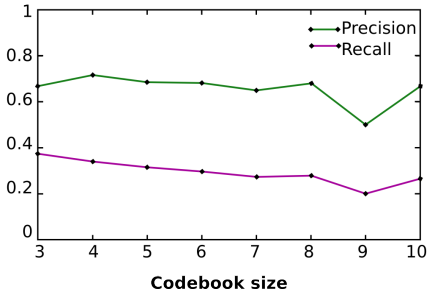
Only MaxMotif has been tested on this dataset. The parameters were: $\lambda = 4$, $\alpha = 80$, $minSupport = 40$, $maxGap = 1$, $minLength = 5$, $maxLength = 20$. The codebook size is greater than in the previous experiment to deal with more different sounds. Pattern extraction is very fast: less than 4 minutes for more than one million of patterns. Some of them are interesting and correspond, for instance, to crowd noises, jingle and music patterns or short silence. However, similarly to the experiment on the synthetic dataset, only very few patterns corresponding to repeated words could be extracted.

5 Conclusion

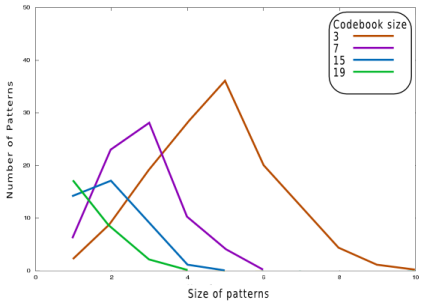
In this paper, we have presented a preliminary work investigating how to use sequential pattern mining algorithms for audio data. The aim of this work was to evaluate whether these algorithms could be relevant for this problem. The experiments pointed out the difficulty to mine audio signals, because of temporal and spectral distortion. Same words pronounced in different contexts and by different speakers can be very different and yield very different patterns. The results are promising but both symbolization and motif extraction should be improved. For instance, to account for spectral variability, considering distances between symbols should improve the overall performance of pattern extraction.



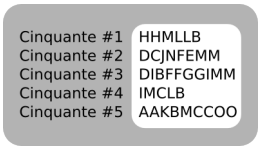
(a) Precision/Recall curves for MaxMotif.



(b) Precision/Recall curves for CMP-Miner with 3 sub-bands.



(c) Pattern size distribution for different size of codebook.



(d) Example of representation for a codebook of size 15.

Fig. 2: Results of experience on synthetic data.

We have also noticed that all the dimensions of the MFCC times series are not as important for the discovery. Selecting or weighting the dimensions of multidimensional time series could improve the performance too.

References

1. A. Mueen, "Enumeration of time series motifs of all lengths," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pp. 547–556, Dec 2013.
2. J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: A novel symbolic representation of time series," *Data Min. Knowl. Discov.*, vol. 15, pp. 107–144, Oct. 2007.
3. C. Herley, "ARGOS: Automatically extracting repeating objects from multimedia streams," *IEEE Trans. on Multimedia*, vol. 8, pp. 115–129, Feb. 2006.
4. P. Esling and C. Agon, "Time-series data mining," *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, p. 12, 2012.
5. C. H. Mooney and J. F. Roddick, "Sequential pattern mining – approaches and algorithms," *ACM Comput. Surv.*, vol. 45, Mar. 2013.

6. A. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transaction on Acoustic, Speech and Language Processing*, vol. 16, pp. 186–197, Jan. 2008.
7. A. Muscariello, G. Gravier, and F. Bimbot, "Audio keyword extraction by unsupervised word discovery," in *INTERSPEECH 2009: 10th Annual Conference of the International Speech Communication Association*, (Brighton, United Kingdom), Sept. 2009.
8. J. J. Burred, "Genetic motif discovery applied to audio analysis," in *International Conference on Acoustics, Speech and Signal Processing*, pp. 361–364, IEEE, 2012.
9. T. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine Learning*, vol. 21, no. 1-2, pp. 51–80, 1995.
10. L. S. d. Oliveira, Z. K. do Patrocinio, S. J. F. Guimarães, and G. Gravier, "Searching for near-duplicate video sequences from a scalable sequence aligner," in *International Symposium on Multimedia*, pp. 223–226, IEEE, 2013.
11. M. Zaharia, W. J. Bolosky, K. Curtis, A. Fox, D. Patterson, S. Shenker, I. Stoica, R. M. Karp, and T. Sittler, "Faster and more accurate sequence alignment with snap," *arXiv preprint arXiv:1111.5572*, 2011.
12. Q. Wang, V. Megalooikonomou, and C. Faloutsos, "Time series analysis with multiple resolutions," *Information Systems*, vol. 35, no. 1, pp. 56–74, 2010.
13. H. Arimura and T. Uno, "An efficient polynomial space and polynomial delay algorithm for enumeration of maximal motifs in a sequence," *Journal of Combinatorial Optimization*, vol. 13, 2007.
14. A. J. Lee, H.-W. Wu, T.-Y. Lee, Y.-H. Liu, and K.-T. Chen, "Mining closed patterns in multi-sequence time-series databases," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 1071 – 1090, 2009.
15. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *International Conference on Data Engineering*, p. 0215, 2001.
16. J. Wang and J. Han, "Bide: efficient mining of frequent closed sequences," in *Data Engineering, Proceedings. 20th International Conference on Data Engineering*, pp. 79–90, March 2004.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Time Series Classification in Dissimilarity Spaces

Brijnesh J. Jain and Stephan Spiegel
Berlin Institute of Technology, Germany
jain@dai-labor.de, spiegel@dai-labor.de

Abstract. Time series classification in the dissimilarity space combines the advantages of the dynamic time warping and the rich mathematical structure of Euclidean spaces. We applied dimension reduction using PCA followed by support vector learning on dissimilarity representations to 43 UCR datasets. Results indicate that time series classification in dissimilarity space has potential to complement the state-of-the-art.

1 Introduction

Time series classification finds many applications in diverse domains such as speech recognition, medical signal analysis, and recognition of gestures [2–4]. Surprisingly, the simple nearest-neighbor method together with the dynamic time warping (DTW) distance still belongs to the state-of-the-art and is reported to be *exceptionally difficult to beat* [1, 5, 10]. This finding is in stark contrast to classification in Euclidean spaces, where nearest neighbor methods often merely serve as baseline. One reason for this situation is that nearest neighbor methods in Euclidean spaces compete against a plethora of powerful statistical learning methods. The majority of these statistical learning methods are based on the concept of derivative not available for warping-invariant functions on time series.

The dissimilarity space approach proposed by [7] offers to combine the advantages of the DTW distance with the rich mathematical structure of Euclidean spaces. The basic idea is to first select a set of k reference time series, called prototypes. Then the dissimilarity representation of a time series consists of k features, each of which represents its DTW distance from one of the k prototypes. Since dissimilarity representations are vectors from $\mathbb{R}^k$, we can resort to the whole arsenal of mathematical tools for statistical data analysis. The dissimilarity space approach has been systematically applied to the graph domain using graph matching [6, 9]. A similar systematic study of the dissimilarity space approach for time series endowed with the DTW distance is still missing.

This paper is a first step towards exploring the dissimilarity space approach for time series under DTW. We hypothesize that combining the advantages of both, the DTW distance and statistical pattern recognition methods, can result in powerful classifiers that may complement the state-of-the-art. The proposed approach applies principal component analysis (PCA) for dimension reduction of the dissimilarity representations followed by training a support vector machine (SVM). Experimental results provide support for our hypothesis.

2 Dissimilarity Representations of Time Series

2.1 Dynamic Time Warping Distance

A time series of length n is an ordered sequence $\boldsymbol{x} = (x_1, \ldots, x_n)$ with features $x_i \in \mathbb{R}$ sampled at discrete points of time $i \in [n] = \{1, \ldots, n\}$. To define the DTW distance between time series $\boldsymbol{x}$ and $\boldsymbol{y}$ of length n and m, resp., we construct a grid $\mathcal{G} = [n] \times [m]$. A warping path in grid $\mathcal{G}$ is a sequence $\phi = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_p)$ consisting of points $\boldsymbol{t}_k = (i_k, j_k) \in \mathcal{G}$ such that

1. $\boldsymbol{t}_1 = (1,1)$ and $\boldsymbol{t}_p = (n,m)$ (boundary conditions)
2. $\boldsymbol{t}_{k+1} - \boldsymbol{t}_k \in \{(1,0),(0,1),(1,1)\}$ (warping conditions)

for all $1 \leq k < p$. The cost of warping $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_m)$ along ϕ is defined by

$$d_\phi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{(i,j)\in\phi} (x_i - y_j)^2,$$

where $(x_i - y_j)^2$ is the local transformation cost of assigning features x_i to y_j. Then the distance function

$$d(\boldsymbol{x}, \boldsymbol{y}) = \min_{\phi} d_\phi(\boldsymbol{x}, \boldsymbol{y}),$$

is the dynamic time warping (DTW) distance between $\boldsymbol{x}$ and $\boldsymbol{y}$, where the minimum is taken over all warping paths in $\mathcal{G}$.

2.2 Dissimilarity Representations

Let $(\mathcal{T}, d)$ be a time series space $\mathcal{T}$ endowed with the DTW distance d. Suppose that we are given a subset

$$\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k\} \subseteq \mathcal{T}$$

of k reference time series $\boldsymbol{p}_i \in \mathcal{T}$, called prototypes henceforth. The set $\mathcal{P}$ of prototypes gives rise to a function of the form

$$\phi : \mathcal{T} \to \mathbb{R}^k, \quad \boldsymbol{x} \mapsto (d(\boldsymbol{x}, \boldsymbol{p}_1), \ldots, d(\boldsymbol{x}, \boldsymbol{p}_k)),$$

where $\mathbb{R}^k$ is the *dissimilarity space* of $(\mathcal{T}, d)$ with respect to $\mathcal{P}$. The k-dimensional vector $\phi(\boldsymbol{x})$ is the *dissimilarity representation* of $\boldsymbol{x}$. The i-th feature of $\phi(\boldsymbol{x})$ represents the dissimilarity $d(\boldsymbol{x}, \boldsymbol{p}_i)$ between $\boldsymbol{x}$ and the i-th prototype $\boldsymbol{p}_i$.

2.3 Learning Classifiers in Dissimilarity Space

Suppose that

$$\mathcal{X} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\} \subseteq \mathcal{T} \times \mathcal{Y}.$$

is a training set consisting of n time series $\boldsymbol{x}_i$ with corresponding class labels $y_i \in \mathcal{Y}$. Learning in dissimilarity space proceeds in three steps: (1) select a

suitable set of prototypes $\mathcal{P}$ on the basis of the training set $\mathcal{D}$, (2) embed time series into the dissimilarity space by means of their dissimilarity representations, and (3) learn a classifier in the dissimilarity space according to the empirical risk minimization principle.

The performance of a classifier learned in dissimilarity spaces crucially depends on a proper dissimilarity representation of the time series. We distinguish between two common approaches:

1. *Prototype selection*: construct a set of prototypes $\mathcal{P}$ from the training set $\mathcal{X}$.
2. *Dimension reduction*: perform dimension reduction in the dissimilarity space.

There are numerous strategies for prototype selection. Naive examples include all elements of the training set $\mathcal{X}$ and sampling a random subset of $\mathcal{X}$. For more sophisticated selection methods, we refer to [8]. Dimension reduction of the dissimilarity representation includes methods such as, for example, principal component analysis (PCA).

3 Experiments

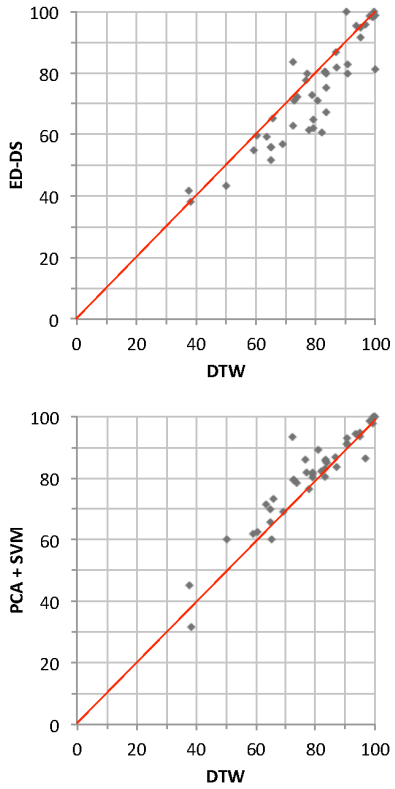
The goal of this experiment is to assess the performance of the following classifiers in dissimilarity space: (1) nearest neighbor using the Euclidean distance (ED-DS), (2) support vector machine (SVM), and (3) principal component analysis on dissimilarity representations followed by support vector machine (PCA+SVM).

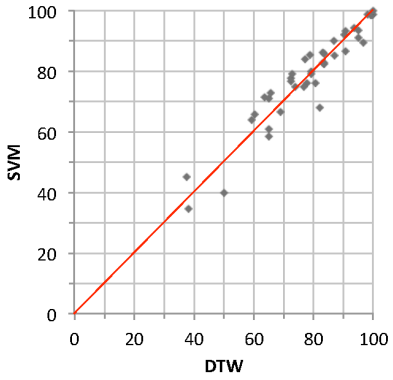
3.1 Experimental Protocol

We considered 43 datasets from the UCR time series datasets [4], each of which comes with a pre-defined training and test set. For each dataset we used the whole training set as prototype set. To embed the training and test examples into a dissimilarity space, we computed their DTW distances to the prototypes.

We trained all SVMs with RBF-kernel using the embedded training examples. We selected the parameters γ and C of the RBF-kernel over a two-dimensional grid with points $(\gamma_i, C_j) = (2^i, 2^j)$, where i, j are 30 equidistant values from $[-10, 10]$. For each parameter configuration (γ_i, C_j) we performed 10-fold cross-validation and selected the parameters (γ_*, C_*) with the lowest average classification error. Then we trained the SVM on the whole embedded training set using the selected parameters (γ_*, C_*). Finally, we applied the learned model to the embedded test examples for estimating the generalization performance.

For PCA+SVM we performed dimension reduction using PCA prior training of the SVM. We considered the q first dimensions with highest variance, where $q \in \{1, 1+a, 1+2a, \ldots, 1+19a\}$ with a being the closest integer of $k/20$ and k is the dimension of the dissimilarity space. For each q, we performed hyper-parameter selection for the SVM as described above. We selected the parameter configuration (q_*, γ_*, C_*) that gave the lowest classification error. Then we applied PCA on the whole embedded training set, retained the first q_* dimensions and trained the SVM on the embedded training set after dimension reduction.





method	better	equal	worse
ED-DS	7	1	34
SVM	20	2	20
PCA+SVM	28	5	9

Fig. 1. Scatter plots of accuracy of DTW against dissimilarity space methods.

Finally we reduced the dimension of the embedded test examples and applied the learned model.

3.2 Results

Figure 1 shows the scatter plots of predictive accuracy of the nearest neighbor using DTW against all three dissimilarity space methods and Table 1 shows the error rates of all classifiers for each dataset.

The first observation to be made is that the dissimilarity space endowed with the Euclidean space is less discriminative than the time series space endowed with the DTW distance. As shown by Figure 1, nearest neighbor (NN) with DTW performed better than the ED-DS classifier in 34 out of 42 cases. Since the DTW distance is non-Euclidean, dissimilarity spaces form a distorted representation of the time series space in such a way that neighborhood relations are not preserved. In most cases, these distortions impact classification results negatively, often by a large margin. In the few cases where the distortions improve classification

Data	DTW	ED-DS	SVM	PCA+SVM
50words	31.0	42.9	33.4	31.0 (+0.0)
Adiac	39.6	40.2	34.0	37.6 (+5.1)
Beef	50.0	56.7	60.0	40.0 (+20.0)
CBF	0.3	0.2	1.4	0.2 (+32.7)
ChlorineConcentration	35.2	48.3	28.9	30.2 (+14.3)
CinC ECG Torso	34.9	44.0	41.4	39.8 (-14.0)
Coffee	17.9	39.3	32.1	17.9 (+0.0)
Cricket X	22.3	38.5	24.1	23.6 (-5.8)
Cricket Y	20.8	37.9	20.0	19.7 (+5.1)
Cricket Z	20.8	34.9	20.8	18.2 (+12.5)
DiatomSizeReduction	3.3	4.2	10.8	13.7 (-315.9)
ECG	23.0	20.0	16.0	18.0 (+21.7)
ECGFiveDays	23.2	22.4	24.9	14.1 (+39.4)
Face (all)	19.2	28.9	23.8	10.9 (+43.0)
Face (four)	17.1	19.3	13.6	17.0 (+0.0)
FacesUCR	9.5	17.1	13.4	9.0 (+5.6)
Fish	16.7	32.6	17.7	14.3 (+14.5)
Gun-Point	9.3	20.0	6.7	8.7 (+7.1)
Haptics	62.3	58.4	54.9	54.9 (+11.9)
InlineSkate	61.6	61.8	65.5	68.4 (-11.0)
ItalyPowerDemand	5.0	8.4	6.5	6.3 (-26.3)
Lighting 2	13.1	18.0	14.8	16.4 (-25.1)
Lighting 7	27.4	37.0	23.3	20.5 (+25.0)
Mallat	6.6	4.6	5.6	5.5 (+17.3)
Medical Images	26.3	27.9	24.9	21.7 (+17.4)
MoteStrain	16.5	24.8	17.2	14.1 (+14.8)
Olive Oil	13.3	13.3	10.0	13.3 (+0.0)
OSU Leaf	40.9	45.0	36.0	38.0 (+7.1)
SonyAIBORobotSurface	27.5	16.3	22.3	6.7 (+75.8)
SonyAIBORobotSurface II	16.9	19.4	17.4	19.3 (-14.2)
Swedish Leaf	21.0	27.2	14.4	18.7 (+10.9)
Symbols	5.0	5.3	8.7	5.3 (-6.5)
Synthetic Control	0.7	1.7	1.3	2.0 (-185.7)
Trace	0.0	1.0	1.0	0.0 (+0.0)
TwoLeadECG	9.6	18.7	7.7	7.1 (+25.9)
TwoPatterns	0.0	18.7	0.0	0.0 (+0.0
uWaveGestureLibrary X	27.3	28.9	20.8	20.6 (+24.6)
uWaveGestureLibrary Y	36.6	40.5	28.6	28.5 (+22.2)
uWaveGestureLibrary Z	34.2	34.6	27.0	26.9 (+21.4)
Wafer	2.0	1.5	1.1	1.5 (+26.2)
WordsSynonyms	35.1	43.9	39.0	34.3 (+2.2)
yoga	16.4	20.0	14.0	14.8 (+9.6)

Table 1. Error rates in percentages. Numbers in parentheses show percentage improvement of PCA+SVM with respect to DTW.

results, the improvements are only small and could also be occurred by chance due to the random sampling of the training and test set.

The second observation to be made is that the SVM using all prototypes complements NN+DTW. Better and worse predictive performance of both classifiers is balanced. This shows that powerful learning algorithms can partially compensate for poor representations.

The third observation to be made is that SVM+PCA outperformed all other classifiers. Furthermore, SVM+PCA is better than NN+DTW in 28 and worse in 9 out of 42 cases. By reducing the dimension using PCA, we obtain better dissimilarity representations for classification. Table 1 highlights relative improvements and declines of PCA+SVM compared to NN+DTW with $\pm 10\%$ or more in blue and red color, respectively. We observe a relative change of at least $\pm 10\%$ in 27 out of 43 cases. This finding supports our hypothesis that learning on dissimilarity representations complements NN+DTW.

4 Conclusion

This paper is a first step to explore dissimilarity space learning for time series classification under DTW. Results combining PCA with SVM on dissimilarity representations are promising and complement nearest neighbor methods using DTW in time series spaces. Future work aims at exploring further elastic distances, prototype selection, dimension reduction, and learning methods.

References

1. G.E. Batista, X. Wang, and E.J. Keogh. A Complexity-Invariant Distance Measure for Time Series. *SIAM International Conference on Data Mining*, 11:699–710, 2011.
2. T. Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.
3. P. Geurts. Pattern extraction for time series classification. *Principles of Data Mining and Knowledge Discovery*, pp. 115–127, 2001.
4. E. Keogh, Q. Zhu, B. Hu, Y. Hao., X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR Time Series Classification/Clustering Homepage: `www.cs.ucr.edu/~eamonn/time_series_data/`, 2011.
5. J. Lines and A. Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 2014.
6. L. Livi, A. Rizzi, and A. Sadeghian. Optimized dissimilarity space embedding for labeled graphs. *Information Sciences*, 266:47–64, 2014.
7. E. Pekalska and R.P.W. Duin *The Dissimilarity Representation for Pattern Recognition.* World Scientific Publishing Co., Inc., 2005.
8. E. Pekalska, R.P.W. Duin, and P. Paclik. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2): 189–208, 2006.
9. K. Riesen and H. Bunke. Graph classification based on vector space embedding. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(6):1053–1081, 2009.
10. X. Xi, E. Keogh, C. Shelton, L. Wei, and C.A. Ratanamahatana. Fast time series classification using numerosity reduction. *International Conference on Machine Learning*, pp. 1033–1040, 2006.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Temporal Density Extrapolation

Georg Krempl
Knowledge Management & Discovery
Otto-von-Guericke University Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany
georg.krempl@iti.cs.uni-magdeburg.de
https://kmd.cs.ovgu.de/res/driftmining

Abstract. Mining evolving datastreams raises the question how to extrapolate trends in the evolution of densities over time. While approaches for change diagnosis work well for interpolating spatio-temporal densities, they are not designed for extrapolation tasks. This work studies the temporal density extrapolation problem and sketches two approaches that addresses it. Both use a set of pseudo-points in combination with spatio-temporal kernel density estimation. The first, weight-extrapolating approach, uses regression on the weights of stationary-located pseudo-points. The second, location-extrapolating approach, extrapolates the trajectory of uniformly-weighted pseudo-points within the feature space.

Keywords: kernel density estimation, density extrapolation, density forecasting, spatio-temporal density, evolving datastreams, nonstationary environments, concept drift, drift mining

1 Introduction

Density estimation methods, like kernel density estimation [14, 13], allow to learn a model from instances observed at different positions in feature space, and to use this model to estimate the density at any position within this feature space. While the original work in [14, 13] is limited to spatial densities of a stationary distribution, the approach was extended for spatio-temporal density estimation of non-stationary distributions in [1, 2]. This so-called velocity density estimation allows to estimate and visualise trends in densities. However, these existing approaches are not directly applicable for predicting the density at *future time points*, for example to extrapolate trends in the evolution of densities and for building classifiers that work with delayed label information [10, 5]. Such *temporal density extrapolation* should predict the densities at spatio-temporal coordinates in the future, given a sample of (historic) instances observed at different positions in feature space and at different times in the past.

We propose and study two approaches to address this problem. Both use extrapolation of pseudo-points in combination with spatio-temporal kernel density estimation. The first approach extrapolates the weights of stationary located pseudo-points, while the second extrapolates the path of moving pseudo-points of fixed weight. Subsequently, these weight- or position-extrapolated pseudo-points are used in a spatio-temporal kernel density estimation.

In the following Section 2, we review the related work, before sketching the two approaches in Section 3 and concluding in Section 4.

2 Related Work

The task of estimating the probability density based on a sample of independently and identically distributed (iid) observations has been intensively studied. *Density estimation* methods, like kernel density estimation [14, 13], allow to learn a model from instances observed at different positions in feature space, and to use this model to estimate the density at any position within this feature space. The difficulties of the early kernel and near-neighbour density estimation techniques when extended to multivariate settings was addressed by approaches like *projection pursuit* density estimation, proposed in [4]. All these density estimation approaches, as well as related curve regression approaches, require an iid sample from a stationary distribution [11].

In the case of a nonstationary distribution, one might be interested in estimating the density at different points in time and space. In [1, 2], this problem of *spatio-temporal density estimation* is addressed by combining spatial kernel density estimation with a temporal weighting of instances. A framework for so-called *change diagnosis* in evolving datastreams is proposed, which estimates the rate of change at each region by using a user-specified temporal window to calculate forward and reverse time slice density estimates. This velocity density estimation technique is applicable for spatio-temporal density *interpolation*, for monitoring and visualising the change of densities in a (past) time window. However, it is not designed for extrapolating the density to (future) time points outside the window of observed historical data.

Related to change diagnosis is *change mining* [3], which aims to understand the changes in data distributions themselves. Within this paradigm, the idea of so-called *drift-mining* approaches [8, 9, 5] is to model the evolution of distributions in order to extrapolate them to future time points, thereby addressing problems of verification latency or label delay. The algorithm APT proposed in [8] uses matching between labelled old and unlabelled new instances to infer the labels of the later, thus indirectly estimating the class-conditional distributions of the new instances. Likewise, an expectation-maximisation approach is used in [9] to track components of a Gaussian mixture model over time. In [5], a mixture model is learned on old labelled data and compared to density estimates on current unlabelled data, thereby inferring changes such as of the class prior. However, these approaches are again not designed for directly extrapolating densities.

Density forecasting approaches [15, 16, 7], on the other hand, focus on the prediction of a single variable's (mostly unimodal) density at a particular future timepoint, based on an observed time series of this variable's values. In the simplest case, as discussed for example in [16], this is done by providing a confidence interval for a point estimate, obtained by assuming a particular distribution of this variable. More sophisticated approaches return (potentially multi-modal)

density estimates by combining several predictions, which are obtained for example by different macroeconomic models, experts, or simulation outcomes, into a distribution estimation by kernel methods. Nevertheless, their multi-modal character originates from the different modes in the combined *unimodal* models. In addition, most works consider only a single variable. One exception is [7], where univariate forecasts of two explanatory variables are converted using conditional kernel density estimation into forecasts of the dependent variable.

In contrast to density forecasting above, we are concerned with temporal density extrapolation of a potentially *multi-modal density distribution*. Furthermore, instead of having a time series of single observations at any one time, our input data consists of multiple observations at any one time. This *temporal* density extrapolation is related to *spatial* density extrapolation [17, 6], which addresses the extrapolation of densities for feature values that have not been seen yet in historical instances. In [17], the authors suggest a Taylor series expansion about the point of interest to estimate the density, while in [6] a statistical test is provided to examine whether the data distribution is distinct from a uniform distribution at the extrapolation position. While modelling time as a feature is possible, there is an important difference in extrapolation between time and feature space: one expects the density to diminish towards unpopulated (and thus unseen) positions in feature space. However, there is no a priori reason to assume densities to decrease towards yet unseen moments in time. On the contrary, it is reasonable to assume that *at each point in time* (whether future, current, or past) the *density integrates to one* over the feature space.

3 Temporal Density Extrapolation

To address the problem of extrapolating the observed, potentially multi-modal density-distribution of instances to future time points, we propose an approach based on *pseudo-points*. These pseudo-points are used in the spatio-temporal kernel density estimation in lieu of the originally observed instances. The resulting kernel density estimation model can be interpreted as a mixture model, where each pseudo-point constitutes itself a component. The pseudo-points evolve over time, either by changing their weight (their component's mixing proportion), or by changing their position (their component's location). Therefore, the learning task is to fit a trend function to the evolution of each pseudo-point. We present each of the two variants in the next Subsections 3.1 and 3.2, before discussing their potential difficulties and limitations in Section 3.3.

3.1 Weight-Extrapolated, Stationary Pseudo-Points

Given a set of stationary pseudo-points, the first approach models their weights as functions of time. These functions are then fit on a window with historical data, such that the distribution therein is modelled with maximum likelihood.

The approach is illustrated for a one-dimensional feature space in Figure 1. At the first time point in the past ($time = 0$), a density estimate is calculated

using historical data collected at that time (solid blue line). Then, a set of pseudo-points (here $1, 2, \cdots 4$) is generated, either by placing them equidistant on a grid or by drawing them at random. Next, the weights $(w_1, w_2, \cdots w_4)$ of all pseudo-points are calculated such that the divergence is minimised between the kernel density estimate over the weighted pseudo-points and the kernel-density estimate over the original data instances at that time point. The pseudo-point's weights are estimated in the same way for subsequent time points (e.g. $t = 1$), as soon as instances become available for them. This results for each pseudo-point in a time series of weight values, for which a polynomial trend function (red curves) is learned by regression. Finally, for a future time point (e.g. $time = 2$), the trend functions' values are predicted ($w_1, w_2, \cdots w_4$ in red at $time = 2$). Using these weighted pseudo-points in a kernel density estimate at $time = 2$, one obtains the extrapolated density (red dotted line), which is later evaluated against the observed density (solid blue-gray line).

3.2 Position-Extrapolated, Uniformly-Weighted Pseudo-Points

The second approach to address this problem is to use uniformly-weighted, but flexibly located pseudo-points. Thus, the pseudo-point's weights are uniform and constant, but their positions are functions of time, fitted such that the divergence on the available historical data is minimised.

In analogy to the previous figure, this approach is illustrated for a one-dimensional feature space in Figure 2. Given a set of historical instances and a specified number of pseudo-points, density estimates (solid blue lines) are made for historical time points ($time = 0$ and $time = 1$). Then, a mixture model with each pseudo-point as a single Gaussian component is formulated. Assuming polynomial trajectories (red solid lines) for the pseudo-points, the parameters of this model are the coefficients of the pseudo-points polynomial trajectories, which are learned using Expectation-Maximisation. Finally, for a future time point ($time = 2$), the pseudo-point's positions are predicted using the polynomial function, and the density (red dotted line) at this time point is estimated using kernel density estimation over the pseudo-points placed at their extrapolated positions.

3.3 Discussion

Both approaches above rely on a regression over time for extrapolating trends in the development of either weights or positions. In order to make this extrapolation more robust, we recommend using regularised trend functions that consider penalties for the models' complexities. The choice of the type of regression function depends on the type of drift, as for example polynomial functions require gradual drift, while trigonometric functions seem to be interesting candidates for modelling recurring context.

The weight-extrapolation in the first approach requires a normalisation, such that the extrapolated weights are all non-negative and sum up to one. An important question concerns the choice of the pseudo-point's location in this approach,

Fig. 1. Temporal Density Extrapolation Using Weight-Extrapolated Pseudopoints

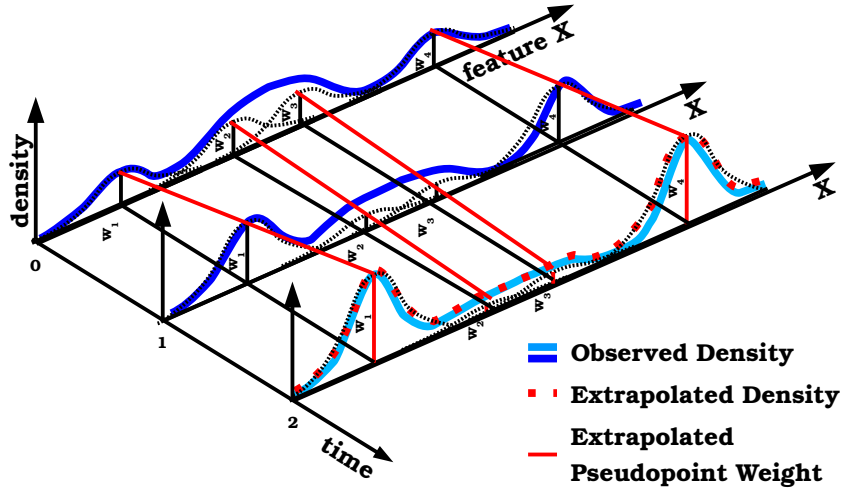
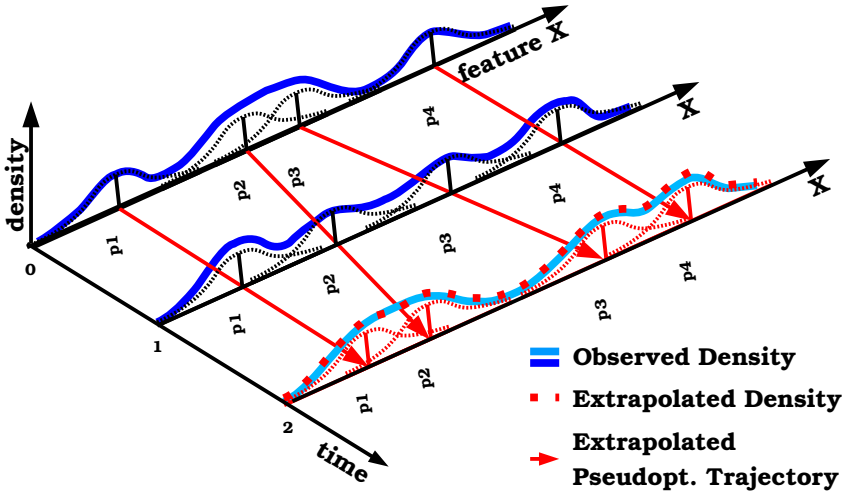


Fig. 2. Temporal Density Extrapolation Using Position-Extrapolated Pseudopoints



as it influences the precision of the extrapolated values: in regions with sparse pseudo-point populations, the model is less flexible than in densely populated ones. Therefore, this approach seems better suited for constricted (bounded) feature spaces. A simple equidistant placement of pseudo-points distributes the precision over the whole feature space. Alternatively, the pseudo-points might be placed at the coordinates of a subsample of the observed instances, thus concentrating the precision on areas with previously high density. However, if densities change largely over time, these areas might become less relevant.

In contrast, the second, position-extrapolating approach determines the positions of each pseudo-point automatically. It aims to adjust the future location of the pseudo-points such that they are densely placed in regions with a high expected density. However, in the case polynomial regression functions are used, a potential drawback is that their trajectories diverge in the long run. Thus, in contrast to the first approach, the second one seems to be better suited for infinite (unbounded) feature spaces.

Related to the choice of the pseudo-points' placements is the question of optimal bandwidth selection, which for kernel density estimation has already been reviewed in [18]. In short, we expect that with an increasing number of pseudo-points the optimal bandwidth decreases, while the extrapolation's precision increases. Furthermore, the number of pseudo-points is also an upper bound on the number of modes that both approaches are able to model.

4 Conclusion

In this paper, we have addressed the problem of *temporal density extrapolation*, where the objective is the prediction of a (potentially multi-modal) density distribution at *future time points*, given a sample of historical instances observed at different positions in feature space and at different times in the past. Two approaches based on pseudo-points were sketched: the first uses an extrapolation of time-varying weights of stationary located pseudo-points, while the second uses an extrapolation of the trajectory of the time-varying location of pseudo-points with uniform weights. Subsequently, these extrapolated pseudo-points are used in a kernel density estimation at future time points.

Having sketched the idea of the two temporal density extrapolation approaches, a more detailed specification and evaluation of these methods needs to be done in future work. Furthermore, the conjectures in the discussion above, in particular the usability of each approach for bounded and unbounded feature spaces, need to be verified. Finally, a known challenge for kernel-based approaches is the curse of dimensionality on multi-dimensional data. A naive approach is to combine multiple univariate temporal density extrapolations. However, an optimisation for multi-variate problems by using either projection pursuit [4] or copula [12] techniques seems worth investigating.

Acknowledgments. We thank Vera Hofer from Karl-Franzens-University Graz, Stephan Möhring, and Andy Koch for insightful discussions.

References

1. Aggarwal, C.C.: A framework for diagnosing changes in evolving data streams. In: Proceedings of the ACM SIGMOD Conference (2003)
2. Aggarwal, C.C.: On change diagnosis in evolving data streams. IEEE Transactions on Knowledge and Data Engineering 17(5), 587–600 (2005)
3. Böttcher, M., Höppner, F., Spiliopoulou, M.: On exploiting the power of time in data mining. ACM SIGKDD Explorations Newsletter 10(2), 3–11 (2008)
4. Friedman, J.H., Stuetzle, W., Schroeder, A.: Projection pursuit density estimation. Journal of the American Statistical Association 79(387) (1984)
5. Hofer, V., Krempl, G.: Drift mining in data: A framework for addressing drift in classification. Computational Statistics and Data Analysis 57(1), 377–391 (2013)
6. Hooker, G.: Diagnosing extrapolation: Tree-based density estimation. In: Knowledge Discovery in Databases (KDD) (2004)
7. Jeon, J., Taylor, J.W.: Using conditional kernel density estimation for wind power density forecasting. Journal of the American Statistical Association 107 (2012)
8. Krempl, G.: The algorithm APT to classify in concurrence of latency and drift. In: Gama, J., Bradley, E., Hollmén, J. (eds.) Advances in Intelligent Data Analysis X, Lecture Notes in Computer Science, vol. 7014, pp. 222–233. Springer (2011)
9. Krempl, G., Hofer, V.: Classification in presence of drift and latency. In: Spiliopoulou, M., Wang, H., Cook, D., Pei, J., Wang, W., Zaïane, O., Wu, X. (eds.) Proceedings of the 11th IEEE International Conference on Data Mining Workshops (ICDMW 2011). IEEE (2011)
10. Krempl, G., Zliobaitė, I., Brzeziński, D., Hüllermeier, E., Last, M., Lemaire, V., Noack, T., Shaker, A., Sievi, S., Spiliopoulou, M., Stefanowski, J.: Open challenges for data stream mining research. SIGKDD Explorations 16(1), 1–10 (2014), special Issue on Big Data
11. Nadaraya, E.A.: Nonparametric estimation of probability densities and regression curves. Kluwer (1989), originally published in Russian by Tbilisi University Press, Translated by S. Klotz
12. Nelsen, R.B.: An Introduction to Copulas. Springer (1999)
13. Parzen, E.: On estimation of a probability density function and mode. Annals of Mathematical Statistics 33, 1065–1076 (1962)
14. Rosenblatt, M.: Remarks on some non-parametric estimates of a density function. Annals of Mathematical Statistics 27(3), 832–837 (1956)
15. Skouras, K., Dawid, A.P.: On efficient probability forecasting systems. Biometrika 86(4), 765–784 (1999)
16. Tay, A.S., Wallis, K.F.: Density forecasting: A survey. Companion to Economic Forecasting pp. 45–68 (2002)
17. Terrell, G.R.: Tail probabilities by density extrapolation. In: Proceedings of the Annual Meeting of the American Statistical Association (2001)
18. Turlach, B.A.: Bandwidth selection in kernel density estimation: A review. Tech. Rep. 9307, Humboldt University, Statistic und Ökonometrie (1991)

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Fuzzy Clustering of Series Using Quantile Autocovariances

Borja Lafuente-Rego and Jose A. Vilar
Research Group on Modeling, Optimization and Statistical Inference (MODES),
Department of Mathematics, Computer Science Faculty, University of A Coruña

Abstract. Unlike conventional clustering, fuzzy cluster analysis allows data elements to belong to more than one cluster by assigning membership degrees of each data to clusters. This work proposes a fuzzy C–medoids algorithm to cluster time series based on comparing their estimated quantile autocovariance functions. The behaviour of the proposed algorithm is studied on different simulated scenarios and its effectiveness is concluded by comparison with alternative approaches.

1 Introduction

In classical cluster analysis each datum is assigned to exactly one cluster, thus producing a "hard" partition of the data set into several disjoint subsets. This approach can be inadequate in the presence of data objects that are equally distant to two ore more clusters. Fuzzy cluster analysis allows gradual memberships of data objects to clusters, thus providing versatility to reflect the certainty with which each data is assigned to the different clusters. An interesting overview of present fuzzy clustering methods is provided by [3]. Interest in this approach has increased in recent years. Proof of this is the large amount of publications in this field (e.g. [6] and [5]).

In this paper, a fuzzy C–medoids algorithm to cluster time series using the quantile autocovariance functions is proposed. Motivation behind this approach is twofold. First, quantile autocovariances have shown a high capability to cluster time series generated from a broad range of dependence models [10]. On the other hand, the use of a fuzzy approach for clustering time series is justified in order to gain adaptivity for constructing the centroids and to obtain a better characterization of the temporal pattern of the series (see discussion in [7]). To illustrate the merits of the proposed algorithm, an extensive simulation study comparing our fuzzy approach with other fuzzy procedures has been carried out. Specifically, we have focused on the classification of heteroskedastic models, which are of great importance in many applications (e.g. to model many financial time series) and have received relatively little attention in the clustering literature.

2 A dissimilarity based on quantile autocovariances

Consider a set of p series $S = \{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(p)}\}$, with $\boldsymbol{X}^{(j)} = (X_1^{(j)}, \ldots, X_T^{(j)})$ being a T-length partial realization from a real valued process $\{X_t^{(j)},\ t \in \mathbb{Z}\}$.

We wish to perform cluster analysis on S in such a way that series with similar generating processes are grouped together. To achieve this goal, we propose to measure dissimilarity between two series by comparing the estimators of their quantile autocovariance functions (QAF), which are formally defined below.

Let $X_1, \ldots, X_T$ an observed stretch of a strictly stationary process $\{X_t;\ t \in \mathbb{Z}\}$. Denote by F the marginal distribution of X_t and by $q_\tau = F^{-1}(\tau)$, $\tau \in [0,1]$, the corresponding quantile function. Fixed $l \in \mathbb{Z}$ and an arbitrary couple of quantile levels $(\tau, \tau') \in [0,1]^2$, consider the cross-covariance of the indicator functions $I\left(X_t \leq q_\tau\right)$ and $I\left(X_{t+l} \leq q_{\tau'}\right)$ given by

$$\gamma_l(\tau, \tau') = \operatorname{cov}\left\{I\left(X_t \leq q_\tau\right), I\left(X_{t+l} \leq q_{\tau'}\right)\right\} = \mathbb{P}\left(X_t \leq q_\tau, X_{t+l} \leq q_{\tau'}\right) - \tau\,\tau'. \tag{1}$$

Function $\gamma_l(\tau, \tau')$, with $(\tau, \tau') \in [0,1]^2$, is called *quantile autocovariance function of lag l*. Replacing in (1) the theoretical quantiles of the marginal distribution F, q_τ and $q_{\tau'}$, by the corresponding empirical quantiles based on $X_1, \ldots, X_T$, $\hat{q}_\tau$ and $\hat{q}_{\tau'}$, we obtain the estimated quantile autocovariance function given by

$$\hat{\gamma}_l(\tau, \tau') = \frac{1}{T-l} \sum_{t=1}^{T-l} I\left(X_t \leq \hat{q}_\tau\right) I\left(X_{t+l} \leq \hat{q}_{\tau'}\right) - \tau\,\tau'. \tag{2}$$

As the quantile autocovariances are able to account for high level dynamic features, a simple dissimilarity criterion between two series $X_t^{(1)}$ and $X_t^{(2)}$ consists in comparing their estimated quantile autocovariances on a common range of selected quantiles. Thus, for L prefixed lags, $l_1, \ldots, l_L$, and r quantile levels, $0 < \tau_1 < \ldots < \tau_r < 1$, we construct the vectors $\boldsymbol{\Gamma}^{(u)}$, $u = 1, 2$, given by

$$\boldsymbol{\Gamma}^{(u)} = \left(\boldsymbol{\Gamma}_{l_1}^{(u)}, \ldots, \boldsymbol{\Gamma}_{l_L}^{(u)}\right), \quad \text{with} \quad \boldsymbol{\Gamma}_{l_i}^{(u)} = \left(\hat{\gamma}_{l_i}^{(u)}(\tau_j, \tau_k);\ j, k = 1 \ldots, r\right), \tag{3}$$

for $i = 1, \ldots, L$, and $\hat{\gamma}$ given in (2). Then, the distance between $X_t^{(1)}$ and $X_t^{(2)}$ is defined as the squared Euclidean distance between their representations $\boldsymbol{\Gamma}^{(1)}$ and $\boldsymbol{\Gamma}^{(2)}$, i.e.

$$d_{QAF}\left(X_t^{(1)}, X_t^{(2)}\right) = ||\boldsymbol{\Gamma}^{(1)} - \boldsymbol{\Gamma}^{(2)}||_2^2 \tag{4}$$

Computing d_{QAF} for all pairs of series in S allows us to set a pairwise dissimilarity matrix, which can be taken as starting point of a conventional hierarchical clustering algorithm. Alternatively, a partitioning clustering, such as the k-means algorithm, can be performed averaging the $\boldsymbol{\Gamma}$ representations to determine the centroids. Then, d_{QAF} is also used to calculate the distances between series and centroids involved in the iterative refinement of the cluster solution.

3 Fuzzy clustering based on quantile autocovariances

Time series are dynamic objects and therefore different temporal patterns may be necessary to characterize the serial behaviour in different periods of time. In

other words, the series are not distributed accurately within a given number of clusters, but they can belong to two or even more clusters. This problem can be adequately treated using a fuzzy clustering procedure, which associates a fuzzy label vector to each element stating its memberships to the set of clusters. In this section we propose a fuzzy C-medoids clustering algorithm for time series by plugging the QAF-dissimilarity introduced in Section 2.

Let $S = \left\{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(p)}\right\}$ be a set of p time series and $\boldsymbol{\Gamma} = \left\{\boldsymbol{\Gamma}^{(1)}, \ldots, \boldsymbol{\Gamma}^{(p)}\right\}$ a set of quantile autocovariances selected to perform clustering. The fuzzy C-medoids clustering finds the subset of $\boldsymbol{\Gamma}$, $\tilde{\boldsymbol{\Gamma}} = \left\{\tilde{\boldsymbol{\Gamma}}^{(1)}, \ldots, \tilde{\boldsymbol{\Gamma}}^{(C)}\right\}$, and the $p \times C$ matrix of fuzzy coefficients $\Omega = (u_{i,c})$ that lead to solve the minimization problem:

$$\min_{\tilde{\Gamma}, \Omega} \sum_{i=1}^{p} \sum_{c=1}^{C} u_{ic}^{m} \left\| \boldsymbol{\Gamma}^{(i)} - \boldsymbol{\Gamma}^{(c)} \right\|_2^2, \text{ subject to } \sum_{c=1}^{C} u_{ic} = 1 \text{ and } u_{ic} \geq 0. \quad (5)$$

Each $u_{ic} \in [0, 1]$ represents the membership degree of the i-th series to the c-th cluster and the parameter $m > 1$ controls the fuzziness of the partition. As the value of m increases, the boundaries between clusters become softer and therefore the classification is fuzzier. If $m = 1$, the hard version of the clustering procedure is obtained, i.e. $u_{ic} \in \{0, 1\}$, that leads to a classical K-means partition of S. The constraints $\sum_{c=1}^{C} u_{ic} = 1$ and $u_{ic} \geq 0$ ensure that no cluster is empty and that all series are included in the cluster partition.

The objective function (5) cannot be minimized directly, and an iterative algorithm that alternately optimizes the membership degrees and the medoids must be used. The update formula for the membership degrees is given by

$$u_{ic} = \left[\sum_{c'=1}^{C} \left(\frac{\left\| \boldsymbol{\Gamma}^{(i)} - \boldsymbol{\Gamma}^{(c)} \right\|_2^2}{\left\| \boldsymbol{\Gamma}^{(i)} - \boldsymbol{\Gamma}^{(c')} \right\|_2^2} \right)^{\frac{1}{m-1}} \right]^{-1}, \quad \text{for } i = 1, \ldots, p. \quad (6)$$

Then, the QAF-based fuzzy C–medoids clustering algorithm is implemented as follows.

i. Pick an initial set of medoids $\widetilde{\boldsymbol{\Gamma}} = \left\{\widetilde{\boldsymbol{\Gamma}}^{(1)}, \ldots, \widetilde{\boldsymbol{\Gamma}}^{(C)}\right\}$ and the fuzzifier m.

ii. Set $\widetilde{\boldsymbol{\Gamma}}_{\text{OLD}} = \widetilde{\boldsymbol{\Gamma}}$.

iii. Compute u_{ic} using (6).

iv. Update the medoids, let's say $\widehat{\boldsymbol{\Gamma}} = \left\{\widehat{\boldsymbol{\Gamma}}^{(1)}, \ldots, \widehat{\boldsymbol{\Gamma}}^{(C)}\right\}$, by minimizing the objective function with the new u_{ic}. Denote by

$$q = \text{argmin}_{1 \leq i' < p} \sum_{i''=1}^{p} u_{i''c}^{m} \left\| \boldsymbol{\Gamma}^{(i'')} - \boldsymbol{\Gamma}^{(i')} \right\|_2^2$$

If the value of q is lower than the one obtained with $\widetilde{\boldsymbol{\Gamma}}$, then $\widetilde{\boldsymbol{\Gamma}} = \widehat{\boldsymbol{\Gamma}}$.

v. If $\widetilde{\boldsymbol{\Gamma}}_{\text{OLD}} = \widetilde{\boldsymbol{\Gamma}}$ or a maximum number of iterations is achieved then end algorithm. Otherwise, return to step ii.

The total number of clusters C has to be preset. For this task classical indexes such as silhouette width or Krazanowski-Lai index can be used.

4 Simulation Study

The proposed fuzzy algorithm was tested against two other fuzzy clustering algorithms via simulation. In particular, the classification of heteroskedastic time series was considered by simulating two different scenarios formed by (i) GARCH(1,1) models and (ii) different structures of conditional heteroskedasticity. The selected generating models at each case are detailed below.

- **Scenario 1:** Consider $X_t = \mu_t + a_t$, with $\mu_t \sim$ AR(1) and $a_t = \sigma_t \epsilon_t$, $\epsilon_t \sim \mathcal{N}(0,1)$. Then, the following GARCH(1,1) structures for the varying conditional variance are considered:

  M1: $\sigma_t^2 = 0.1 + 0.01a_{t-1}^2 + 0.9\sigma_{t-1}^2$ M3: $\sigma_t^2 = 0.1 + 0.1a_{t-1}^2 + 0.1\sigma_{t-1}^2$
  M2: $\sigma_t^2 = 0.1 + 0.9a_{t-1}^2 + 0.01\sigma_{t-1}^2$ M4: $\sigma_t^2 = 0.1 + 0.4a_{t-1}^2 + 0.5\sigma_{t-1}^2$

- **Scenario 2:** Consider $X_t = \mu_t + a_t$, with $\mu_t \sim$ MA(1) and $a_t = \sigma_t \epsilon_t$, $\epsilon_t \sim \mathcal{N}(0,1)$. Then, the following ARCH(1), GARCH(1,1), GJR-GARCH and EGARCH structures are considered for the varying conditional variance:

  M1: $\sigma_t^2 = 0.1 + 0.8a_{t-1}^2$
  M2: $\sigma_t^2 = 0.1 + 0.1a_{t-1}^2 + 0.8\sigma_{t-1}^2$
  M3: $\sigma_t^2 = 0.1 + (0.25 + 0.3\mathrm{N}_{t-1})a_{t-1}^2 + 0.5\sigma_{t-1}^2$; $N_{t-1} = \mathrm{I}(a_{t-1} < 0)$
  M4: $\ln(\sigma_t^2) = 0.1 + \epsilon_{t-1} + 0.3\left[|\epsilon_{t-1}| - \mathbb{E}(|\epsilon_{t-1}|)\right] + 0.4\ln(\sigma_{t-1}^2)$

In all cases ϵ_t consisted of independent zero-mean Gaussian variables with unit variance. For each scenario, five series of length $T = 200$ were generated from each model over $N = 100$ trials.

Two fuzzy clustering algorithms specifically designed to deal with GARCH models were used and compared with our proposal. Both algorithms rely on different dissimilarity measures constructed using the AR representation of a GARCH(p,q) process given by

$$\sigma_t^2 = \gamma + \sum_{i=1}^{p} \alpha_i a_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \tag{7}$$

with $\gamma > 0$, $0 \leq \alpha_i < 1$ and $0 \leq \beta_j < 1$, for $i = 1, \ldots, p$ and $j = 1, \ldots, q$, and $(\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j) < 1$. Then, the dissimilarities are defined as follows.

1. Dissimilarity based on the autoregressive representation of the GARCH models [12, 11]. Given $\boldsymbol{X}^{(k)}$ and $\boldsymbol{X}^{(k')}$ in S, we define

$$d_{AR}^2(\boldsymbol{X}^{(k)}, \boldsymbol{X}^{(k')}) = \sum_{r=1}^{R} \left(\widehat{\pi}_{rk} - \widehat{\pi}_{rk'}\right)^2,$$

   with $\widehat{\pi}_{rz}$ an estimator of the r-th coefficient $\pi_r = (\alpha_r + \beta_r) + \sum_{j=1}^{\min(q,r)} \beta_j \pi_{r-j}$, for the series z, $z = k, k'$. Parameter R determines the maximum number of

autoregressive coefficients π_r. A GARCH–based fuzzy C-medoids clustering is proposed in [4] by considering the optimization problem:

$$\min_{\hat{\Pi},\Omega} \sum_{i=1}^{p} \sum_{c=1}^{C} u_{ic}^{m} \sum_{r=1}^{R} \left(\widehat{\pi}_{ri} - \widehat{\pi}_{rc}\right)^2, \text{ subject to } \sum_{c=1}^{C} u_{ic} = 1 \text{ and } u_{ic} \geq 0. \quad (8)$$

2. GARCH-based distance measure [1] given by

$$d_{\text{GARCH}}(\boldsymbol{X}^{(k)}, \boldsymbol{X}^{(k')}) = (\boldsymbol{L_k} - \boldsymbol{L_{k'}})' (\boldsymbol{V_k} + \boldsymbol{V_{k'}})^{-1} (\boldsymbol{L_k} - \boldsymbol{L_{k'}}) \quad (9)$$

with $\boldsymbol{L_j} = \left(\widehat{\boldsymbol{\alpha}}_{\boldsymbol{j}}, \widehat{\boldsymbol{\beta}}_{\boldsymbol{j}}\right)$ the vector of estimated parameters and $\boldsymbol{V_j}$ the estimated covariance matrix for $\boldsymbol{L_j}$, for $j = k, k'$. An alternative GARCH-based fuzzy C-medoids clustering is proposed in [4] by minimizing:

$$\sum_{i=1}^{p} \sum_{c=1}^{C} u_{ic}^{m} \left[(\boldsymbol{L_i} - \boldsymbol{L_c})' (\boldsymbol{V_i} + \boldsymbol{V_c})^{-1} (\boldsymbol{L_i} - \boldsymbol{L_c}) \right], \quad (10)$$

subject to $\sum_{c=1}^{C} u_{ic} = 1$ and $u_{ic} \geq 0$.

The three fuzzy clustering algorithms were performed using a fuzziness parameter $m = 1.5$ on $N = 100$ trials for each scenario. At each trial, the quality of the clustering procedure was evaluated comparing the experimental cluster solution with the true cluster partition. Two different agreement measures were used, namely the Gavrilov index [8] and the adjusted Rand index [9]. The mean values and standard deviations of these indexes based on the 100 trials using both hard and fuzzy cluster analysis are provided in Table 1.

Table 1. Averages and standard deviations (in brackets) of two cluster similarity indexes obtained from 100 trials.

		Scenario 1		Scenario 2	
		Gavrilov	Adj. Rand	Gavrilov	Adj. Rand
Hard cluster	d_{AR}	0.859 (.109)	0.685 (.198)	0.712 (.146)	0.469 (.215)
	d_{GARCH}	0.574 (.059)	0.286 (.072)	0.504 (.078)	0.137 (.116)
	d_{QAF}	0.843 (.109)	0.726 (.152)	0.918 (.081)	0.825 (.135)
Fuzzy cluster	d_{AR}	0.541 (.056)	0.271 (.080)	0.486 (.076)	0.128 (.100)
	d_{GARCH}	0.553 (.088)	0.241 (.132)	0.535 (.076)	0.188 (.107)
	d_{QAF}	0.842 (.116)	0.704 (.181)	0.925 (.072)	0.833 (.125)

Results from Table 1 show that the metrics based on quantile autocovariances and on the AR representation led to the best scores in Scenario 1 when the hard cluster is carried out. When the fuzzy approaches were considered, the behaviour of the d_{AR} substantially worsened, while the very similar (even somewhat higher)

results were obtained with d_{QAF}. The worst results were obtained with the GARCH-based dissimilarity both for the hard and the fuzzy versions.

The metric based on quantile autocovariances also obtained the best results in Scenario 2, with indexes of agreement above 0.8 and a slight improvement by using the fuzzy clustering. The GARCH-based metrics, d_{AR} and d_{GARCH} where strongly affected by the model misspecification and produced the worst results for both the hard and the fuzzy versions of the cluster analysis.

To assess the effect of the fuzziness parameter in the partitions the algorithm was implemented for several values of m. However this results were here omitted due to the limitation of space.

5 A case study

In this section, the proposed fuzzy C–medoids clustering algorithm is used to perform clustering on a set of series of electricity demand. Specifically, our database consists of hourly electricity demand in the Spanish market from 1st January 2011 to 31th December 2012. All data are sourced from the official website of Operador del Mercado Iberico de Energia[1]. Records corresponding to Saturdays and Sundays have been removed from the database because electricity demand is lower in the weekends. Thus we have 24 time series (one for each hour of the day) of length $T = 731$. Since all series are non–stationary in mean, the original series are transformed taking one regular difference.

Table 2 presents the membership degrees for the case with two and three clusters. The results obtained for the two–cluster partition formed by $C_1 = \{H24, H1, H2, H3, H4, H5, H6, H7\}$ and C_2 grouping the remaining series. The cluster C_1 corresponds with the hours of the day where the electricity demand is low, while the C_2 identifies the time of the day when the power consumption is greater. In the case of the three–cluster partition, the cluster C_1 is divided in two subclusters. One formed with the hours of the day with the lowest demand of electricity, and a second cluster with an intermediate electricity consumption.

6 Concluding remarks

In this paper, we focus on the classification of time series featuring a fuzzy clustering algorithm in the framework of a partitioning around medoids. A dissimilarity–based approach is considered. In particular, we propose a C–medoids fuzzy clustering algorithm using an innovative dissimilarity measure based on the quantile autocovariances (d_{QAF}).

The simulation study shows that the proposed dissimilarity produces satisfactory results by performing fuzzy cluster analysis. The proposed clustering algorithm was tested against two GARCH–based fuzzy clustering algorithm present in the literature in two different heteroskedastic scenarios. The fuzzy clustering algorithm based on d_{QAF} led to the best results. In fact, apart from d_{QAF}, none

[1] `http://http://www.omel.es/files/flash/ResultadosMercado.swf`

Table 2. Membership degrees obtained with QAF–based FCM with $m = 1.5$ considering 2 and 3 clusters.

	2 Clusters			3 Clusters			
	Membership degrees		Crisp	Membership degrees			Crisp
	C_1	C_2		C_1	C_2	C_3	
H1	0.63044	0.36956	1	1.00000	0.00000	0.00000	1
H2	1.00000	0.00000	1	0.99878	0.00098	0.00024	1
H3	0.98282	0.01718	1	0.99484	0.00395	0.00121	1
H4	0.94118	0.05882	1	0.99229	0.00574	0.00197	1
H5	1.00000	0.00000	1	0.92388	0.06811	0.00801	1
H6	0.99923	0.00077	1	0.30793	0.66185	0.03021	2
H7	0.98282	0.01718	1	0.00000	1.00000	0.00000	2
H8	0.00003	0.99997	2	0.05793	0.09475	0.84733	3
H9	0.00003	0.99997	2	0.00097	0.00294	0.99610	3
H10	0.00077	0.99923	2	0.00045	0.00149	0.99806	3
H11	0.00077	0.99923	2	0.00171	0.00507	0.99323	3
H12	0.00002	0.99998	2	0.00011	0.00049	0.99940	3
H13	0.00002	0.99998	2	0.00027	0.00178	0.99794	3
H14	0.00002	0.99998	2	0.00032	0.00107	0.99861	3
H15	0.00002	0.99998	2	0.00134	0.00940	0.98927	3
H16	0.00002	0.99998	2	0.00088	0.00636	0.99277	3
H17	0.00002	0.99998	2	0.00000	0.00000	1.00000	3
H18	0.00056	0.99944	2	0.00051	0.00498	0.99451	3
H19	0.00000	1.00000	2	0.00002	0.00014	0.99984	3
H20	0.00003	0.99997	2	0.00020	0.00135	0.99846	3
H21	0.00056	0.99944	2	0.00047	0.00384	0.99569	3
H22	0.01718	0.98282	2	0.00136	0.01488	0.98377	3
H23	0.00003	0.99997	2	0.01539	0.13091	0.85370	3
H24	0.99998	0.00002	1	0.00206	0.98054	0.01740	2

of the remaining examined dissimilarities shown acceptable results by clustering heteroskedastic processes, thus emphasizing the usefulness of d_{QAF} in this framework.

Note that a limitation of our procedure is that series are assumed to be strictly stationary and hence further research must be carried out. Although we have followed a dissimilarity-based approach, it is worthy to emphasize that model-based techniques can be also an interesting alternative. Likewise the fuzzy approach, the use of probabilistic models such as mixture models (see e.g. [2]) allows us to assign each datum to one single cluster although this assignment relies on a probabilistic approach since the mixing proportions are estimated from the data. Unlike the fuzzy approach, no fuzziness parameter is required by using mixture models, although the model selection problem must be solved in the latter case.

References

1. Caiado, J., Crato, N.: A garch-based method for clustering of financial time series: International stock markets evidence. Mpra paper, University Library of Munich, Germany (2007), http://EconPapers.repec.org/RePEc:pra:mprapa:2074
2. Chen, W.C., Maitra, R.: Model-based clustering of regression time series data via apecman aecm algorithm sung to an even faster beat. Statistical Analysis and Data Mining 4(6), 567–578 (2011)

3. Döring, C., Lesot, M.J., Kruse, R.: Data analysis with fuzzy clustering methods. Computational Statistics & Data Analysis 51(1), 192 – 214 (2006)
4. D'Urso, P., Cappelli, C., Lallo, D.D., Massari, R.: Clustering of financial time series. Physica A 392(9), 2114–2129 (2013)
5. D'Urso, P., Giovanni, L.D., Massari, R.: Time series clustering by a robust autoregressive metric with application to air pollution 141(15), 107–124 (2015)
6. D'Urso, P., Giovanni, L.D., Massari, R., Lallo, D.D.: Noise fuzzy clustering of time series by the autoregressive metric 71(3), 217–243 (2013)
7. D'Urso, P., Maharaj, E.A.: Autocorrelation-based fuzzy clustering of time series. Fuzzy Sets Syst. 160(24), 3565–3589 (2009)
8. Gavrilov, M., Anguelov, D., Indyk, P., Motwani, R.: Mining the stock market (extended abstract): Which measure is best? In: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 487–496. KDD'00, ACM, New York, USA (2000)
9. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. 2(1), 193–218 (1985)
10. Lafuente-Rego, B., Vilar, J.A.: Clustering of time series using quantile autocovariances. Advances in Data Analysis and Classification pp. 1–25 (2015)
11. Maharaj, E.A.: Clusters of time series. J. Classification 17(2), 297–314 (2000)
12. Piccolo, D.: A distance measure for classifying arima models. J. Time Series Anal. 11(2), 153–164 (1990)

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Causality on Longitudinal Data: Stable Specification Search in Constrained Structural Equation Modeling

Ridho Rahmadi[1,2], Perry Groot[2], Marianne Heins[4],
Hans Knoop[3], and Tom Heskes[2]
[1] Department of Informatics, Universitas Islam Indonesia.
[2] Institute for Computing and Information Sciences, Radboud University Nijmegen.
[3] Expert Centre for Chronic Fatigue, Radboud University Medical Centre, Nijmegen.
[4] Netherlands Institute for Health Services Research, Utrecht.
r.rahmadi@cs.ru.nl

Abstract Developing causal models from observational longitudinal studies is an important, ubiquitous problem in many disciplines. A disadvantage of current causal discover algorithms, however, is the inherent instability in structure estimation. With finite data samples small changes in the data can lead to completely different optimal structures. The present work presents a new causal discovery algorithm for longitudinal data that is robust for finite data samples. We validate our approach on a simulated data set and real-world data on Chronic Fatigue Syndrome patients.

Keywords: Longitudinal data, Causal modeling, Structural equation model, Stability selection, Multi-objective evolutionary algorithm.

1 Introduction

Developing causal models from observational longitudinal studies is an important, ubiquitous problem in many disciplines, which has led to the development of a variety of causal discovery algorithms in the literature [1–5]. A disadvantage of current causal discovery algorithms, however, is the inherent instability in structure learning. With finite data samples small changes in the data can lead to completely different optimal structures, since errors made by the discovery algorithm may be propagated and lead to further errors [6]. In [7] we developed a robust causal discovery algorithm for cross-sectional data. The method performs structure search over Structural Equation Models (SEMs) by maximizing model scores in terms of data fit and complexity. The present work extends our causal discovery algorithm to longitudinal data. We describe how longitudinal causal relationships can be modelled for an arbitrary number of time slices. Furthermore, we show how a longitudinal causal model can easily be scored using standard SEM software by data reshaping. The algorithm produces accurate structure estimates and is shown to be robust for finite samples. We validate our approach on one simulated longitudinal data set and one real-world longitudinal data set for Chronic Fatigue Syndrome.

2 Proposed method

We use a SEM for causal modeling. The general form of the equations is

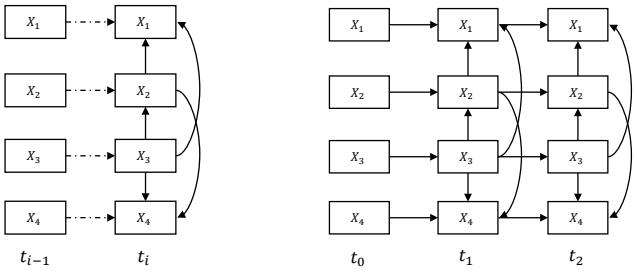
$$x_i = f_i(\mathrm{pa}_i, \varepsilon_i), \quad i = 1, \ldots, n. \tag{1}$$

where pa_i denotes the *parents* which represent the set of variables considered to be direct causes of X_i and ε_i represents errors on account of omitted factors that are assumed to be mutually independent [8]. In this study, we focus on causal models with no reciprocal relationships, and no latent variables. Thus the causal model can also be represented by a *Directed Acyclic Graph* (DAG). We score models using both the *chi-square* χ^2 (measuring the data fit) and the *model complexity* (measuring the number of parameters).

We use the method we developed in [7] to perform exploratory search over SEM models. Based on the idea of stability selection [9], the method subsamples the data D with size $\lfloor |D|/2 \rfloor$ without replacement and generates Pareto optimal models for each subset. After that, all Pareto optimal models are transformed into their corresponding model equivalent classes, called *Completed Partially Directed Acyclic Graph* (CPDAG) [10]. From these CPDAGs we compute the edge and causal path stability graph, such as Figure 3a, by grouping them according to model complexity and computing their *selection probability*, i.e., the number of occurrences divided by the total number of models for a certain level of model complexity. Stability selection is then performed by specifying two thresholds, π_{sel} (boundary of selection probability) and π_{bic} (boundary of complexity). For example, setting $\pi_{\mathrm{sel}} = 0.6$ means that all causal relationships with edge stability or causal path stability (Figure 3) above this threshold are considered *stable*. The second threshold π_{bic} is used to control overfitting. We set π_{bic} to the level of model complexity at which the minimum average *Bayesian Information Criterion* (BIC) score is found. For example, $\pi_{\mathrm{bic}} = 7$ means that all causal relationships with an edge stability or a causal path stability lower than this threshold (Figure 3) are considered *parsimonious*. Causal relationships that intersect with the top-left region are considered both stable and parsimonious and called *relevant*, from which we can derive a causal model.

The method in [7] only handles cross-sectional data. Based on the idea of "unrolling" the network in Dynamic Bayesian Networks [4, 5], we extended the method to handle longitudinal data. We model longitudinal causal relationships with a SEM model consisting of two time slices (Figure 1a) that can be "unrolled" into a network with an arbitrary number of time slices (Figure 1b). Time slice t_i represents the relationships *within* a time slice (intra-slice causal relationships, solid arcs in Figure 1a). Causal relationships *between* time slices (inter-slice causal relationships, dashed arcs in Figure 1a) always go forward in time, i.e., from time slice t_{i-1} to time slice t_i.

To score our models on longitudinal data we use data reshaping. In the reshaped data, the first n data points contain the relations that occur in the first two time slices t_0 and t_1. The next n data points contain the relations that occur in time slices t_1 and t_2. The i-th subset of n data points contain



(a) (b)

Figure 1: (a) The longitudinal causal model. (b) The "unrolled" causal graph used to generate longitudinal data. It contains four continuous variables $(X_1, \ldots, X_4)$ in three different time slices $t_0, \ldots, t_2$.

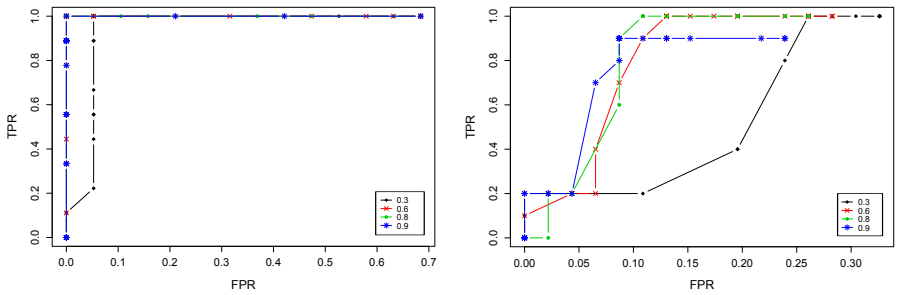
the relations in time slices t_{i-1} and t_i. The reshaped data then allows us to use standard SEM software to compute the scores.

3 Application to Simulated Data

For this experiment, we generated a longitudinal data set with 400 instances from a causal graph as depicted in Figure 1b. The data set consists of three time slices with four continuous variables for each time slice.[1] When we searched over SEM models we added prior knowledge that variables X_1 and X_2 do not cause variable X_3 directly. We performed the search over 200 subsets.

As the true model is known, we measure the performance of our method by means of the *Receiver Operating Characteristic* (ROC) [11] for both edges and causal paths. The threshold π_{sel} is fixed to a value ($\pi_{\text{sel}} \in \{0.3, 0.6, 0.8, 0.9\}$) while π_{bic} is varied. We compute the *True Positive Rate* (TPR) and the *False Positive Rate* (FPR) from the CPDAG of the true model. As for an example, in the case of edge stability, a true positive means that an edge that appears within the top-left region bounded by π_{sel} and π_{bic} also exists in the CPDAG of the true model. Figure 2 portrays the ROC curves for both edge and causal path stability. Generally we can see that higher values of π_{sel} tend to give better ROC curves. This suggests that our approach is able to find the underlying structure with high reliability scores. A notable point is that the ROC curves stop at a TPR and/or FPR value lower than 1. Since some of the edges and paths are disallowed (i.e., no edges in time-slice t_{i-1} and no paths from t_i to t_{i-1}) some of the edges and causal paths in the stability graphs end up with a selection probability of 0 and the result is that the ROC curves cannot reach the upper right corner with TPR = FPR = 1.

[1] Available at http://bit.ly/1L6dBOo



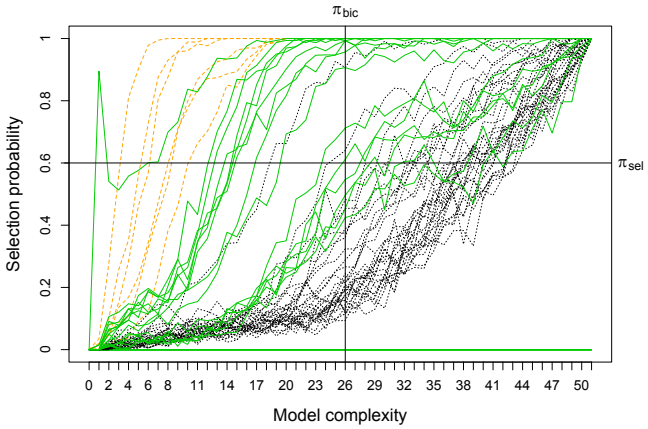
(a) (b)

Figure 2: A plot of ROC curves for (a) the edge stability and (b) the causal path stability, for different values of π_{sel}. A higher π_{sel} shows a better ROC curve. See the main text for an explanation why the ROC curves stop at some point.

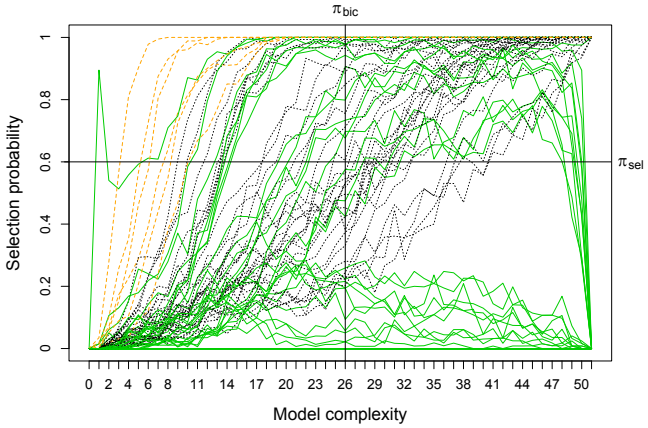
4 Application to Real-world Data

For an application to real-world data, we consider a data set about *Chronic Fatigue Syndrome* (CFS) which consists of 183 subjects and five time slices with six discrete variables [12]. The variables are, *fatigue* severity, the sense of *control* over fatigue, *focusing* on the symptoms, the objective activity of the patient (*oActivity*), the subject's perceived activity (*pActivity*), and the physical *functioning*. We use *Expectation Maximization* implemented in SPSS [13] to impute the missing values. As all of the variables have large scales, e.g., in the range between 0 to 155, we treat them as continuous variables. We added prior knowledge that the variable fatigue does not cause any of the other variables directly. We performed the search over 200 subsets.

Figure 3a shows that nineteen relevant edges were found, consisting of eleven intra-slice and eight inter-slice relationships which among of these, six are between the same variables and two are between different variables. Figure 3b shows that thirty-two relevant causal paths were found, consisting of twelve intra-slice and twenty inter-slice relationships which among of these, six are between the same variables and fourteen are between different variables. For a more intuitive representation, we combine the stability graphs into a model using the following procedure. First, the nodes are linked according to the nineteen relevant edges. Second, edges are oriented according to our background knowledge. Eight of the inter-slice relationships are oriented from time slice t_{i-1} to t_i and five of the intra-slice edges can be oriented since it is known that the variable fatigue does not directly cause any other variable. Third, the edges are oriented according to the relevant causal paths, which results in another twenty-eight directed edges. The inferred model is shown in Figure 4. Each edge is annotated with a reliability score which is the maximum score obtained in the top-left region of the edge stability graph.



(a)



(b)

Figure 3: The stability graphs for CFS together with π_{sel} and π_{bic}, yielding four regions. The top-left region contains the relevant causal relations. (a) The edge stability graph. (b) The causal path stability graph. Orange-dashed lines represent inter-slice relationships between the same variables, black-dotted lines represent inter-slice relationships between different variables, green-solid lines represent intra-slice relationships.

From the stability graphs we can see that the most stable causal relations are the inter-slice relations between the same variables followed by some of the intra-slice causal relations. Almost all of the inter-slice relations between different variables are not considered relevant. A directed edge $X \to Y$ in Figure 4 indicates

that a change in variable X causes a change in variable Y. In the intra-slice causal relationships, we found that all variables are direct causes for fatigue severity. We also found all variables, except *fatigue*, to be direct causes for the perceived activity. Furthermore, the variable *control* is a direct cause for both focusing on the symptoms and physical functioning. Generally the inter-slice relationships show direct causes between the same variables. In addition, the variables *pActivity* and *control* indicate a stronger direct cause for fatigue severity and focusing on symptoms, respectively, as they contribute a direct cause in both time slices. The inferred model is consistent with results reported in the medical literature [12, 14, 15].

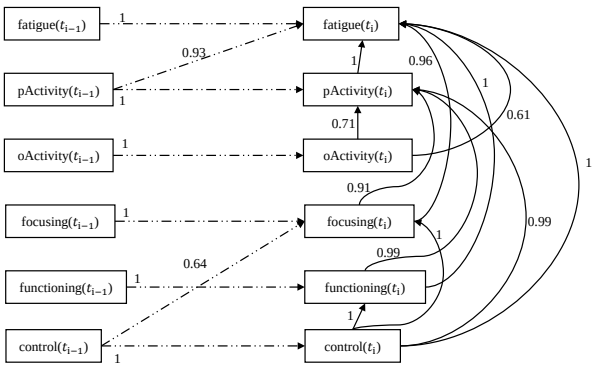


Figure 4: The inferred model of CFS by combining the edge stability and causal path stability graphs.

5 Conclusion

Causal discovery from longitudinal data is an important, ubiquitous problem in science. Current causal discovery algorithms, however, have difficulty dealing with the inherent instability in structure estimation. The present work introduces a new discovery algorithm for longitudinal data that is robust for finite samples. Experimental results on both artificial and real-world data sets show that the method results in reliable structure estimates. Future research will aim to estimate the size of causal effects.

Acknowledgments

The research leading to these results has received funding from the DGHE of Indonesia and the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 305697.

References

1. Riva, A., Bellazzi, R.: Learning temporal probabilistic causal models from longitudinal data. Artificial Intelligence in Medicine **8**(3) (1996) 217–234
2. Parner, J., Arjas, E.: Causal reasoning from longitudinal data. Rolf Nevanlinna Inst., University of Helsinki (1999)
3. Marsh, H.W., Yeung, A.S.: Causal effects of academic self-concept on academic achievement: Structural equation models of longitudinal data. Journal of educational psychology **89**(1) (1997) 41–54
4. Friedman, N., Murphy, K., Russell, S.: Learning the structure of dynamic probabilistic networks. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. (1998) 139–147
5. Murphy, K., Mian, S., et al.: Modelling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA (1999)
6. Spirtes, P.: Introduction to causal inference. The Journal of Machine Learning Research **11** (2010) 1643–1662
7. Rahmadi, R., Groot, P., Heins, M., Knoop, H., Heskes, T., The OPTIMISTIC consortium: Causality on cross-sectional data: Stable specification search in constrained structural equation modeling. arXiv:1506.05600 [stat.ML] (2015)
8. Pearl, J.: Causality: models, reasoning and inference. Cambridge Univ Press (2000)
9. Meinshausen, N., Bühlmann, P.: Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **72**(4) (2010) 417–473
10. Chickering, D.M.: Learning equivalence classes of Bayesian-network structures. The Journal of Machine Learning Research **2** (2002) 445–498
11. Fawcett, T.: ROC graphs: Notes and practical considerations for researchers. Machine learning **31** (2004) 1–38
12. Heins, M.J., Knoop, H., Burk, W.J., Bleijenberg, G.: The process of cognitive behaviour therapy for chronic fatigue syndrome: Which changes in perpetuating cognitions and behaviour are related to a reduction in fatigue? Journal of psychosomatic research **75**(3) (2013) 235–241
13. IBM Corp. Armonk, NY: IBM SPSS Statistics for Windows, Version 19.0. (2010)
14. Vercoulen, J., Swanink, C., Galama, J., Fennis, J., Jongen, P., Hommes, O., Van der Meer, J., Bleijenberg, G.: The persistence of fatigue in chronic fatigue syndrome and multiple sclerosis: development of a model. Journal of psychosomatic research **45**(6) (1998) 507–517
15. Wiborg, J.F., Knoop, H., Frank, L.E., Bleijenberg, G.: Towards an evidence-based treatment model for cognitive behavioral interventions focusing on chronic fatigue syndrome. Journal of psychosomatic research **72**(5) (2012) 399–404

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

CourboSpark: Decision Tree for Time-series on Spark

Christophe Salperwyck[1], Simon Maby[2], Jérôme Cubillé[1], and Matthieu Lagacherie[2]
[1] EDF R&D,
1 avenue du Général de Gaulle, 92140 Clamart, France
[2] OCTO Technology,
50 avenue des Champs-Élysées, 75008 Paris, France

Abstract. With the deployment of smart meters across many countries, data are being collected at a large scale and volume. These data are collected for billing purposes but also to get analytical insights. Our main goal here is to build an understandable model able to explain the electric consumption patterns regarding several features. We chose to use decision tree models as they are easily comprehensible and have already been parallelized with success. In our industrial context, we often have to work on electrical time-series where the target to predict is neither a label (classification) nor a numerical value (regression) but a time-series representing a load curve. Therefore we use a different split criterion to separate time-series: the inertia. We also need a dedicated method for categorical features since the standard implementation would not work for time-series. This method is based on a hierarchical clustering in order to have a good trade-off between the computational complexity and the exploration of the possible bi-partition splits. We demonstrate the performance of our implementation on datasets with different sizes (up to a terabyte).

Keywords: CourboTree, Hadoop, Spark, Decision Tree, Parallelization

1 Introduction

With the deployment of smart meters across many countries, data are being collected at a large scale and volume. Electric meters measure and transmit electric power consumption from every individual household and enterprise at a rate of a measurement from every 24 hours down to 10 minutes to a centralized information system. In France, EDF provides electricity to more than 35 million customers leading to a massive amount of data to process. These data are collected for billing purposes but also to get analytical insights. Our main goal in this paper is to build an understandable model able to explain the electrical consumption patterns regarding several features such as localization or type of contract. We chose to use decision tree models as they are easily comprehensible and have already been parallelized with success in the Hadoop ecosystem.

CourboSpark is part of the X-Data project: `http://www.xdata.fr/`

Decision trees are well known methods in machine learning which are mainly used for classification and regression tasks. In our industrial context, we often have to work on electrical time-series where the target to predict is neither a label (classification) nor a numerical value (regression) but a time-series representing a load curve. Many versions of decision trees were proposed on top of Hadoop. The Spark [3] implementation seems to be the most suitable for our use-case. Therefore we extended the current Spark/MLlib [4] implementation of decision trees so that it can deal with a time-series as a target.

We first present previous work on parallel decision trees and explain why we choose to reuse the MLlib implementation. Then we describe the time-series regression problem. In section 4, the inertia criterion used to separate time-series is presented. Section 5 will focus on the algorithm adaptation for categorical features with a method based on a hierarchical clustering. Finally, we demonstrate the performance of our implementation on datasets with different sizes (up to a terabyte).

2 Previous work on parallel decision trees

Our goal is to port our in-house software, CourboTree [1, 2], into a distributed system so that more data can be processed. Our CourboTree software build decision tree on time-series based on the inertia criterion.

Many implementations of parallel decision trees have been proposed. In this paper we focus on implementations that can run on top of Hadoop. Hadoop clusters offer a very good trade-off between computing power/storage and price. Hadoop is based on horizontal scaling: the computing power/storage is quasilinear with the number of nodes in the cluster. To have more power, more nodes have to be added into the cluster. Table 1 presents different implementations of decision trees on Hadoop. We used the following criteria to compare these implementations:

- partitioning: horizontal means the algorithm will parallelize computations on the lines, vertical on the columns of the dataset;
- engine: the execution engine that will run the parallel algorithm. These engines can also optimize the execution graph of the algorithm (as for Spark);
- target type: categorical for classification, numerical for regression (can be both)
- ensemble: ability to build an ensemble of trees (random forest, bagging of random trees...)
- pruning: does the algorithm prune the tree to avoid over-fitting?
- open-source: is the source code available so that we can easily reuse it?

As we aimed to reuse an open-source implementation and our datasets mainly grow in an horizontal way, we choose to adapt the MLlib implementation. Moreover it uses the Spark engine which is faster than the original Map/Reduce engine in Hadoop.

	Partitionning	Engine	Target type	Ensemble	Pruning	Open-source
MLlib [4]	Horizontal	Spark	Num + Nom	Yes	No	Yes
MR C4.5 [5]	Vertical	MR	Nom	No	No	No
PLANET [6]	Horizontal	MR	Num + Nom	Yes	No	No
SAMOA [7]	Vertical	Storm/S4	Num + Nom	Yes	Yes	Yes

Table 1. Comparison of parallel decision trees in the Hadoop ecosystem (MR: original Map-Reduce, Storm/S4: streaming frameworks, Spark: new computing engine – widely used to replace MR).

3 Problem description: time-series regression

The problem is the same as the one stated in CourboTree [1, 2]: explain load curves pattern using explanatory features. It can be seen as time-series regression. For this problem, we define our dataset as follows:

- $1, ..., n$: the examples of the dataset;
- $w_1, ..., w_n$: the weights of the examples;
- $X_1, ..., X_j, ..., X_p$: the p explanatory features where x_{ij} is the value for the example i for X_j, these features can be either numerical or categorical;
- $Y_1, ..., Y_k, ..., Y_q$: the q numerical variables defining the time-series where y_{ik} is the value for the example i for Y_k. This time-series is the target of the regression.

Therefore an example i is described as the following tuple: $w_i, x_{i1}...x_{ip}, y_{i1}...y_{iq}$. There can be missing values for either the explanatory features or the time-series.

As in CourboTree, the model used to explain the load curves is a tree:

- l: a node/leaf of the tree,
- g_l: the center of gravity of the examples in l. Its coordinates $g_{l1}, \ldots, g_{lq}$ are the weighted mean of the examples in the node/leaf l for the q points of the time-series Y.

4 Split criteria: inertia

Decision trees used in classification aim to lower the impurity in the leaves and use a criterion such as the entropy gain or the Gini coefficient. For regression task, variance reduction is often used. As we deal with time-series we want to lower the variance between the time-series within a leaf/node. In this paper we use the euclidean distance to compute the variance. Other distances could be used.

Given a node t, its intra inertia is defined as:

$$I_w(l) = \sum_{i \in l} w_i \sum_{k=1,...,q} (y_{ik} - g_{lk})^2$$

This criteria can be used in the same way as the criterion used in classification and regression trees. The best split to divide a leaf l into two leaves l_L and l_R is the one minimizing the intra-inertia of these two new leaves:

$$argmin\Big(I_w(l_L) + I_w(l_R)\Big)$$

Computing this intra-inertia is expensive but we can use the König-Huyghens theorem which states that the global inertia is the sum of the intra-inertia I_w and inter-inertia I_B:

$$I = I_w + I_B$$

Therefore we can maximize the inter-inertia instead of minimizing the intra-inertia (which is also known as the "Ward's method" [8] in statistics). This corresponds to find the centers of gravity of the two new leaves which are the furthest possible (relatively to their center weights). The computation of the inter-inertia is much more effective. For each potential split points we do not need to recompute the intra-inertia on all the points to the two new centers of gravity, but just the distance between the two new centers of gravity.

For times-series this means to maximize the distance between the average curves in the two leaves (l_L, l_R) having weights (w_L, w_R):

$$argmax\left(\frac{w_L.w_R}{w_L + w_R}\sum_{i\in 1\ldots q}(g_{Li} - g_{Ri})^2\right)$$

As computing the inter-inertia criterion is more effective we have only used this criterion in CourboSpark. Both criteria are sums that can be parallelized. Therefore the computation can be easily spread across the nodes in the cluster in order to take advantage of its computational power.

5 Categorical variables: hierarchical clustering

For numerical feature, the MLlib algorithm has a parameter to specify the number of "bins" that will be tested as split points. These bins correspond to quantiles computed on a random sample of the whole dataset. For the categorical features the next section explains how the current implementation works and how it was adapted for time-series.

Building a binary decision tree with categorical feature requires to evaluate all the bi-partition/split of the modalities. The bi-partition with the lowest value of the criteria is the one chosen to split the node into two leaves. This evaluation can be costly as for m modalities there are $(2^{m-1} - 1)$ possible bi-partitions. With m up to 30, the computation of all these partitions can be computed in a reasonable time (less than a minute on recent hardware). Our datasets have categorical features up to 100 modalities (the number of French departments for example). An exhaustive test of all the bi-partitions would take too much time as the number of bi-partitions to test is exponential with the number of modalities.

There are many ways to deal with categorical features in a binary decision tree in order to limit the number of potential split points. The simplest one is to only test the bi-partitions having one modality versus all the other modalities. A more sophisticated approach is based on ordering the modalities. For classification, modalities are ordered relatively to their entropy or Gini and for regression relatively to their means. This option is the one chosen in the MLlib decision tree implementation. As we deal with time-series, we can not order modalities. We can illustrate that with 4 stable curves: (a) at 0.6, (b) at 0.4, (c) at -0.45 and (d) at -0.55. Their average curve is at 0.0. If we order them by their euclidean distance to the average curve we have: (b,c,a,d), which gives 3 possible bi-partitions: ((b,(c,a,d)), ((b,c)(a,d)) and ((b,c,a)(d)). Looking at this example, the best bi-partitions is ((a,b)(c,d)) but was not evaluated using this heuristic. Therefore we chose to use the hierarchical clustering [8] to find better splits without doing an exhaustive test of all the bi-partitions.

Hierarchical clustering is a bottom-up heuristic. The first step is to aggregate the curve of the two modalities that maximize the loss of inertia. Using the inter-inertia criterion, this aggregation corresponds to the computation of the new average curve based on the weighting sum of the average curve of the two modalities. These two aggregated modalities are seen as one cluster and then we can continue to merge the modalities/cluster in the same way. Figure 1 shows an example of hierarchical clustering with 7 modalities. The algorithm stops once there are only two clusters. In order to optimize the final bi-partition we perform a post-optimization by trying to move all the modalities from one cluster to the other one. Modalities are moved one by one, until no loss of inertia can be achieved. The complexity of this heuristic is much lower: $O(m^3)$.

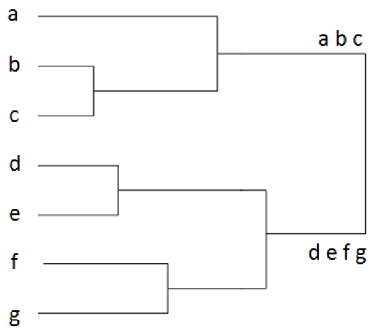


Fig. 1. Example of ascending hierarchical clustering.

6 Experiments

This section presents our experiments on a generated, but realistic, dataset with different sizes.

Dataset: In order to control the experiments we have developed a generator. This generator takes as parameters: the depth of the tree, number of points in the time-series, number of features, number of modalities for categorical features. We configured the generator to have a tree of depth 4, with 10 numerical and 10 categorical features (50 modalities each). Each time-series has 144 points (step of 10 minutes for one day). We generated from 10 to 1,000 millions time-series.

Configuration: The experiments were run on a cluster of 10 machines. Each machine has 32 cores, 48 GB of RAM and 14 SATA spinning disks. Spark was configured to run with 9 executors with 20 GB and 8 cores each.

Results: The results of the the experimentation are presented in Figure 2. As we can see the Spark MLlib implementation of decision tree scales almost linearly with the dataset size.

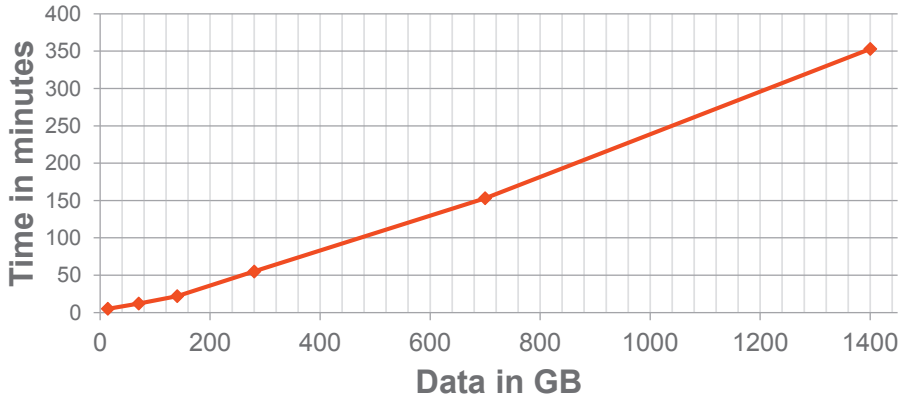


Fig. 2. Time to build the tree depending on the dataset size (10 to 1,000 millions curves).

More experiments were conducted with similar results. We tested datasets with up to 100 features, and also up to 500 modalities for categorical features.

7 Future works

We plan to extend our implementation so that it can deal with "cyclic features" as day of the week, month of the year... which are common in our datasets.

This case is similar to numerical feature but gives two times more possible split points.

The MLlib library only build trees which are balanced (each leaf is at the same depth). The current CourboTree implementation first grows the part of the tree that gives the greatest loss of inertia and therefore can produce unbalanced trees. As we would like to have the lowest global inertia for a given number of leaves, we would either need to drive the tree construction to expand just a part of the tree or to do post-pruning to remove parts of the tree. A next step could be to use this pruning to control over-fitting.

More generally we plan to do more extensive tests to study how the Spark configuration (number of executors, memory...) impact performance depending on the datasets properties (number of features, modalities/bins...).

References

1. Stéphan, V., Cogordan, F.: Courbotree : Application Des Arbres De Regression Multivariés Pour La Classification De Courbes. XXXVIèmes Journées de statistique à Montpellier. http://www.agro-montpellier.fr/sfds/CD/textes/stephan1.pdf
2. Stéphan, V.: CourboTree : Une Méthode de Classification de Courbes Appliquée au Load Profiling pp. 129–138. Revue MODULAB 33 (2005). http://www.modulad.fr/archives/numero-33/stephan-33/stephan-33.pdf
3. Zaharia M., Chowdhury M., Franklin M.J., Shenker S., Stoica I.: Spark: Cluster Computing with Working Sets, Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (2010).
4. Amde M., Das H., Sparks E. Talwalkar A.: Scalable Distributed Decision Trees in Spark MLLib, Spark Summit 2014. http://spark-summit.org/wp-content/uploads/2014/07/Scalable-Distributed-Decision-Trees-in-Spark-Made-Das-Sparks-Talwalkar.pdf
5. Dai W., Wei J.: A MapReduce Implementation of C4.5 Decision Tree Algorithm. International Journal of Database Theory and Application. Vol. 7, No. 1, pp. 49–60 (2014) http://www.chinacloud.cn/upload/2014-03/14031920373451.pdf
6. Panda B., Herbach J., Basu S., Bayardo R.: PLANET: massively parallel learning of tree ensembles with MapReduce. Proceedings of the VLDB Endowment, pp. 1426–143 (2009).
7. De Francisci Morales G. , Bifet A.: SAMOA: Scalable Advanced Massive Online Analysis. Journal of Machine Learning Research, Volume 16, pp. 149–153 (2015).
8. Ward, J. H. : Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association 58 , no. 301 pp. 236–244 (1963).

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Anomaly Detection in Temporal Graph Data: An Iterative Tensor Decomposition and Masking Approach

Anna Sapienza[1,2], André Panisson[2], Joseph Wu[3],
Laetitia Gauvin[2,*], Ciro Cattuto[2]
[1] Polytechnic University of Turin, Turin, Italy
[2] Data Science Laboratory, ISI Foundation, Turin, Italy
[3] School of Public Health, University of Hong Kong, Hong Kong

Abstract. Sensors and Internet-of-Things scenarios promise a wealth of interaction data that can be naturally represented by means of time-varying graphs. This brings forth new challenges for the identification and removal of temporal graph anomalies that entail complex correlations of topological features and activity patterns. Here we present an anomaly detection approach for temporal graph data based on an iterative tensor decomposition and masking procedure. We test this approach using high-resolution social network data from wearable sensors and show that it successfully detects anomalies due to sensor wearing time protocols.

Keywords: Data cleaning, anomaly detection, non-negative tensor factorization, high-resolution social networks, sensors, temporal networks.

1 Introduction

Emerging applications in the big data and Internet-of-Things domains pose new problems for data cleaning. Time-resolved interaction data, in particular, are especially challenging because the relational nature of the data yields anomalies that entangle temporal and topological aspects. Several studies have focused on identifying anomalous behaviors in graph-based datasets [1] and time-varying networks [2]. However, mesoscale anomalies that mimic normal behaviors are observed in empirical data and call for further research.

Here we focus on time-varying graphs [3] represented as three-mode tensors and we present an semi-supervised anomaly detection method based iterative tensor decomposition and masking. We report on the performance of this method in detecting and removing anomalies in an empirical social network dataset gathered by using wearable proximity sensors in a school.

2 Methodology

A static graph can be represented by an adjacency matrix $M \in \mathbb{R}^{N \times N}$, where $M_{ij} = 1$ if a contact between i and j occurred and $M_{ij} = 0$ otherwise. This description can be generalized to the case of a time-varying graph, by using a sequence of S consecutive adjacency matrices, that can be easily arranged as a tensor $\mathcal{T} \in \mathbb{R}^{N \times N \times S}$.

The extraction of latent structures can then be performed by following the iterative approach described below. This framework allows to carry out the data cleaning by unearthing at each iteration group behaviours of nodes having correlated activities and classifying these patterns of activities as meaningul or anomalous.

Step 1. The Non-negative Tensor Factorization [6] is used as a powerful tool to approximate the tensor $\mathcal{T}$ as a sum of R rank-one tensors $\mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$, called components. In the specific case of temporal networks, $\mathbf{a}_r$ and $\mathbf{b}_r$ provide the membership of nodes to the component r, whereas $\mathbf{c}_r$ is the temporal activity pattern of the component. Moreover, it is possible to consider $\mathbf{a}_r \approx \mathbf{b}_r$ if the graph is undirected. The components can be recovered by solving an optimization problem with non-negative constraints. The minimization problem

$$\min \left\| t_{ijk} - \sum_{m=1}^{R} \sum_{n=1}^{R} \sum_{l=1}^{R} a_{im} a_{jn} c_{kl} \right\|_F^2 \quad \text{s.t. } a_{im}, a_{jn}, c_{kl} \geq 0 \tag{1}$$

is computed by the alternating non-negative least squares method [7], solved by using the block principal pivoting algorithm [8]. The selection of a suitable number R of components is guided at each iteration by the Core Consistency Diagnostic [9, 10], and performed in order to prevent overfitting.

Step 2. The extracted components are analysed in order to discriminate between those dominated by anomalous activities or meaningful behaviours. To this end, a classifier working on the temporal activity patterns of each component $\mathbf{c}_r$ was developed.

Step 3. Spurious contact patterns highlighted by the anomalous components are combined into a mask, used to clean the original tensor. The nodes involved in each of these contacts are detected by analysing the level of membership given by $\mathbf{a}_r$. The occurrence times of these contacts are given by the anomalous windows found in the temporal patterns $\mathbf{c}_r$. These windows are recovered by using a step detection algorithm based on the Otsu threshold [11].

Step 4. The mask is applied to the tensor $\mathcal{T}$ in order to erase the invalid entries. The cleaned tensor $\mathcal{T}'$ becomes then the input of the consecutive iteration in the iterative framework.

Step 5. The procedure is repeated until no component is classified as anomalous in step 2.

3 Results and Validation

The current investigation involves the analysis of a high-resolution dataset which describes the interactions of people in a primary school in Hong Kong. The school population consisted in 709 children and 65 teachers divided into 30 classes. Data were collected by using wearable proximity sensors [4, 5] over 10 consecutive days in March 2013, from Monday 18th to Thursday 27th. These sensors record spatial proximity with a resolution of 20s. As a result, a time-varying network with $N = 774$ nodes was created. The data were then aggregated over a time-window of 5min, leading to a division of the overall network in $S = 2680$ snapshots.

The protocol was as follows: the proximity of the sensors was recorded during the whole experiment duration, and the sensors were grouped in each class at the end of the school day. Hence, activity patterns composed by strong steady contacts withinh each class were observed during the school closing time. In order to clean the data, these anomalous patterns must be retrieved. A general methodology is thus developed here to deal with the anomaly detection of temporal graph-based data, and then used to perform the data cleaning of the present problem.

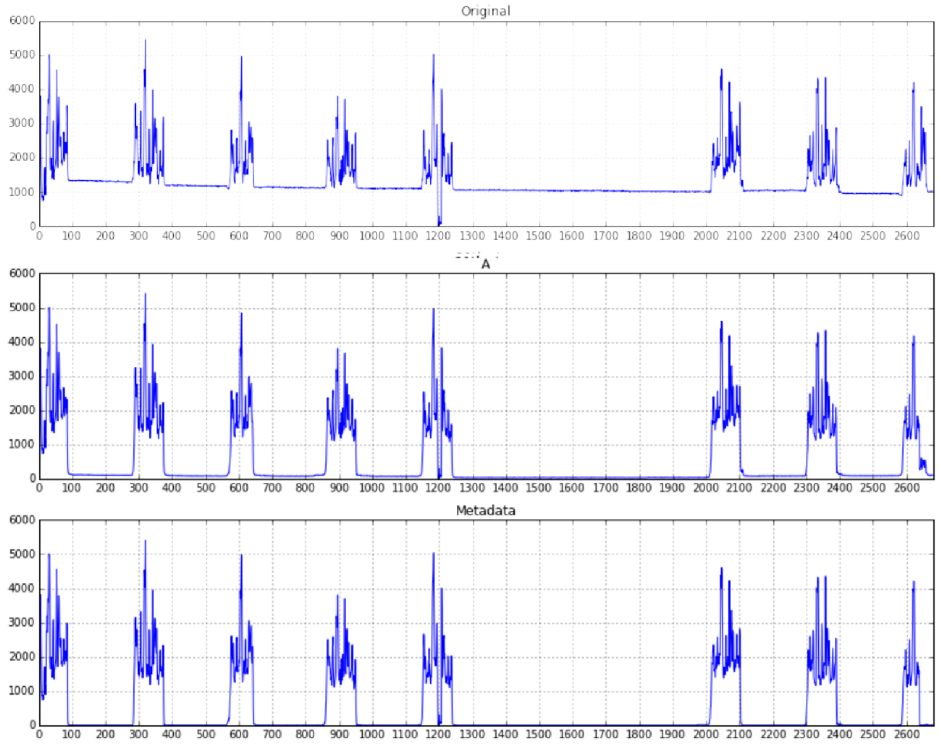


Fig. 1: **Total contact number measured in the original, metadata and cleaned (A) tensors with respect to time.** An obvious amount of contacts in the original state is distributed along the entire time-line. By contrast, the cleaning procedure managed to identify and erase most of the anomalies.

The results after 23 iterations of the iterative framework presented in Section 2 are summarised in Fig. 1, that shows the total contact number evolving in time measured in the original tensor and in the cleaned tensor generated by the iterative process. The total amount of contacts during the school closure is extremely reduced as a result of the cleaning process. Normal interactions belonging to the classes emerge and meaningful patterns are recovered.

In order to validate the method, a reference tensor was created and used as a ground truth. To this end, anomalous behaviours were identified and removed from the original dataset by applying the step detection on the temporal contact

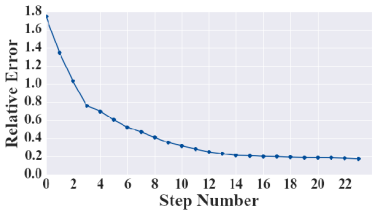


Fig. 2: Relative error, computed at each iteration by using the L_1-norm.

Table 1: Retrieval measures of tensor entry labelling. The entries classified as anomalous or meaningful are respectively marked as 0 or 1.

	Precision	Recall	F1-score
0	0.98	0.92	0.95
1	0.88	0.96	0.92

densities of each class. The ensuing tensor was compared with the cleaned tensor $\mathcal{T}'$ at each iteration, by computing the relative error with the L_1-norm. This quantity, shown in Fig. 2, monotonically decreases with the number of steps of the iterative approach and stabilizes around the low value of 0.2.

The reference tensor was then used to label the entries of the original tensor as anomalous or meaningful, in order to compute the confusion matrix summarizing the performances of the iterative approach. The resulting recall and precision values at each entry level reported in Tab. 1 and the overall accuracy of 0.94 highlight the high performance of the iterative approach.

4 Conclusions

Time-varying graphs can expose both topology and temporal correlations, which make the anomaly detection a major challenge. The iterative approach introduced here captures such correlations and enables to discriminate between meaningful and anomalous patterns. The evaluation measurements of the anomaly detection, achieved on the primary school dataset, highlights the high performance of the implemented method.

This iterative method is a principled approach, which provides an unsupervised way to identify and select meso-scale data anomalies. However, some limitations are worth noting. The method relies on the temporal activity profile of a latent component to be able to classify it as anomalous or not. Moreover, the implication of the NTF makes the iterative approach computationally costly, in terms of memory and time. The latter problem is being tackled as a lot of research is devoted to improve the efficiency of the implementation.

Finally, the iterative framework could be extended to the case of a greater number of dimensions, e.g. with sensors localized in space.

5 Acknowledgements

This research was supported by the Lagrange Project of the ISI Foundation funded by the CRT Foundation, by the Q-ARACNE project funded by the Fondazione Compagnia di San Paolo, by the FET Multiplex Project (EU-FET-317532) funded by the European Commission, and by the Harvard Center for Communicable Disease Dynamics from the National Institute of General Medical Sciences (grant no. U54 GM088558). The funding bodies had no role in study design, data collection and analysis, preparation of the manuscript, or the decision to publish.

References

1. W. Eberle, L. Holder, Discovering structural anomalies in graph-based data. In *ICDM Workshops*, pp. 393-398, 2007.
2. M. Mongiovi, et al., Netspot: Spotting significant anomalous regions on dynamic networks. *SIAM International Conference on Data Mining*, 2013.
3. P. Holme, J. Saramäki. Temporal networks. *Phys. Reps.* 519:97-125, 2012.
4. J. Stehlé, N. Voirin, A. Barrat,C. Cattuto, L. Isella, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PLoS One* 6:e23176, 2011.
5. C. Cattuto, et al. Dynamics of person-to-person interactions from distributed RFID sensor networks. *PLoS ONE* 5:e11596, 2010.
6. T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455-500, 2009.
7. H. Kim, L. Eldé n, and H. Park. Non-negative tensor factorization based on alternating large-scale nonnegativity-constrained least squares. In *Proceedings of IEEE 7th International Conference on Bioinformatics and Bioengineering* (BIBE07), volume II, pages 1147-1151, 2007.
8. J. Kim and H. Park. Fast nonnegative tensor factorization with an active-set-like method. In *High-Performance Scientific Computing*, Springer, London, 2012, pp. 311-326
9. R. Bro and H. A. L. Kiers, A new efficient method for determining the number of components in PARAFAC models, *Journal of Chemometrics*, 17 (2003), pp. 274-286.
10. E.E. Papalexakis and C. Faloutsos, Fast efficient and scalable core consistency diagnostic for the parafac decomposition for big sparse tensors, in *Acoustics, Speech and Signal Processing* (ICASSP), 2015 IEEE International Conference on. IEEE, 2015.
11. M. Fang, G. Yue, and Q. Yu, The Study on An Application of Ostu Method in Canny Operator, *Proceeding of the 2009 International Symposium on Information Processing*, Huangshan, P. R. China, pp. 109-112, August 2009.

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Progressive and Iterative Approaches for Time Series Averaging

Saeid Soheily-Khah, Ahlame Douzal-Chouakria, and Eric Gaussier
Université Grenoble Alpes, CNRS - LIG/AMA, France
{Saeid.Soheily,Ahlame.Douzal,Eric.Gaussier}@image.fr

Abstract. Averaging a set of time series is a major topic for many temporal data mining tasks as summarization, extracting prototype or clustering. Time series averaging should deal with the tricky multiple temporal alignment problem; a still challenging issue in various domains. This work compares the major progressive and iterative averaging time series methods under dynamic time warping (DTW).

1 Introduction

Time series centroid estimation is a major issue for many temporal data analysis and mining tasks as summarization, extracting temporal prototype or clustering. Estimating the centroid of a set of time series under time warp should deal with the tricky multiple temporal alignment problem [1–4]. Temporal warping alignment of time series has been an active research topic in many scientific disciplines. To estimate the centroid of two time series under temporal metrics, as the dynamic time warping [5–7], one standard way is to embed the time series into a new Euclidean space defined by their temporal warping alignment. In this space, the centroid can be estimated as the average of the linked elements. The problem becomes more complex where the number of the time series is more than two, as one needs to determine a multiple alignment that links simultaneously all the time series on their commonly shared similar elements.

A first manner to determine a multiple alignment is to search, by dynamic programming, the optimal path within an n-dimensional grid that crosses the n time series. The complexity of this approach prevents its use, as it constitutes an NP-complete problem with a complexity of $O(T^n)$ that increases exponentially with the number of time series n and the time series length T. A second way, that identifies progressive approaches, is based on combining progressively pairwise time series centroids to estimate the global one. The progressive approaches may suffer of the early error propagation through the set of pairwise centroid combinations. The third approach is iterative, it works similarly to the progressive approach, but mainly reduces the error propagation by repeatedly refining the barycenter and realigning it to the initial time series. In general, the main progressive and iterative approaches are of heuristic nature limited to the dynamic time warping metric, that provide an estimation of the barycenter without guarantee of an optimal solution.

The main contribution of this work is to introduce some major progressive and iterative approaches for time series centroid estimation, prior to present their

characteristics, as well as an extensive comparison between the mentioned methods throughout real and synthetic datasets, where to the best of our knowledge this necessary study is never conducted before.

The rest of this paper is organized as follows: In the next section, different approaches are studied and Section 3 presents the conducted experimentation and discuss the results obtained. Lastly, Section 4 concludes the paper.

2 Progressive and iterative approaches

The progressive and iterative methods for averaging a set of time series are mostly derived from the multiple sequence alignment methods to address the tricky multiple temporal alignment problem. In the following, we review the major progressive and iterative approaches for time series averaging under time warp.

Gupta et al. in [8] used the DTW in the sequence alignment to average a set of time series. The method, called "*NonLinear Alignment and Averaging Filters* (NLAAF)", uses a tournament scheme averaging approach that it's simplest averaging ordering consists in pairwise averaging sequences following a tournament scheme. That way, $N/2$ average sequences are created at first step. Then those $N/2$ sequences, in turn, arepairwise averaged into $N/4$ sequences, and so on, until one sequence is obtained. In this approach, the averaging method between two sequence is applied $(N-1)$ times. NLAAF works by placing each element of the average sequence of two time sequences, as the center of each association created by DTW. Its main drawback lies in growth of its resulting length, because each use of the average method can almost double the length of the average sequence. That is why NLAAF is generally used in conjunction with a process reducing the length of the average, leading to a loss of information and thus to an unsatisfactory approximation. Additionally, the average strongly depends on the order of time series sequences and so, different orders of sequences give different average sequence.

To avoid the bias induced by random selection, Niennattrakul et al. [11, 12] proposed a framework of shape averaging called "*Prioritized Shape Averaging* (PSA)", which uses hierarchical clustering with a new DTW averaging function, labeled "*Scaled Dynamic Time Warping*" with extra capability in stretching some parts of warping path so that the result is more similar to a sequence time series with more weight. Niennattrakul used hierarchical clustering as a heuristic to order the priority. In spite of this hierarchical averaging method aims to prevent the order dependency, the length of average sequences remains a problem. Local averaging strategies like NLAAF or PSA may let an initial approximation error propagate throughout the averaging process. If the averaging process has to be repeated, the effects may dramatically alter the quality of the result. This is why a global approach is desirable, where sequences would be averaged all together, with no sensitivity to their order of consideration.

A direct manner to estimate the centroid proposed by Abdulla et al. [1], called "*Cross-Words Reference Template* (CWRT)", which uses medoid as the

reference time series as follows. First, the time series medoid is selected. The whole time series are then described in the representation space defined by the reference medoid. In the next step, all sequences are aligned by DTW to a single medoid and then the average is computed by averaging the time-aligned time series across each point. Petitjean et al. [3] proposed a global averaging method, called "*Dtw Barycenter Averaging* (DBA)", which consists in iteratively refining an initially average sequence, in order to minimize its distance to the averaged sequence. As a summary, the DBA under temporal warping is a global approach that can average a set of sequences all together.

All the methods define heuristic approaches, although with no guarantee of optimal solutions, the provided approximations are accurate particularly for time series that behave similarly within the set. However these approaches may fail principally for time series with similar global behavior and local temporal differences, as one needs to deploy local instead of global averaging process.

3 Experimental study

The experiments are conducted to compare the above approaches on classes of time series composing various datasets. The datasets can be divided into two categories. The first one is composed of time series that have similar global behavior within the classes, where the time series of the second category may have distinct global behavior, while sharing local characteristics [9]. For the comparison, the induced inertia reduction rate and the required run time are evaluated as well as the qualitative comparison of the centroids obtained by a visualization. In the following, we first describe the datasets used, then specify the validation process and discuss the obtained results.

3.1 Data descrpition

The experiments are first carried out on four well known public datasets CBF, CC, DIGITS and CHARACTER TRAJ. [10]. These data define a favorable case for the averaging task as time series behave similarly within the classes. Then, we consider more complex datasets: BME[1], UMD[1], SPIRAL [4], NOISED SPIRAL[1] and CONSSEASON [10]. They are composed of time series that behave differently within the same classes while sharing several local characteristics. Table 1 indicates for each data set: the number of classes it includes (Nb. Class), the number of instances (Nb. TS), the number of attributes (Nb. Att), the time series length (TS length) and the global or local nature of similarity within the classes (Type).

3.2 Validation process

The four mentioned methods NLAAF, PSA, CWRT and DBA described in Section 2 is compared together. The performances of these approaches are evaluated through the centroid estimation of each class of the above described datasets.

[1] http://ama.liglab.fr/∼douzal/data

Table 1: Data description

DATASET	NB. CLASS	NB. TS.	NB. ATT. TS.	LENGTH	TYPE
CBF	3	930	1	128	GLOBAL
CC	6	600	1	60	
DIGITS	10	220	2	85	
CHAR. TRAJ.	20	200	3	20	
BME	3	150	1	90	LOCAL
UMD	3	150	1	121	
SPIRAL	1	50	3	95	
NOISED SPIRAL	1	50	3	300	
CONSSEASON	2	365	1	144	

Particularly, the efficiency of each approach is measured through: a) the reduction rate of the inertia criterion; the initial inertia being evaluated around the time series medoid that minimizes the distances to the rest of time series and b) the space and time complexity. The results reported hereafter are averaged through a bootstrap process, with 10 repetitions. Finally for all reported results, the best one which is significantly different from the rest(t-test at 1% risk) is indicated in bold.

Inertia reduction rate Time series averaging approaches are used to estimate centroid of the time series classes described above, then the inertia w.r.t. the centroids is measured. Lower is the inertia higher representative is the extracted centroid. Table 2, gives the obtained inertia reduction rates IRR=$1-\frac{\sum_{i=1}^{N} D(x_i,c)}{\sum_{i=1}^{N} D(x_i,m)}$, averaged per dataset; $x_1, ..., x_N$ being the set of time series, D the metric, $\mathbf{c}$ the determined centroid and m the initial medoid. Table 2 shows that the DBA provides the highest IRR for the most datasets. Some negative rates observed indicate an inertia increase.

Table 2: Comparison of Inertia Reduction Rate(IRR)

DATASET	NLAAF	PSA	CWRT	DBA
CBF	8.3%	12.3%	-61.3%	**32.1**%
CC	9.8%	28.6%	6.8%	**34.2**%
Digits	26.1%	79.5%	77.6%	**82.2**%
CHAR. TRAJ.	67.1%	87.7%	85.2%	**90.6**%
BME	34.9%	43.1%	-11.8%	**59.4**%
UMD	25.6%	**51.1**%	-56.2%	48.8%
SPIRAL	59.8%	64.4%	64.2%	**65.8**%
NOISED SPIRAL	61.4%	**66.3**%	9.3%	9.8%
CONSSEASON	**84.1**%	70.5%	4.6%	21.4%

Time and space complexity In Table 3 the studied approaches are compared w.r.t their space and time complexity. The results, averaged per dataset, reveal almost DBA the faster method and PSA the slowest one. The CWRT approach is not comparable to the rest of the methods as it performs directly an euclidean distance on the time series once the initial DTW matrix evaluated. Remark that for NLAAF and PSA the centroid lengths are very large making these approaches unusable for large time series. The centroid lengths for the remaining methods are equal to the length of the initial medoid. The higher time consumptions observed for NLAAF and PSA are mainly explained by the progressive increase of the centroid length during the pairwise combination process.

Table 3: Comparison of Time/Space complexity

	NLAAF		PSA		DBA	
DATASET	LENGTH	TIME	LENGTH	TIME	LENGTH	TIME(NB-IT.)
CBF	8283	392.32	35042	9999.99	128	**42.91**(30)
CC	992	**4.15**	1677	12.75	60	6.46(40)
DIGITS	313	0.52	530	1.09	85	**0.51**(15)
CHAR. TRAJ.	33	0.06	29	0.06	20	**0.03**(10)
BME	2027	5.46	2781	11.92	90	**3.93**(30)
UMD	2729	10.32	4280	28.87	121	**4.75**(30)
SPIRAL	660	1.62	1122	3.33	95	**1.19**(10)
NOISED SPIRAL	1699	**16.13**	9030	269.93	300	34.84(25)
CONSSEASON	5741	77.10	32706	3680.81	144	**29.79**(35)

3.3 Discussion

From Table 2, we can see that DBA and PSA lead to the highest inertia reduction rates, where the best scores (indicated in bold) are reached by DBA for almost all datasets. However it is significantly lower for some challenging datasets. Finally, CWRT has the lowest inertia reduction rates. The negative rates observed for CWRT indicate an inertia increase. As expected, the DBA method that iteratively optimizes an inertia criterion, in general, reaches higher values than the non-iterative methods (NLAAF, PSA and CWRT).

From Table 3, the results reveal DBA the fastest method and the PSA the slowest one. For NLAAF and PSA the estimated centroids have a drastically large dimension (i.e. a length around 10^4) making these approaches unusable for large time series datasets. The NLAAF and PSA methods are highly time-consuming, largely because of the progressive increase of the centroid length during the pairwise combination process. The centroid lengths for the remaining methods are equal to the length of the initial medoid (Table 3). Finally, PSA appears greatly slower than NLAAF; this is due to the hierarchical clustering on the whole time series. We finally visualize here some of the centroids obtained by the different methods to compare their shape to the one of the time series they represent.

Figure (1) and (2) display the centroids obtained by the mentioned methods respectively for the class "*funnel*" of CBF and "*cyclic*" of data set CC. As one can note, for global datasets, almost all the approaches succeed in obtainging centroids more or less similar to the initial time series. However, we observe generally less representative centroids for NLAAF and PSA.

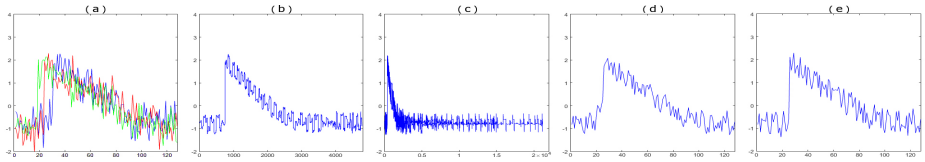


Fig. 1: CBF-"funnel" centroids: (a) ground through, (b) NLAAF, (c) PSA, (d) CWRT, (e) DBA

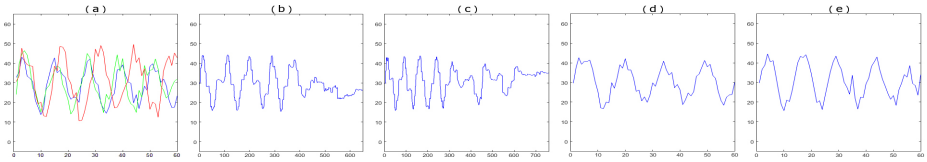


Fig. 2: CC-"cyclic" centroids: (a) ground through, (b) NLAAF, (c) PSA, (d) CWRT, (e) DBA

4 Conclusion

The DTW is among the most frequently used metrics for time series in several domains as signal processing, temporal data analysis and mining or machine learning. However, for time series clustering, approaches are generally limited to Kmedoid to circumvent time series averaging under DTW and tricky multiple temporal alignments problem. The present study compares the major progressive and iterative time series averaging approaches under dynamic time warping. The experimental validation is based on global datasets in which time series share similar behaviors within classes, as well as on more complex datasets exhibiting time series that share only local characteristics, that are multidimensional and noisy. Both the quantitative evaluation, based on an inertia criterion and time and space complexity, and the qualitative one (consisting in the visualization of the centroids obtained by different methods) show the effectiveness of DBA approach. In particular, the DBA method that iteratively optimizes an inertia criterion, not only, reaches higher values than the non-iterative methods (NLAAF, PSA and CWRT), but also provides a fast time series averaging for global and local datasets.

References

1. Abdulla, W.H. and Chow, D. and Sin, G.: Cross-words reference template for DTW-based speech recognition systems. Proc. TENCON, Pages 1576–1579, Vol. 2 (2003)
2. Hautamaki, V. and Nykanen, P. and Franti, P.: Time-series clustering by approximate prototypes. 19th International Conference on Pattern Recognition, (2008).

3. Petitjean, F. and Ketterlin, A. and GanÇarski, P.: A global averaging method for dynamic time warping, with applications to clustering. Pattern Recognition, Pages 678-693, Vol. 44 (2011)
4. F. Zhou and F. De la Torre: Generalized time warping for multi-modal alignment of human motion. IEEE, Computer Vision and Pattern Recognition (CVPR), Pages 1282–1289 (2012)
5. Kruskall, J.B. and Liberman, M.: The symmetric time warping algorithm: From continuous to discrete. Addison-Wesley, Time Warps Journal (1983)
6. Sakoe, H. and Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, Pages 43–49, Vol. 26 (1978)
7. Sankoff, D. and Kruskal, J.B.: Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. Cambridge University Press, Addison-Wesley, (1983)
8. Lalit Gupta, D. Molfese, R. Tammana, P. Simos: Nonlinear alignment and averaging for estimating the evoked potential. IEEE T. on Biomedical Engineering, No. 4, Pages 348–356, Vol. 43 (1996)
9. C. Frambourg, A. Douzal-Chouakria and E. Gaussier: Learning Multiple Temporal Matching for Time Series Classification. In Advances in Intelligent Data Analysis XII (pp. 198-209). Springer Berlin Heidelberg. (2013)
10. UCI Machine Learning Repository, "*http://archive.ics.uci.edu/ml/*"
11. V. Niennattrakul, C. Ratanamahatana: On Clustering Multimedia Time Series Data Using K-means and Dynamic Time Warping. Multimedia and Ubiquitous Engineering, MUE'07. International Conference on IEEE, Pages 733–738, (2007)
12. V. Niennattrakul, C. Ratanamahatana: Shape Averaging under Time Warping. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 6th International Conference on IEEE, Vol. 2, Pages 626–629, May (2009)

Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Classification Factored Gated Restricted Boltzmann Machine

Ivan Sorokin
ITMO University, Department of Secure Information Technology,
9 Lomonosova str., St. Petersburg, 191002, Russia
i.sorokin@cit.ifmo.ru

Abstract. Factored gated restricted Boltzmann machine is a generative model, which capable to extract the transformation from an image pair. We extend this model by adding discriminative component, which allows directly use this model as a classifier, instead of using the hidden unit responses as features for another learning algorithm. To evaluate the capabilities of this model, we have created a synthetically transformed image pairs and demonstrated that the model is able to determine the velocity of object presented on two consecutive images.

Keywords: Multiplicative interaction, temporal coherence, translational motion, gated Boltzmann machine, supervision learning

1 Introduction

The gated Boltzmann machine is one of the models that uses *multiplicative interactions* [8] for learning the representation, which can be useful to extract the transformation between pairs of *temporally coherent* video frames [12]. Factorized version of this model is presented in [9], where authors train the model on shifts of random dot images and demonstrate that the model is able to identify the different directions correctly. We continue this research by studying the possibility to predict not only directions, but also a shift value. From all types of motion, we chose only translational motion, because it gives a great opportunity to use this model in many vision tasks, such as object tracking or visual odometry [4]. Therefore, the main objective of this work is to create a model that is trained to identify velocity vector in the image coordinate.

Instead of using additional model on top of the *mapping units*, we are adding discriminative component directly to the model. This technique was first applied for restricted Boltzmann machine [6] and since that has become widely used for similar models [11, 10]. In this paper, we are focused on the model that extracts transformation from two consecutive images. Without considering the additional discriminative component, there are several approaches of three-way structure model training [9, 13]. We propose a simple learning algorithm and show that it is not inferior to the existing. Moreover, our learning algorithm takes into account additional label variables and we demonstrate how it effects the training discriminative features. We refer to our model variants as *classification factored gated restricted Boltzmann machine* (cfgRBM).

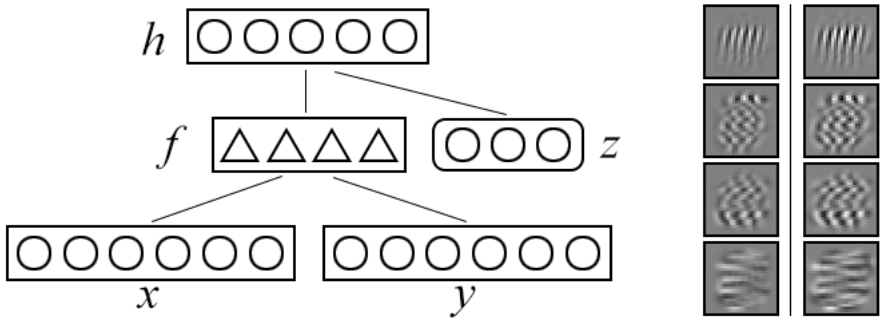


Fig. 1. Left shows the schematic representation of the cfgRBM model. Factorized form of the multiplicative interactions between two visible x, y and hidden h vectors depicted by triangles. The discriminative component is presented as one-hot encoded label vector z. Right shows specially chosen four filter pairs learned on horizontally shifted handwritten digits.

2 The Model

We propose a model (Fig. 1) in which the hidden units h not only captures the relationship between two images x and y, but also interacts with associated label z. The model is defined in the terms of its energy function and the function consists of two basic parts. The first of these is the factored three-way Boltzmann machine [13] and the second is classification restricted Boltzmann machine [5]. Combining these two models we defined expression for the energy function as follows:

$$E(x,y,z,h) = -\sum_f (\sum_i W^x_{if} x_i)(\sum_j W^y_{jf} y_j)(\sum_k W^h_{kf} h_k) - \sum_{kl} h_k V_{kl} z_l \\ - \sum_i a_i x_i - \sum_j b_j y_j - \sum_k c_k h_k - \sum_l d_l z_l \ , \quad (1)$$

where matrices W^x, W^y, W^h has size $I \times F, J \times F$ and $K \times F$ respectively, I and J are equal size of visible units, F - number of factors, K - number of hidden units. The discriminative component is weight matrix V with size $K \times L$ and one-hot encoded label vector z with L classes. Bias terms a, b, c and d associated with two visible, hidden and label vectors respectively. We will assume that the visible vectors are binary, but the model can be defined with real-valued units [1]. Every column $W^x_{\cdot f}$ and $W^y_{\cdot f}$ can be consider as filter pairs (Fig. 1).

To train the model, we also need to define the joint probability distribution over three vectors:

$$p(x,y,z) = \frac{\sum_h exp(-E(x,y,z,h))}{\sum_{x,y,z,h} exp(-E(x,y,z,h))} \ , \quad (2)$$

where the numerator is summing over all possible hidden vectors and denominator is partition function which cannot be efficiently approximated.

2.1 Inference

The inference task of proposed model is defined as the problem of classifying the motion between two related images. In order to choose the most probable label under this model, we must compute conditional distribution $p(z|x, y)$. We have adapted the calculations from the case of single input units [5] for the case of three-way interaction. As a result, for reasonable numbers of labels L, this conditional distribution can be also computed exactly and efficiently, by writing it as follows:

$$p(z_l = 1 \mid \boldsymbol{x}, \boldsymbol{y}) = \frac{exp(d_l) \prod_k (1 + exp(o_{kl}(\boldsymbol{x}, \boldsymbol{y})))}{\sum_{l^*} exp(d_{l^*}) \prod_k (1 + exp(o_{kl^*}(\boldsymbol{x}, \boldsymbol{y})))} \ , \tag{3}$$

where

$$o_{kl}(\boldsymbol{x}, \boldsymbol{y}) = c_k + V_{kl} + \sum_f W^h_{kf} (W^{x\top}_{\cdot f} \boldsymbol{x})(W^{y\top}_{\cdot f} \boldsymbol{y}) \tag{4}$$

is an input to k hidden unit received from images x, y and estimated label l.

2.2 Learning

In order to train a cfgRBM to solve a classification problem, we need to learn the model parameters $\Theta = (W^x, W^y, W^h, V, a, b, c, d)$. Given a training set $\mathcal{D}_{train} = \{(x^\alpha, y^\alpha, z^\alpha)\}$ and a predefined joint distribution (2) between three variables, the model can be trained by minimizing the negative log-likelihood:

$$\mathcal{L}_{gen}(\mathcal{D}_{train}) = - \sum_{a=1}^{|\mathcal{D}_{train}|} \log p(x^\alpha, y^\alpha, z^\alpha) \ . \tag{5}$$

In order to minimize this function the gradient for any cfgRBM parameters $\theta \in \Theta$ can be written as follows:

$$-\mathrm{E}_{h|x^\alpha, y^\alpha, z^\alpha} \left[\frac{\partial E(x^\alpha, y^\alpha, z^\alpha, h)}{\partial \theta} \right] + \mathrm{E}_{x,y,z,h} \left[\frac{\partial E(x, y, z, h)}{\partial \theta} \right] \ , \tag{6}$$

where subscript of the expectation denotes the distribution for variables. There exists a learning rule [2], called "Contrastive Divergence", which can be used to approximate this gradient. Taking this rule into consideration we proposed the Algorithm 1 for the training of cfgRBM model. The main difference from the other approaches for training three-way interaction is in symmetrically sample vectors $\boldsymbol{x}, \boldsymbol{y}$ in the negative phase. Detailed information about the partial derivatives with respect to the model parameters can be obtained from [9, 5].

In case of factored three-way interactions the calculation of the gradient (6) involves numerical instabilities. Especially when using a large input vectors. To avoid this we also use a norm constraint on columns of matrics W^x and W^y. It is a common approach to stabilizing learning. For example, the same recommendations are given by [3] for method "Adaptive Subspace Self-Organizing Map" to learn invariant properties of moving input patterns.

Algorithm 1 Symmetric training update of the cfgRBM model

Require: training triplet $(x^\alpha, y^\alpha, z^\alpha)$ and learning rate λ

Notation
$a \leftarrow b$ means a is set to value b
$a \sim p$ means a is sampled from p

Positive phase
$x^0 \leftarrow x^\alpha$, $y^0 \leftarrow y^\alpha$, $z^0 \leftarrow z^\alpha$
$h_k^0 \leftarrow sigm(o_{kl^0}(x^0, y^0))$

Sample
$\hat{h} \sim p(h|x^0, y^0, z^0)$

Negative phase
$\boldsymbol{x^1} \sim p(x|y^0, \hat{h})$, $\boldsymbol{y^1} \sim p(y|x^0, \hat{h})$, $\boldsymbol{z^1} \sim p(z|\hat{h})$
$h_k^1 \leftarrow sigm(o_{kl^1}(\boldsymbol{x^1}, \boldsymbol{y^1}))$

# Update
for $\theta \in \Theta$ **do**
$\theta \leftarrow \theta - \lambda \left(\frac{\partial E(x^0, y^0, z^0, h^0)}{\partial \theta} - \frac{\partial E(x^1, y^1, z^1, h^1)}{\partial \theta} \right)$
end for

3 Experiments

The main goal of this research is to build a model that is capable to extract translational motion from two related images. Therefore, we created a synthetic data consisting of image pairs in which the second image is horizontally and relatively shifted towards the first. We take MNIST dataset[1] and randomly choose a shift value in the range [-3 3] for each image. As a result we get 7 possible labels for $60,000$ training and $10,000$ test image pairs of relatively shifted handwritten digits. All the models in the following experiments have 200 factors and 100 hidden units. For detailed information about learning parameters we refer to our implementation[2] of the models.

In the first experiment (Fig. 2), we compare different learning strategies for the cfgRBM model. The first learning method is taken from [9], where authors described a conditional model. The second method is proposed in [13], where authors define the joint distribution for an image pair. The results show that Algorithm 1 in the end of learning has the lowest classification and reconstruction test error. It is also interesting to note that there are different delays before filters become specialized in their frequency and phase-shift characteristics.

In the second experiment (Fig. 3), we compare hidden units activities of models with and without a discriminative component. In the first case we trained a model completely unsupervised without any labeled information. In the second case cfgRBM model was trained using Algorithm 1. The results show that

[1] `http://yann.lecun.com/exdb/mnist/`
[2] `https://cit.ifmo.ru/~sorokin/cfgRBM/`

descriminative component has a strong effect on hidden features. In addition, we also demonstrate an effect on the hidden units in the case with wrong label information.

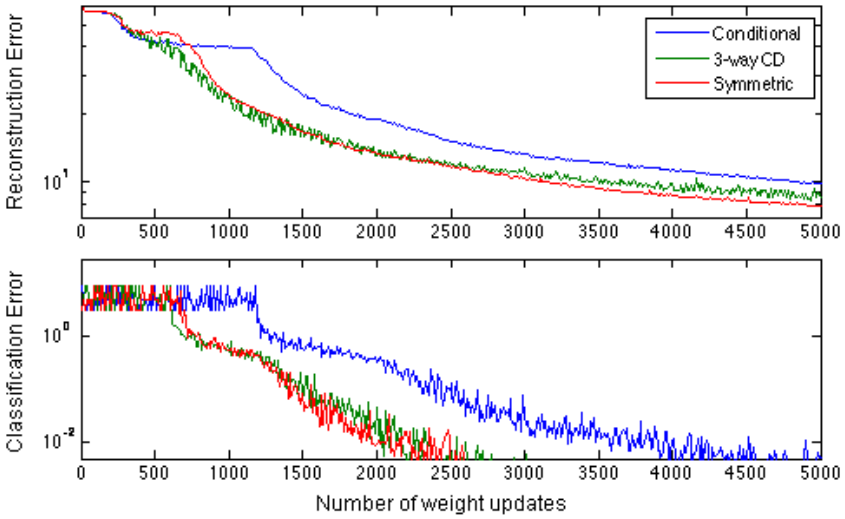


Fig. 2. Three learning stratagies for cfgRBM model. The reconstruction was calculated only for y visible units. In both graphs the error value obtained on test set and the ordinate is scaled logarithmically. Best view in color.

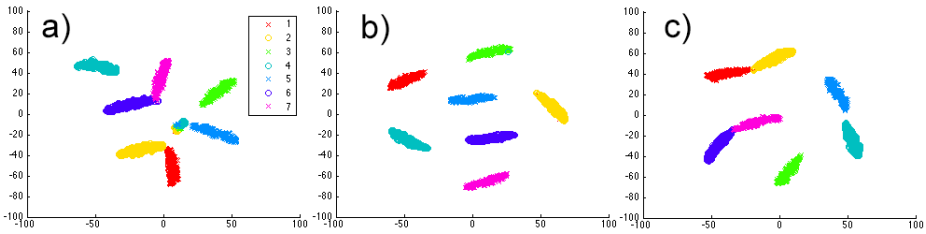


Fig. 3. Hidden units activations. For every test sample, activation of 100 hidden units projected to 2D coordinates using t-SNE [7]. a) model trained without discriminative component. b) model extendet with additional labeled units. c) exactly the same model as in the case (b), but labels of classes {-3,-2} and {2,3} are deliberetly combined.

4 Conclusion

In this paper, we incorporate supervision learning for factored gated restricted Boltzmann machine model. Our results show that proposed model is capable to identify the velocity of the object presented on two consecutive images. In the future work we plan to apply this model for videos which may be represented as a temporally ordered sequence of images. Particularly, the ability to extract translational motion will be useful for tracking tasks.

References

1. Fischer, A., Igel, C.: Training restricted Boltzmann machines: an introduction. Pattern Recognition 47(1), pp. 25–39 (2014)
2. Hinton, G.: Training products of experts by minimizing contrastive divergence. Neural computation 14, pp. 1771-1800 (2002)
3. Kohonen, T.: The adaptive-subspace som (assom) and its use for the implementation of invariant feature detection. In: Proc. ICANN95, Int. Conf. on Artificial Neural Networks, pp. 3-10 (1995)
4. Konda, K., Memisevic, R.: Learning visual odometry with a convolutional network. International Conference on Computer Vision Theory and Applications. (2015)
5. Larochelle, H., Mandel, M., Pascanu, R., Bengio, Y.: Learning algorithms for the classification restricted boltzmann machine. Journal of Machine Learning Research 13, pp. 643-669 (2012)
6. Larochelle, H., Bengio, Y.: Classification using discriminative restricted Boltzmann machines. In: Proceedings of the 25th international conference on Machine learning, pp. 536–543 (2008)
7. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research 9, pp. 2579–2605 (2008)
8. Memisevic, R.: Learning to relate images. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, pp. 1829-1846 (2013)
9. Memisevic, R., Hinton, G.E.: Learning to represent spatial transformations with factored higher-order boltzmann machines. Neural Computation 22, pp. 1473-1492 (2010)
10. Reed, S., Sohn, K., Zhang, Y., Lee, H.: Learning to disentangle factors of variation with manifold interaction. In: Proceedings of the 31st International Conference on Machine Learning, pp. 1431-1439 (2014)
11. Sohn, K., Zhou, G., Lee, C., Lee, H.: Learning and selecting features jointly with point-wise gated boltzmann machines. In: Proceedings of The 30th International Conference on Machine Learning, pp. 217-225 (2013)
12. Srivastava, N.: Unsupervised Learning of Visual Representations using Videos. Department of Computer Science, University of Toronto. Technical Report. (2015) Retrived from `http://www.cs.toronto.edu/~nitish/depth_oral.pdf`
13. Susskind, J., Memisevic, R., Hinton, G., Pollefeys, M.: Modeling the joint density of two images under a variety of transformations. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2793-2800 (2011)

Index of Authors

Amsaleg, L. p. 71
Andrade-Pacheco, R. p. 3
Bailly, A. p. 11
Billard, L. p. 19
Bondu, A. p. 27
Boullé, M. p. 27
Camps-Valls, G. p. 1
Cattuto, C. p. 117
Chapel, L p. 11
Cornuéjols, A. p. 27
Cubillé, J. p. 109
Do, C.-T. p. 39
Douzal-Chouakria, A. p. 19, 39, 123
Dupont, M. p. 47
Gallicchio, C. p. 57
Gaussier, E. p. 123
Gauvin, L. p. 117
Gibberd, A.J. p. 63
Gravier, G. p. 71
Groot, P. p. 101
Guyet, T. p. 11
Hardy, C. p. 71
Heins, M. p. 101
Heskes, T. p. 101
Jain, B.J. p. 79
Knoop, H. p. 101
Krempl, G. p. 85
Lafuente-Rego, B.R. p. 93
Lagacherie, M. p. 109
Lawrence, N. p. 3
Maby, S. p. 109
Malinowski, S. p. 11, 71
Marie, S. p. 39
Marteau, P.-F. p. 47
Micheli, A. p. 57
Mubangizi, M. p. 3
Nelson, J.D.B. p. 63
Panisson, A. p. 117
Parodi, O. p. 57
Pedrelli, L. p. 57
Quiniou, R. p. 71
Quinn, J. p. 3
Rahmadi, R. p. 101
Rombaut, M. p. 39
Salperwyck, C. p. 109
Samadi, S.Y. p. 19
Sapienza, A. p. 117
Soheily-Khah, S. p. 123
Sorokin, I. p. 131
Spiegel, S. p. 79
Tavenard, R. p. 11
Vilar, J.A. p. 93
Vozzi, F. p. 57
Wu, J. p. 117