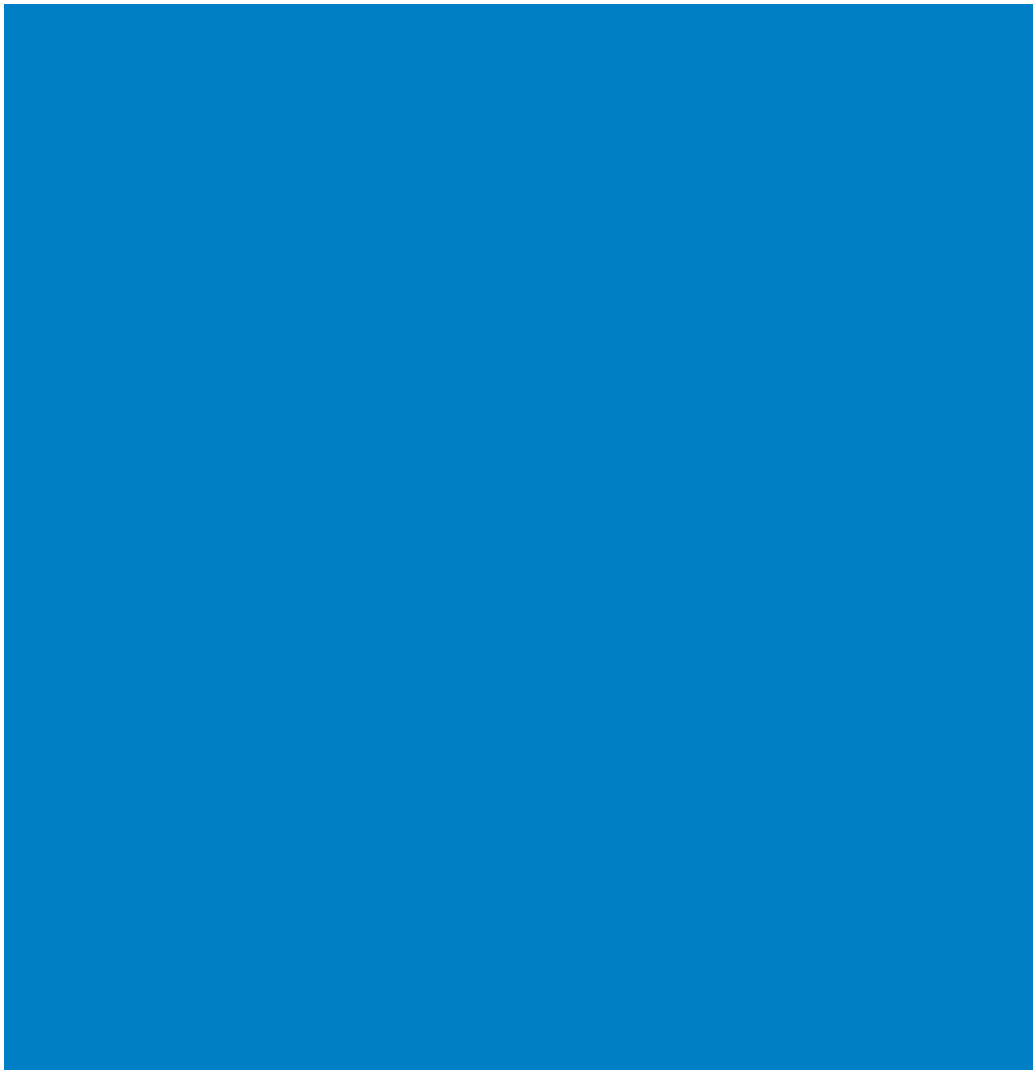
Proceedings of the ECMLPKDD 2015 Doctoral Consortium

Jaakko Hollmén, Panagiotis Papapetrou (editors)



Aalto University

SCIENCE + TECHNOLOGY

CONFERENCE PROCEEDINGS

Aalto University publication series
SCIENCE + **TECHNOLOGY** 12/2015

Proceedings of the ECMLPKDD 2015 Doctoral Consortium

Jaakko Hollmén, Panagiotis Papapetrou (editors)

Aalto University
School of Science

Aalto University publication series
SCIENCE + **TECHNOLOGY** 12/2015

ISBN 978-952-60-6443-7 (pdf)
ISSN-L 1799-4896
ISSN 1799-4896 (printed)
ISSN 1799-490X (pdf)
http://urn.fi/URN:ISBN:978-952-60-6443-7

Unigrafia Oy
Helsinki 2015

Finland

Preface

We are proud to present the Proceedings of the ECMLPKDD 2015 Doctoral Consortium, which was organized during the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2015) in Porto, Portugal during September 7-11, 2015. The objective of the ECMLPKDD 2015 Doctoral Consortium is to provide an environment for students to exchange their ideas and experiences with peers in an interactive atmosphere and to get constructive feedback from senior researchers in machine learning, data mining, and related areas.

Call for Papers was published and distributed widely to the machine learning and data mining community. The community responded enthusiastically, we received altogether 30 submissions. Each paper was read and evaluated by three members of the Program Committee. Based on the reviewer comments, the decisions were made by the chairs of the Program Committee. We decided to accept 27 contributed papers to be included in the program of the doctoral consortium. The program consisted of 2 invited talks, 6 contributed talks, and 21 poster presentations.

We thank our invited invited speakers Sašo Džeroski from Jožef Stefan Institute in Ljubljana, Slovenia and Jefrey Lijffitj from University of Bristol, UK, for their insightful talks. Jefrey Lijffitj's talk titled *So what? A guide on acing your PhD* viewed the PhD journey from a recent graduate's point of view. Sašo Džeroski's presentation titled *The art of science: Keep it simple; Make connections* highlighted success criteria behind science from a more senior point of view. The abstracts of the talks as well as the biographies of our invited speakers are included in the proceedings.

Organizing the ECMLPKDD 2015 Doctoral Consortium has been a true team effort. We wish to thank the ECMLPKDD 2015 Organization Committee for their support and their efforts to distribute the Call for Papers.

In particular, we wish to thank ECMLPKDD 2015 Conference Chairs João Gama and Alípio Jorge. We also thank the members of the Program Committee for their effort to provide insightful and constructive feedback to the authors. Last, but not least, we thank the previous edition's organizers Radim Belohlavek and Bruno Crémilleux for their advice and expertise on the organization of the doctoral consortium.

Helsinki and Stockholm, October 8, 2015,

Jaakko Hollmén and Panagiotis Papapetrou

Organization

Program Committee Chairs

Jaakko Hollmén, Aalto University, Finland
Panagiotis Papapetrou, Stockholm University, Sweden

Program Committee Members

Vassilis Athitsos, University of Texas at Arlington, USA
Isak Karlsson, Stockholm University, Sweden
Eamonn Keogh, University of California, Riverside, USA
Alexios Kotsifakos, Microsoft, USA
Jefrey Lijffijt, University of Bristol, UK
Jesse Read, Aalto University, Finland
Senjuti Roya, University of Washington, USA
Indrė Žliobaitė, Aalto University, Finland

Advisory Chairs

Radim Belohlavek, Palacky University, Czech Republic
Bruno Crémilleux, University of Caen, France

Contents

The art of science: Keep it simple; Make connections

Sašo Džeroski

Jožef Stefan Institute, Ljubljana, Slovenia

Abstract

In my research career, I have spent a lot of time trying to identify classes of models that are of high generality and practical relevance, yet are simple enough to be learned in a computationally tractable way. I have done this in the context of different machine learning tasks. I have also been stealing ideas from some subfields of machine learning and applying them in other. In the talk, I will describe a few examples of these two successful strategies.

Biography of Sašo Džeroski

Sašo Džeroski is a scientific councillor at the Jozef Stefan Institute and the Centre of Excellence for Integrated Approaches in Chemistry and Biology of Proteins, both in Ljubljana, Slovenia. He is also a full professor at the Jozef Stefan International Postgraduate School. His research is mainly in the area of machine learning and data mining (including structured output prediction and automated modeling of dynamic systems) and their applications (mainly in environmental sciences, incl. ecology, and life sciences, incl. systems biology). He has organized many scientific events, most recently two workshops on Machine Learning in Systems Biology and the International Conference on Discovery Science.

He is co-author/co-editor of more than ten books/volumes, including Inductive Logic Programming, Relational Data Mining, Learning Language in Logic, Computational Discovery of Scientific Knowledge and Inductive Databases & Constraint-Based Data Mining. He has participated in many international research projects (mostly EU-funded) and coordinated two of them in the past: He is currently the coordinator of the FET XTrack project MAESTRA (Learning from Massive, Incompletely annotated, and Structured Data) and one of the principal investigators in the FET Flagship Human Brain Project.

So what? A guide on acing your PhD

Jefrey Lijffijt

University of Bristol, UK

Abstract

A PhD is always a challenge. You will be put to the test in a variety of ways; knowledge, skills, inventiveness, perseverance, etc. There are many guides pointing out the obvious competencies that you have to acquire in graduate school in order to successfully complete your PhD. In this talk, I will try to cover less obvious aspects of the most essential problems: how to motivate yourself, how to be productive, how to get published, and how to make your PhD research meaningful.

Biography of Jefrey Lijffijt

Jefrey Lijffijt is a Research Associate in Data Science at the University of Bristol. He obtained his D.Sc. (Tech.) diploma in Information and Computer Science, graded with distinction, in December 2013 from Aalto University, Finland. His thesis received the Best Doctoral Thesis of 2013 award from the Aalto University School of Science. He obtained a BSc and MSc degree in Computer Science at Utrecht University in 2006 and 2008 respectively. He has worked as a research intern at Philips Research, Eindhoven, and as a consultant in predictive analytics at Crystalloids, Amsterdam. His research interests include (visual interactive) mining of interesting/surprising patterns in transactional, sequential, and rela-

tional data, including graphs, as well as text mining, natural language processing, statistical significance testing, and maximum entropy modelling.

Detecting Contextual Anomalies from Time-Changing Sensor Data Streams

Abdullah-Al-Mamun[1], Antonina Kolokolova[1], and Dan Brake[2]

[1] Memorial University of Newfoundland
[2] EMSAT Corporation

Abstract. This work stems from the project with real-time environmental monitoring company EMSAT Corporation on online anomaly detection in their time-series data streams. The problem presented several challenges: near real-time anomaly detection, absence of labeled data, time-changing data streams. In this project, we have explored parametric statistical approach using Gaussian-based model as well as the non-parametric Kernel Density Estimation (KDE). The main contribution of this work is extending KDE to work for evolving data streams, in particular in presence of the concept drift. To address that, a framework has been developed for integrating Adaptive Windowing (ADWIN) change detection algorithm with the non-parametric method above. We have initially implemented and tested this approach on several real world data sets and received positive feedback from our industry collaborator. We also discuss several research directions for expanding this M.Sc. leading to PhD work.

Keywords: Data Stream, Anomaly Detection, Change Detection, Concept Drift, KDE, ADWIN

1 Introduction

Large amounts of quickly generated data have shifted the focus in data processing from offline, multiple-access algorithms to online algorithms tailored towards processing a stream of data in real time. Data streams are temporally ordered, fast changing and potentially infinite. Wireless sensor network traffic, telecommunications, on-line transactions in the financial market or retail industry, web click streams, video surveillance, and weather or environment monitoring are some sources of data streams. As these kinds of data can not be stored in a data repository, effective and efficient management and online analysis of data streams brings new challenges. Knowledge discovery from data streams is a broad topic which is covered in several books [4, 19], [35, ch. 4], [8, ch. 12], with [21, 5] focusing specifically on sensor data.

Outlier detection is one of the most interesting areas in data mining and knowledge discovery. This area is also referred to as anomaly detection, event detection, novelty detection, deviant discovery, fault detection, intrusion detection, or misuse detection [23]. Here, we will use the term outlier and anomaly

interchangeably. Some well established definitions of outliers are provided in [22, 25, 10]. These seemingly vague definitions covers a broad spectrum for outliers which provide the opportunity to define outlier differently in various application domains. As a result, outlier detection is a process to effectively detect outliers based on the particular definition. It is highly unlikely to find a general purpose outlier detection technique. Moreover, anomalies are mainly divided into three types: point, contextual and collective [13]. Recently a new type called contextual collective anomaly has been presented in [31].

The impetus for this work came from EMSAT Corporation, which specializes in real-time environment monitoring. With the aggregation and visualization components of their software already present, they were interested in further preprocessing and knowledge discovery in these data streams, in particular, incorporating advanced real-time quality control techniques and anomaly detection mechanism. Although some types of noise can be removed with simple rule-based techniques, much of the more subtle quality control is still done manually; we were interested in automating as much of this process as possible.

Due to the lack of labelled data in our problem domain, we focused on unsupervised methods for outlier detection. In general, they can be categorized into several groups: (i) Statistical methods; (ii) Nearest neighbour methods; (iii) Classification methods; (iv) Clustering methods; (v) Information theoretic methods and (vi) Spectral decomposition methods [13, 50]. For the types of data we were seeing, such as time-labeled streams of multivariate environmental and meteorological sensor measurements (wind speed, temperature, ocean current, etc), statistical methods seemed most appropriate.

We first explored parametric-based statistical approach using a Gaussian-based model. This technique works well if the underlying distribution fits properly and the distribution is fixed over time. But in case of evolving data stream, it is often the case is that the distribution in non-Gaussian and the underlying distribution changes over time due to concept drift. In such cases, the assumption needed for parametric approach do not apply.

And indeed, parametric approach was not showing good performance on our datasets. To remedy that, we switched to Kernel-Density Estimation (KDE) [45], following online outlier detection methods proposed in [39, 48]. KDE is primarily attractive because of four reasons: no prior assumption about the data distribution, initial data for building the model can be discarded after the model is built, scale up well for multivariate data and computationally inexpensive [50].

But even though KDE has been shown to handle evolving streams, there is no explicit mechanism to deal with concept drift. However, to improve detection of contextual anomalies, it is useful to know when the statistical properties of the data, context, changes. Even though KDE gradually adapts to the change, it may misclassify points that are close to the change point. There is a number of dedicated methods for detecting such changes in evolving data stream [20], with ADWIN [12] one of the most well-known. ADWIN has been incorporated into several predictive and clustering methods, but our goal was to integrate it with statistical approaches such as KDE.

More specifically, after initial outlier detection, we use ADWIN [12] to detect where the change has occurred, and, providing there is enough data between change points, retrain KDE on this more homogeneous stretch of the data stream. Then, some data points can be relabeled more accurately. Although change detection inevitably introduces a delay in data processing, if the data is coming fast enough, this is still a viable approach, especially provided that there is a preliminary labeling done in real time.

In the work in progress, we are working on expanding this idea to the setting of outlier ensembles of [7]. We are exploring a variety of directions, from manipulating KDE bandwidth in a sequential model-based ensemble approach, to considering an ensemble of multiple disparate outlier detection and change detection algorithms. And in the longer term, we are proposing to consider more complex anomalies such as discords, as well as investigating properties of the data which can suggest the techniques most applicable to that setting.

2 Related Work

Several extensive surveys for anomaly detection are present in the literature [29, 13, 34]. Some surveys are more focused on particular domain. Outlier detection methods for wireless sensor networks are covered in [50, 36]. In [14], the topics related to discrete sequences are present. The research issues of outlier detection for data streams are provided in [42]. For temporal/time-series data, a detail overview is presented in [18, 17, 23]. An overview of outlier detection for time-series data streams is presented in [19, ch. 11] and [6, ch. 8] . Moreover, a separate comprehensive chapter on outlier detection is presented in [24, ch. 12].

In the context of anomaly detection for environmental sensor data, a variety of ways to construct predictive models from a sensor data stream is presented in [28, 27]; the authors considered issues specific to the sensor data setting such as significant amounts of missing data and possible correlation between sensor readings that can help classify a measurement as anomalous. But these are mostly supervised methods and required a significant amount of training data. A median based approach has been used in [11]. Moreover, some simple algorithms are present for peak detection in online setting in [40].

Recently, the research direction of outlier detection is moving towards "Outlier Ensembles" [7]. Moreover, the research issues have been elaborated for outlier ensembles with a focus on unsupervised methods [51]. In [37], the authors have emphasised on using techniques from both supervised and unsupervised approaches to leverage the idea of outlier ensembles.

Another important task in processing of evolving data streams is change detection. For temporal data, the task of change detection is closely related with anomaly detection but different [6, p. 25]. The following different modes of change have been identified in the literature: concept drift (gradual change) and concept shift (abrupt change). [19, ch. 3] and [4, ch. 5] are separate chapters to cover change detection for data streams. Detecting concept drift is more difficult than concept shift. Extensive overview for detecting concept change is provided

in [44, 20]. In contrast with anomaly detection, for concept drift detection, two distributions are being compared, rather than comparing a given data point against a model prediction. Here, a sliding window of most recent examples is usually maintained, which is then compared against the learned hypothesis or performance indicators, or even just a previous time window. Much of the difference between the change detection algorithms is in the way the sliding windows of recent examples are maintained and in the types of statistical tests performed (except for CVFDT [30]), though some algorithms like ADWIN [12] allow different statistical tests to be used. These statistical tests varies from a comparison of means of old and new data, to order statistics [33], sequential hypothesis testing [38], velocity density estimation [3], density test method [46], Kullback Leibler (KL) divergence [16]. Different tests are suitable for different situations; in [15], a comparison of applicability of several of the above mentioned tests is made. There has been publicly available implementations of some of them: in particular, the MOA software environment for online learning of evolving data stream [2].

One of the most well-known algorithms for change detection is ADWIN (stands for Adapting Windowing) [12]. We base our experiments on the available implementation of ADWIN (`http://adaptive-mining.sourceforge.net`). Alternatively, we also considered using OnePassSampler [43]. Although it seems to have good performance in terms of false positive/true positive rate, its detection delay is much higher.

In [48], the proposed outlier detection method can model distribution effectively that changes over time. But it has been mentioned that detecting those changes in the distribution is difficult. It has been suggested that external change detector can be used to identify changes in distribution of streaming data. In [9, 41], the authors have proposed a regression learning framework which combines change detection mechanism with regression models. Three different external change detection mechanisms have been used and ADWIN is one of them. The framework presented in [41] detects outliers first and eliminates them. After that, change detection is done for better prediction. The main motivation of this work is not outlier detection rather improving the robustness of online prediction. In [11], the main motivation is cleaning noisy data rather than detecting contextual anomalies. This work does not consider the issue of change detection. Another framework on contextual anomaly detection for big sensor data has been presented recently [26]. The framework has both offline and online components. It generates k clusters and k Gaussian classifier for each sensor profile. The evaluation of Gaussian classifier is done online. The nature of the problem is closely related with our problem domain but this work also does not consider the issue of concept drift.

Unified techniques for change point and outlier detection are presented in [49, 32, 47]. Particularly in [49], the unified framework for change and anomaly detection has been presented. Here, the outlier detection is done in the first step. Change detection is performed later using the outcome of outlier detection.

3 Description of the Framework

For simplicity, let us consider univariate time series of environmental sensor data. There are several user-defined parameters for each stream, including maximum and minimum acceptable values, minimal sliding window size, and sensitivity threshold. The minimal sliding window size N will vary according to a particular data set. Typically it should be large enough to have a decent initial density estimation. The threshold parameter t is usually between 10^{-4} and 10^{-6}.

At the start of execution, the sliding window W will contain the initial N values. ADWIN will run on W, detecting change points. But ADWIN will stop at change point c where $|x_1...x_c| < N*l$. That is, if we cut $W = \{x_1, x_2, x_3, ...x_c, ...x_t\}$ at point c into to sub-windows then the size of first sub-window W_{prev} must be less than $N * l$, where l is an internal parameter (change point limit). This is done to ensure that the second sub-window W_{cur} will contain enough data so that the KDE can produce a fairly accurate density estimation. Now, data will be discarded from the beginning up to index $c - p$ where p is the fixed number of previous data points from last change point. As the change is sometimes detected with some delay, keeping some previous data from the change point c will not lose any data generated from current distribution. After discarding the data up to $c - p$, W is allowed to grow until $|W| = N$ again. Thus ADWIN will run on W periodically when it will reach the initial window size. For the new incoming data point x_{t+1}, it will be checked first whether it falls within the predefined acceptable range. If not, it will be flagged as bad data and discarded; instead, mean value of the current window can be used in calculations. If x_{t+1} is within acceptable range, KDE will run on W_{cur} with respect to x_{t+1} using the following equation:

$$f_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

Here, $K()$ is the kernel and h is the bandwidth. We have used the following Gaussian Kernel for our framework:

$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}} \tag{2}$$

For the Gaussian kernel, the bandwidth parameter h is calculated using the Silverman's rule-of-thumb:

$$h = \left(\frac{4\sigma^5}{3n}\right)^{1/5} \tag{3}$$

Now, the returned probability of the x_{t+1} being generated from the same distribution will be checked against the threshold t. If the probability is less than t, then it will be flagged as an anomaly, otherwise as normal. It is observed that if we decrease the value of h, the sensitivity of anomaly detection will also decrease. That is the KDE will be more restrictive. To further verify whether x_{t+1}

is an anomaly, we can repeat the same steps again by decreasing the bandwidth. This would provide us with a score of how anomalous the point is.

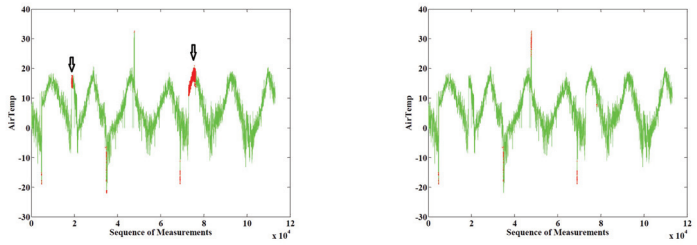
In general, we use flag values for anomalies described in the Manual for the Use of Real-Time Oceanographic Data Quality Control Flags by IOOS [1].

4 Experimental Results

We have used a publicly available data set from the SmartAtlantic Alliance project called SmartBay (`http://www.smartatlantic.ca/Home`). In Particular, the data is from a buoy placed at the Placentia Bay, Newfoundland. It measures several types of data such as Average Wind Speed, Peak Wind Speed, Wind Direction, Air Temperature, Barometric Pressure, Humidity Dew Point, Sea Surface Temperature, Maximum Wave Height, Sea Surface Salinity, Significant Wave Height etc. We have used data from August'18, 2006 - October'16, 2014. The total number of data points is around 120,000. Each measurement is taken within 20-30 minutes interval.

In all cases we are using the first N points for our initial density estimation. Thus these points are excluded for anomaly detection. The internal parameters for all cases are: ADWIN's $\delta = 0.03$, change point limit $l = 0.83$, points since last change $p = 70$. We have used Gaussian kernel for the density estimation and the bandwidth h is calculated using Silverman's rule-of-thumb as a optimal choice.

We have used the window size $N = 7000$ and different threshold value t for different data types, in particular for air temperature data $t = 10^{-4}$ and for the dew point data set $t = 10^{-5}$. We have performed all our experiments with the same parameter setting for general KDE and ADWIN+KDE.

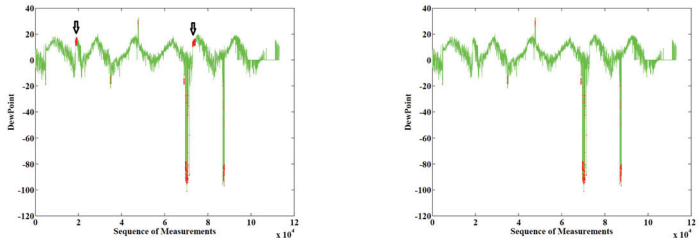


(a) ADWIN+KDE (b) KDE

Fig. 1: Comparison of air temperature anomalies detected by ADWIN+KDE versus only KDE

In case of Air Temperature, the proposed method detects one significant anomalous region where the the increase of temperature is abrupt. On the other

hand, it has correctly detected more anomalies than the general KDE.



(a) ADWIN+KDE (b) KDE

Fig. 2: Comparison of dew point anomalies detected by ADWIN+KDE versus only KDE

In case of Dew Point data set, the proposed method detects two significant anomalous regions where the the change of measurement is unusual. Again, the general purpose KDE fails to detect such events.

It seems that combining KDE with ADWIN does lead to detection of more anomalies on some data sets. However, we have seen data sets where KDE outlier detection did not change significantly with introduction of ADWIN.

5 Conclusion and Future work

Motivated by a specific problem coming from a real-world application for EMSAT Corp. real-time environmental monitoring, we have explored statistical techniques and their combination with change detection for unsupervised anomaly detection in environmental data sets. In general, KDE performed better than parameterized methods, and combination of ADWIN and KDE was able to detect possible events of interest that KDE by itself did not catch. EMSAT has found these results promising, and plans to incorporate these techniques into their product.

There are many possible directions of research and other applications of this approach. The framework requires a large-scale sensitivity analysis of its parameters. In short term, we are interested in creating ensembles of anomaly detection techniques and change detection, and evaluating their performance on environmental sensor data. We plan to include both variants of the same technique with differing parameters (for example, KDE with different kernels and/or bandwidth), and a range of different techniques. Exploring ways to address challenges specific to multivariate/high dimensional data is another part of our work in progress.

Another direction is to incorporate detection of others, more complex types of anomalies. In addition to better detection of collective anomalies, we would like to investigate detecting discords, unusual patterns in the data streams. This would depend crucially on the types of data we would have access to, as we expect different types of data to have very different structure with respect to frequent/unusual pattern occurrences.

Overall, for a longer term project, we would like to understand what properties of data streams and outlier definition make certain techniques or classes of techniques more applicable. Our current work with statistical techniques and change detection already shows that outlier detection on some data sets benefits from adding change detection, while for others KDE by itself detects outliers just as well. Analysing performance of ensembles may shed more light on such differences between types of data and outliers.

6 Acknowledgement

We are grateful to SmartBay project (http://www.smartatlantic.ca/) for allowing us to use raw data generated by their buoys.

References

1. Integrated ocean observing system. `http://www.ioos.noaa.gov`.
2. Massive online analysis. `http://moa.cms.waikato.ac.nz`.
3. Aggarwal, C. C. A framework for diagnosing changes in evolving data streams. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (2003), ACM, pp. 575–586.
4. Aggarwal, C. C. *Data streams: models and algorithms*, vol. 31. Springer, 2007.
5. Aggarwal, C. C. *Managing and mining sensor data*. Springer Science & Business Media, 2013.
6. Aggarwal, C. C. *Outlier analysis*. Springer Science & Business Media, 2013.
7. Aggarwal, C. C. Outlier ensembles: position paper. *ACM SIGKDD Explorations Newsletter 14*, 2 (2013), 49–58.
8. Aggarwal, C. C. An introduction to data mining. In *Data Mining* (2015), Springer, pp. 1–26.
9. Bakker, J., Pechenizkiy, M., Žliobaitė, I., Ivannikov, A., and Kärkkäinen, T. Handling outliers and concept drift in online mass flow prediction in cfb boilers. In *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data* (2009), ACM, pp. 13–22.
10. Barnett, V., and Lewis, T. *Outliers in statistical data*, vol. 3. Wiley New York, 1994.
11. Basu, S., and Meckesheimer, M. Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems 11*, 2 (2007), 137–154.
12. Bifet, A., and Gavalda, R. Learning from time-changing data with adaptive windowing. In *SDM* (2007), vol. 7, SIAM, p. 2007.
13. Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM Computing Surveys (CSUR) 41*, 3 (2009), 15.

14. Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection for discrete sequences: A survey. *Knowledge and Data Engineering, IEEE Transactions on 24*, 5 (2012), 823–839.
15. Dasu, T., Krishnan, S., and Pomann, G. M. Robustness of change detection algorithms. In *Advances in Intelligent Data Analysis X*. Springer, 2011, pp. 125–137.
16. Dasu, T., Krishnan, S., Venkatasubramanian, S., and Yi, K. An information-theoretic approach to detecting changes in multi-dimensional data streams. In *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications* (2006).
17. Esling, P., and Agon, C. Time-series data mining. *ACM Computing Surveys (CSUR) 45*, 1 (2012), 12.
18. Fu, T.-c. A review on time series data mining. *Engineering Applications of Artificial Intelligence 24*, 1 (2011), 164–181.
19. Gama, J. *Knowledge Discovery from Data Streams*. Chapman and Hall / CRC Data Mining and Knowledge Discovery Series. CRC Press, 2010.
20. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR) 46*, 4 (2014), 44.
21. Ganguly, A. R., Gama, J., Omitaomu, O. A., Gaber, M., and Vatsavai, R. R. *Knowledge discovery from sensor data*. CRC Press, 2008.
22. Grubbs, F. E. Procedures for detecting outlying observations in samples. *Technometrics 11*, 1 (1969), 1–21.
23. Gupta, M., Gao, J., Aggarwal, C., and Han, J. Outlier detection for temporal data. *Synthesis Lectures on Data Mining and Knowledge Discovery 5*, 1 (2014), 1–129.
24. Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.
25. Hawkins, D. M. *Identification of outliers*, vol. 11. Springer, 1980.
26. Hayes, M. A., and Capretz, M. A. Contextual anomaly detection framework for big sensor data. *Journal of Big Data 2*, 1 (2015), 1–22.
27. Hill, D. J., and Minsker, B. S. Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling & Software 25*, 9 (2010), 1014–1022.
28. Hill, D. J., Minsker, B. S., and Amir, E. Real-time bayesian anomaly detection in streaming environmental data. *Water resources research 45*, 4 (2009).
29. Hodge, V. J., and Austin, J. A survey of outlier detection methodologies. *Artificial Intelligence Review 22*, 2 (2004), 85–126.
30. Hulten, G., Spencer, L., and Domingos, P. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (2001), ACM, pp. 97–106.
31. Jiang, Y., Zeng, C., Xu, J., and Li, T. Real time contextual collective anomaly detection over multiple data streams. *Proceedings of the ODD* (2014), 23–30.
32. Kawahara, Y., and Sugiyama, M. Change-point detection in time-series data by direct density-ratio estimation. In *SDM* (2009), vol. 9, SIAM, pp. 389–400.
33. Kifer, D., Ben-David, S., and Gehrke, J. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30* (2004), pp. 180–191.
34. Kriegel, H.-P., Kröger, P., and Zimek, A. Outlier detection techniques. In *Tutorial at the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2009).

35. Leskovec, J., Rajaraman, A., and Ullman, J. D. *Mining of massive datasets*. Cambridge University Press, 2014.
36. McDonald, D., Sanchez, S., Madria, S., and Ercal, F. A survey of methods for finding outliers in wireless sensor networks. *Journal of Network and Systems Management 23*, 1 (2015), 163–182.
37. Micenková, B., McWilliams, B., and Assent, I. Learning outlier ensembles: The best of both worlds–supervised and unsupervised.
38. Muthukrishnan, S., van den Berg, E., and Wu, Y. Sequential change detection on data streams. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on* (2007), IEEE, pp. 551–550.
39. Palpanas, T., Papadopoulos, D., Kalogeraki, V., and Gunopulos, D. Distributed deviation detection in sensor networks. *ACM SIGMOD Record 32*, 4 (2003), 77–82.
40. Palshikar, G., et al. Simple algorithms for peak detection in time-series. In *Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence* (2009).
41. Pechenizkiy, M., Bakker, J., Žliobaitė, I., Ivannikov, A., and Kärkkäinen, T. Online mass flow prediction in cfb boilers with explicit detection of sudden concept drift. *ACM SIGKDD Explorations Newsletter 11*, 2 (2010), 109–116.
42. Sadik, S., and Gruenwald, L. Research issues in outlier detection for data streams. *ACM SIGKDD Explorations Newsletter 15*, 1 (2014), 33–40.
43. Sakthithasan, S., Pears, R., and Koh, Y. S. One pass concept change detection for data streams. In *Advances in Knowledge Discovery and Data Mining*. Springer, 2013, pp. 461–472.
44. Sebastiao, R., and Gama, J. A study on change detection methods. In *4th Portuguese Conf. on Artificial Intelligence, Lisbon* (2009).
45. Silverman, B. W. *Density estimation for statistics and data analysis*, vol. 26. CRC press, 1986.
46. Song, X., Wu, M., Jermaine, C., and Ranka, S. Statistical change detection for multi-dimensional data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (2007), ACM, pp. 667–676.
47. Su, W.-x., Zhu, Y.-l., Liu, F., and Hu, K.-y. On-line outlier and change point detection for time series. *Journal of Central South University 20* (2013), 114–122.
48. Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., and Gunopulos, D. Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd international conference on Very large data bases* (2006), VLDB Endowment, pp. 187–198.
49. Takeuchi, J.-i., and Yamanishi, K. A unifying framework for detecting outliers and change points from time series. *Knowledge and Data Engineering, IEEE Transactions on 18*, 4 (2006), 482–492.
50. Zhang, Y., Meratnia, N., and Havinga, P. Outlier detection techniques for wireless sensor networks: A survey. *Communications Surveys & Tutorials, IEEE 12*, 2 (2010), 159–170.
51. Zimek, A., Campello, R. J., and Sander, J. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *ACM SIGKDD Explorations Newsletter 15*, 1 (2014), 11–22.

Infusing Prior Knowledge into Hidden Markov Models

Stephen Adams, Peter Beling, and Randy Cogill

University of Virginia
Charlottesville, VA 22904 USA

Abstract. Prior knowledge about a system is crucial for accurate modeling. Conveying this knowledge to traditional machine learning techniques can be difficult if it is not represented in the collected data. We use informative priors to aid in feature selection and parameter estimation on hidden Markov models. Two well known manufacturing problems are used as case studies. An informative prior method is tested against a similar method using non-informative priors and methods without priors. We find that informative priors result in preferable feature subsets without a significant decrease in accuracy. We outline future work that includes a methodology for selecting informative priors and assessing trade-offs when collecting knowledge on a system.

Keywords: informative priors, feature selection, hidden Markov models

1 Introduction

Prior knowledge about a system can come from many sources. The cost of collecting a data stream (financial, computational, or difficulty in acquiring the feature) is not always easy to convey in a data set. Some systems have physical restrictions or properties the model must adapt to, and this can also be difficult to capture in collected data. Using these two types of information, which would not be included if only collected data were considered, will lead to models that more closely represent the system.

In Bayesian estimation, non-informative priors (NIPs) are typically used on model parameters, meaning that it assigns equal weight to all possibilities. Priors are chosen by the researchers; therefore, the researcher can influence the estimation by the selection of a prior distribution or the parameters for that distribution. Bayesians who promote NIPs wish for the data to be the only factor driving estimation, and prevent any bias or influence being injected into the estimation by the practitioner. We argue that the use of informative priors (IPs) when modeling systems is crucial for two reasons. First, knowledge about a system that is not present in the collected data can be conveyed to the estimation process through prior distributions. Second, good IPs can increase model accuracy and other notions about model performance.

Most decisions when modeling a data set are based on prior information. By choosing a class of models that one believes will accurately reflect the data,

the researcher has begun using prior information and, in a sense, has already established a prior distribution. For example, by choosing logistic regression over other classifiers, one has placed a prior with probability 1 on logistic regression and probability 0 on all other classifiers. Using the notion that IPs encompass any type of decision when modeling data, we use prior knowledge and IPs for three tasks: 1) selecting the type of model 2) selecting model structure and 3) parameter estimation.

We present case studies of two well known manufacturing problems to showcase the advantages of IPs when modeling systems. In the tool wear case study, the objective is to predict the wear given data collected from the cutting process. In the activity recognition case study, the objective is to classify the activity of a human subject using collected data such as upper-body joint positions. We use hidden Markov models (HMMs) [23] to model both systems. We chose HMMs because of their success in modeling time series data and the ability to train HMMs using unsupervised learning algorithms, which is desirable due to the difficulty labeling these types of data sets.

One area of prior knowledge we wish to convey to the model is that input features are associated with some form of cost. Test costs include the financial cost of collecting features, the time consumed collecting features and the difficulty to collect a feature [19]. We wish to construct accurate models with minimal test cost through feature selection (FS). FS with respect to test cost has been studied [8, 18]; however, most of these methods compare the trade-off between misclassification cost and test cost requiring a supervised FS technique. In addition to a reduction in test cost, FS can increase the accuracy of models, decrease computation time, and improve the ability to interpret models.

In light of this prior knowledge, we propose a feature saliency HMM (FSHMM) that simultaneously estimates model parameters and selects features using unsupervised learning. IPs are placed on some model parameters to convey test cost to the algorithm. We demonstrate, using the case studies, that the IPs produce models that compare favorably to similar models that use either no priors or NIPs. The primary contributions of this work are: a study of incorporating prior knowledge about a system into statistical modeling and the use of IPs as a mode for incorporating cost into FS using the FSHMM. It should be noted that the FSHMM outlined in Section 3, as well as some of the numerical results in Section 5, first appear in a paper under submission by two of the coauthors of this work [1], but the discussion of IPs, their use in the approach section, and future research is novel to this work.

2 Background

IPs have been used to overcome numerous modeling issues: zero numerator problems [27], low sample means and small sample sizes [16], and zero inflated regression problems [10]. Furthermore, IPs have been used with several types of models and domains: improve forecasting for monthly economic data [11], epidemiological studies using hierarchical models [25], classification and regression

trees [2], parameter estimation of non-linear systems [3], and network inference [21]. These studies demonstrate that good IPs can improve parameter estimation and predictive ability of models, however there is no well established method for converting prior knowledge into prior distributions. In [6] and [28], different methods for constructing IPs are compared but no method clearly dominates the others.

The research for FS specific to HMMs is lacking. In most applications, domain knowledge is used to select features [20]. Several studies use popular dimensionality reduction techniques such as principal component analysis (PCA) [14] or independent component analysis [26]. While these methods reduce the feature space, all features must be collected to perform the transformation so test cost is not reduced. Nouza [22] compares sequential forward search (SFS), discriminative feature analysis (DFA) and PCA. The supervised methods (SFS and DFA) outperform PCA, the unsupervised method. In [17], FS is performed using boosting. A single HMM is trained for each feature and class, then the AdaBoost algorithm is used to learn weights for each feature by increasing the weight for misclassified observations. This method requires supervised data.

Feature saliency, which recasts FS as a parameter estimation problem, was first introduced for Gaussian mixture models (GMMs) [15]. New parameters, *feature saliencies* represented by ρ, are added to the conditional distribution of the GMM. The emission probability consists of mixture-dependent and mixture-independent distributions, and ρ represent the probability that a feature is relevant and belongs to the mixture-dependent distribution. In [15], the expectation-maximization (EM) algorithm and the minimum message length criterion are used to estimate model parameters and the number of clusters. Bayesian parameter estimation techniques have also been used on the feature saliency GMM [4]. In [30], the authors use a variational Bayesian (VB) technique to jointly estimate model parameters and select features for HMMs. This method does not require the number of states to be known a priori.

3 FSHMM

Three FSHMM formulations that assume the number of states is known and use the EM algorithm to solve for model parameters are outlined in [1]. The maximum likelihood (ML) approach does not use priors on the model parameters. Two maximum a priori (MAP) formulations are given: one uses an exponential distribution with support on $[0, 1]$ as a prior on ρ and the other uses a beta distribution. The hyperparameters for these priors can be used to force estimates for ρ towards zero. It is through the IP on ρ that the test cost of a feature is conveyed to the algorithm. Features with a higher test cost must provide more relevant information to be included in the selected feature subset.

In [1], the EM models are compared with the VB method in [30]. It is shown that the EM methods outperform VB in terms of accuracy and feature subset selection. ML overestimates the relevance of irrelevant features and MAP using a beta prior underestimates the relevance of relevant features. The authors con-

clude that MAP with an exponential distribution should be used. In this section, we give a brief overview of the FSHMM using MAP with an exponential prior on ρ described in [1].

Given an HMM with continuous emissions and I states, let y_{lt} represent the l^{th} component of the observed data at time t, and let x_t represent the hidden state for $t = 0...T$. Let $a_{ij} = P(x_t = j|x_{t-1} = i)$, the transition probabilities, and $\pi_i = P(x_0 = i)$, the initial probabilities. Let $p(y_{lt}|\mu_{il}, \sigma_{il}^2)$ be the state-dependent Gaussian distribution with mean μ_{il} and variance σ_{il}^2, and $q(y_{lt}|\epsilon_l, \tau_l^2)$ be the state-independent Gaussian distribution with mean ϵ_l and variance τ_l^2. Λ is the set of all model parameters.

The EM algorithm [23] can be used to calculate maximum likelihood estimates for the model parameters. Priors can be placed on the parameters to calculate the MAP estimates [5]. The following two subsections give the E-step and M-step for the MAP FSHMM from [1].

3.1 Probabilities for E-step

First use the forward-backward algorithm [23] to calculate the posterior probabilities $\gamma_t(i) = P(x_t = i|\mathbf{y}, \Lambda)$ and $\xi_t(i,j) = P(x_{t-1} = i, x_t = j|\mathbf{y}, \Lambda)$. Then calculate the following probabilities: $e_{ilt} = P(y_{lt}, z_l = 1|x_t = i, \Lambda) = \rho_l r(y_{lt}|\mu_{il}, \sigma_{il}^2)$, $h_{ilt} = P(y_{lt}, z_l = 0|x_t = i, \Lambda) = (1 - \rho_l)q(y_{lt}|\epsilon_l, \tau_l^2)$, $g_{ilt} = P(y_{lt}|x_t = i, \Lambda) = e_{ilt} + h_{ilt}$, $u_{ilt} = P(z_l = 1, x_t = i|\mathbf{y}, \Lambda) = \frac{\gamma_{it}e_{ilt}}{g_{ilt}}$, and $v_{ilt} = P(z_l = 0, x_t = i|\mathbf{y}, \Lambda) = \frac{\gamma_{it}h_{ilt}}{g_{ilt}} = \gamma_{it} - u_{ilt}$.

3.2 MAP M-step

The priors used for MAP estimation are: $\pi \sim \mathrm{Dir}(\pi|\boldsymbol{\beta})$, $A_i \sim \mathrm{Dir}(A_i|\boldsymbol{\alpha}_i)$, $\mu_{il} \sim \mathcal{N}(\mu_{il}|m_{il}, s_{il}^2)$, $\sigma_{il}^2 \sim \mathrm{IG}(\sigma_{il}^2|\zeta_{il}, \eta_{il})$, $\epsilon_l \sim \mathcal{N}(\epsilon_l|b_l, c_l^2)$, $\tau_l^2 \sim \mathrm{IG}(\tau_l|\nu_l, \psi_l)$, $\rho_l \sim \frac{1}{Z}e^{-k_l\rho_l}$, where Dir is the Dirichlet distribution, $\mathcal{N}$ is the Gaussian distribution, IG is the inverse gamma distribution, A_i is row i of the transition matrix, and Z is the normalizing constant for the truncated exponential. The parameter update equations are:

$$\pi_i = \frac{\gamma_0(i) + \beta_i - 1}{\sum_{i=1}^{I}(\gamma_0(i) + \beta_i - 1)}, \qquad a_{ij} = \frac{\sum_{t=0}^{T}\xi_t(i,j) + \alpha_{ij} - 1}{\sum_{j=1}^{I}\left(\sum_{t=0}^{T}\xi_t(i,j) + \alpha_{ij} - 1\right)},$$

$$\mu_{il} = \frac{s_{il}^2\sum_{t=0}^{T}u_{ilt}y_{lt} + \sigma_{il}^2 m_{il}}{s_{il}^2\sum_{t=0}^{T}u_{ilt} + \sigma_{il}^2}, \qquad \sigma_{il}^2 = \frac{\sum_{t=0}^{T}u_{ilt}(y_{lt} - \mu_{il})^2 + 2\eta_{il}}{\sum_{t=0}^{T}u_{ilt} + 2(\zeta_{il} + 1)},$$

$$\epsilon_l = \frac{c_l^2\sum_{t=0}^{T}\left(\sum_{i=1}^{I}v_{ilt}\right)y_{lt} + \tau_l^2 b_l}{c_l^2\sum_{t=0}^{T}\left(\sum_{i=1}^{I}v_{ilt}\right) + \tau_l^2}, \qquad \tau_l^2 = \frac{\sum_{t=0}^{T}\left(\sum_{i=1}^{I}v_{ilt}\right)(y_{lt} - \epsilon_l)^2 + 2\psi_l}{\sum_{t=0}^{T}\left(\sum_{i=1}^{I}v_{ilt}\right) + 2(\nu_l + 1)},$$

$$\rho_l = \frac{T + 1 + k_l - \sqrt{(T + 1 + k_l)^2 - 4k_l\left(\sum_{t=0}^{T}\sum_{i=1}^{I}u_{ilt}\right)}}{2k_l}.$$

4 Data and Approach

The first case study is a data set used in the 2010 Prognostics and Health Management (PHM) Society Conference Data Challenge. [1] This data set contains force and vibration measurements for six tools; however, only three of the tools have corresponding wear measurements. Force and vibration (represented by F or V) are measured in three directions (represented by X, Y, and Z) and three features are calculated from each sensor's direction. The second case study is a human activity recognition data set of a worker engaged in a painting process in an in-production manufacturing cell. This data set was collected by researchers from the University of Virginia [24] and is publicly available. [2] Ten upper body joints were tracked using a Microsoft Kinect.

HMMs are widely used for modeling tool wear [13] and activity recognition data [9], which are both time series data with measured features correlated with a hidden variable. This prior knowledge leads to the first way we use IPs outlined in the introduction: the selection of HMMs to model the data.

Tool wear is non-decreasing. This physical attribute can be incorporated into an HMM by restricting the Markov chain to be left-to-right (LTR). A LTR Markov chain can only self-transition and transition to the next highest state. Decisions similar to choosing HMMs or restricting the Markov chain to be LTR can be considered using IPs to convey knowledge to the model structure. This is the second way we use IPs: selecting model structure.

In the PHM case study, it is assumed that the force sensor costs twice as much as the vibration sensor. This assumption was made after reviewing the price of several commercial sensors, and we believe it adequately reflects the real world. The features in the Kinect data set have a different notion about cost. The Kinect collects data every 30^{th} of a second and records three coordinates for each of the upper-body joints. The size of this data can grow rapidly, thus we associate cost with a growth in data. Irrelevant features add to computation time for the model and degrade its accuracy. Each feature is assumed to have the same collection cost, and the smallest feature subset is desired.

The third way we use IPs is to influence parameter estimates. IPs are used to convey the two previously outlined notions about cost to the FS algorithm by penalizing more costly features or larger feature subsets. For the PHM data, $k_l = 1200$ for force features and $k_l = 600$ for vibration features, which are half of the assumed cost. When modeling the Kinect data, $k_l = 15,000$, which is rougly $T/4$. These hyperparameters were selected based on intuition and not formal methodology, which is left to future work and will be discussed in a later section.

IPs are also used on $p(\cdot|\cdot)$ and $q(\cdot|\cdot)$ for the Kinect data set. A supervised initialization set is used for selecting starting values for EM. We set m to the mean of this initialization set by assuming that μ calculated on a small portion of the data will be close to the μ estimated from the training set. We assume that

[1] http://www.phmsociety.org/competition/phm/10
[2] http://people.virginia.edu/ djr7m/incom2015/

Algorithm 1 Informative Prior Distribution FSHMM Algorithm

1. Select initial values for $\pi_i, a_{ij}, \mu_{il}, \sigma_{il}, \epsilon_l, \tau_l$ and ρ_l for $i = 1..I, j = 1...I$, and $l = 1...L$
2. Select hyperparameters $\beta_i, \alpha_{ij}, m_{il}, s_{il}, \zeta_{il}, \eta_{il}, b_l, c_l, \nu_l, \psi_l$, and k_l for $i = 1..I, j = 1...I$, and $l = 1...L$
3. Select stopping threshold δ and maximum number of iterations M
4. Set absolute percent change in posterior probability between current iteration and previous iteration $\Delta\mathcal{L} = \infty$ and number of iterations $m = 1$
4. **while** $\Delta\mathcal{L} > \delta$ and $m < M$ **do**
5. E-step: calculate probabilities Section 3.1
6. M-step: update parameters Section 3.2
7. calculate $\Delta\mathcal{L}$
8. $m = m + 1$
9. **end while**
10. Perform FS based on ρ_l and construct reduced models

ϵ will be relatively close to the global mean of the data and set b to the mean of the training set. For the PHM data, the features are normalized, therefore, b is set to 0.

A general algorithm for the MAP FSHMM models is given in Algorithm 1. Leave-one-out cross validation is used for the PHM data: a supervised tool is removed, a model is trained on the remaining 5 tools, and then tested on the withheld tool. For the Kinect data, the first two thirds of the noon hour are used for training and the remaining third is reserved for testing. The first 2000 observations of the training set are used as the supervised initialization set.

In the first set of numerical experiments, we compare two FSHMM formulations: MAP using the exponential prior and the VB formulation in [30]. These results, along with the results for ML and MAP using a beta prior, are given in [1]. MAP uses IPs, while VB uses NIPs, but no priors on the feature saliencies. Prediction is performed using the Viterbi algorithm [23]. Full models using the entire feature set and reduced models using the selected feature subsets are tested. In the second set of experiments, which are novel to this work and not given in [1], the FSHMM is compared with unsupervised sequential searches. Both greedy forward and backward selection [12] are used to search the feature space, AIC and BIC are used as evaluation functions, and two stopping criteria for the search are tested. These standard FS techniques do not use priors and have no notion of test cost.

5 Numerical Experiments and Results

FS is performed on the PHM data in the first set of numerical experiments by removing the sensor direction with the lowest average ρ for the three calculated features. For comparison, models are built assuming 5 and 20 states ([1] also compares 10 state models). The root mean squared error (RMSE) between the predicted wear value and the true wear value are calculated. The predicted wear

value is the median of the predicted wear state. The average RMSE over the test tools for the MAP algorithm are: full 20 state - 22.92, reduced 20 state - 23.03, full 5 state - 24.07, and reduced 5 state - 26.05. The average RMSE for VB are: full 20 state - 36.90, reduced 20 state - 39.68, full 5 state - 34.90, and reduced 5 state - 31.24. The MAP formulation consistently removes FY, which has a higher cost than a vibration sensor. VB removes VY for the 5 state model, but the removed sensor varies depending on the training data for the 20 state model (FY for Tool 1, FX for Tool 4, and VY for Tool 6).

For the Kinect tests, FS is performed by removing features with ρ below 0.9. The fraction of correctly classified time steps is calculated and referred to as the accuracy. The MAP full and reduced models have accuracies of 0.7473 and 0.7606, while the VB full and reduced models are 0.6415 and 0.6561. The MAP formulation removes 18 features, while VB only removes 5. Effectively, the test cost for the reduced MAP model is more than three times lower than for the reduced VB model.

The first criteria stops the sequential search when there is no improvement to the evaluation function. This does not allow for control over the number of features in the feature subset. Both evaluation functions and search directions result in the same model, with VZ as the only sensor. The average RMSE is 24.36. MAP performs better on two out the three tools, but the sequential methods give a better average RMSE. For a better comparison to the test performed in the previous experiment, a second stopping criteria, which removes a single sensor then stops the search, is tested. The average RMSE for this criteria is 30.24. MAP with 5 states outperforms two of the three tools and gives a lower average RMSE. The sensor chosen for removal is dependent upon the training set (Tools 1 and 4 remove FY, while FX is removed for Tool 6). For each training set, a force sensor is removed. MAP produces a lower cost feature subset.

For the Kinect data, the initialization, training, and testing sets are divided as in the previous experiments comparing FSHMM formulations, and 6 hidden states are assumed. For SFS, both AIC and BIC yield the same final model and have 16 features. The accuracy for this model is surprisingly low at 0.0549. The "Unknown" task is predicted for all time steps except the first. SFS includes several features associated with Y and Z, and excludes features associated with X. For comparison, the FSHMM removes features in the Y direction and prefers features associated with X and Z. For SBS, AIC and BIC yield the same feature subset and both remove 3 features. The reduced model produces an accuracy of 0.5714 on the test set.

6 Discussion and Conclusion

The first set of numerical experiments demonstrate that the MAP method using IPs outperform the VB method using NIPs. MAP gives a lower RMSE on the PHM data and a higher accuracy on the Kinect data than VB. MAP also selects feature subsets that are preferable over those selected by VB. MAP consistently selects a less expensive feature subset for the PHM experiments and the smaller

feature subset in the Kinect experiments. Guyon and Elisseeff [7] state that variance in feature subset selection is an open problem that needs to be addressed in future research. Therefore, we view a consistent feature subset as a valuable trait when evaluating FS algorithms. VB, which does not use priors ρ, select subsets that vary with the training set on the PHM data. Furthermore, VB does not allow for the estimation of a LTR model. We know that wear is non-decreasing and that this should be reflected in the Markov chain. In a broader sense, the VB formulation does not allow for the use of an IP on the model structure.

For the sequential searches, AIC and BIC produce the same models, so there is little difference in these evaluation functions. When the sequential searches are run with the stopping criteria of no improvement in the evaluation function on the PHM, a single sensor is left in the reduced set. When the search is restricted to removing a single sensor, a force sensor is removed. MAP has a lower RMSE and produces a lower cost feature subset for the PHM data. The sequential searches perform much worse in terms of accuracy on the Kinect data set. MAP typically excludes features in the Y direction. This makes sense as joints in the Y direction should not vary significantly for different tasks. For example, the position of the head in the Y direction does not change much between painting and loading. From the experiments on the Kinect data, we see that the sequential search methods select features that increase the likelihood, not features that help the model accurately distinguish between states.

In conclusion, we have shown that IPs can be used and improves the modeling of two manufacturing systems using HMMs. IPs are used in the selection of the type of model, model structure, and parameter estimation. The IPs improve the models by reducing the feature set given some notion of cost of features without significantly reducing or in some cases increasing accuracy.

7 Work in Progress and Future Work

It seems logical that each of these manufacturing case studies would have some type of duration associated with the state. The explicit duration HMM (EDHMM) [29] models the residual time in a state as a random variable. There is no work concerning FS specifically for EDHMMs. Due to the significant increase in training times for these models, sequential methods that train and evaluate several models at each iteration, are eliminated from consideration. Filters or embedded techniques, such as the FSHMM, should be preferred. Current work is focused on developing an FSEDHMM for testing on these two data sets.

We are also studying using prior knowledge when selecting an emission distribution, because HMMs are not restricted to a Gaussian emission distribution, which we have assumed in this study. We are investigating GMMs, the exponential and gamma distributions, and discrete distributions such as the Poisson. GMMs are a logical choice if there appears to be multiple clusters in each state. The exponential or gamma distributions can be applied if there are no negative values in the data set.

In the current work, hyperparameters are chosen based on intuition. Future work will develop a methodology for selecting the hyperparameters for all model parameters, including calculating hyperparameters from prior knowledge, converting expert knowledge into hyperparameters, and the selection of the type of distribution used for the prior. The beta and exponential distributions are used to convey the cost of features. Other types of distributions could be used to convey different information such as physical properties.

Given a limited amount of time to study a system, the allocation of resources is important. For example, should one focus on learning as much as possible about the system before modeling, or should they focus on exploring all possible aspects of modeling. This is a trade-off between better priors and better models for the likelihood. Non-parametric Bayesian methods could provide better models for the likelihood. They have an infinite number of parameters which significantly increases their computation. Bayesian methods in general perform better when the priors give accurate information about the system. In a Bayesian setting, we now have two competing objectives: either make the priors as strong as possible or significantly increase the number of model parameters to better model the data. Non-parametric Bayesian methods with respect to IPs is an area of future research.

References

1. Adams, S., Cogill, R.: Simultaneous feature selection and model training for hidden Markov models using MAP estimation. Under Review (2015)
2. Angelopoulos, N., Cussens, J.: Exploiting informative priors for Bayesian classification and regression trees. In: IJCAI. pp. 641–646 (2005)
3. Coleman, M.C., Block, D.E.: Bayesian parameter estimation with informative priors for nonlinear systems. AIChE journal 52(2), 651–667 (2006)
4. Constantinopoulos, C., Titsias, M.K., Likas, A.: Bayesian feature and model selection for Gaussian mixture models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 28(6), 1013–1018 (2006)
5. Gauvian, J.L., Lee, C.H.: Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov models. IEEE Trans. Speech and Audio Processing 2(2), 291–298 (April 1994)
6. Guikema, S.D.: Formulating informative, data-based priors for failure probability estimation in reliability analysis. Reliability Engineering & System Safety 92(4), 490–502 (2007)
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182 (2003)
8. Iswandy, K., Koenig, A.: Feature selection with acquisition cost for optimizing sensor system design. Advances in Radio Science 4(7), 135–141 (2006)
9. Jalal, A., Lee, S., Kim, J.T., Kim, T.S.: Human activity recognition via the features of labeled depth body parts. In: Impact Analysis of Solutions for Chronic Disease Prevention and Management, pp. 246–249. Springer (2012)
10. Jang, H., Lee, S., Kim, S.W.: Bayesian analysis for zero-inflated regression models with the power prior: Applications to road safety countermeasures. Accident Analysis & Prevention 42(2), 540–547 (2010)

11. Jaynes, E.: Highly informative priors. Bayesian Statistics 2, 329–360 (1985)
12. John, G.H., Kohavi, R., Pfleger, K., et al.: Irrelevant features and the subset selection problem. In: Machine Learning: Proceedings of the Eleventh International Conference. pp. 121–129 (1994)
13. Kang, J., Kang, N., Feng, C.j., Hu, H.y.: Research on tool failure prediction and wear monitoring based HMM pattern recognition theory. In: Wavelet Analysis and Pattern Recognition, 2007. ICWAPR'07. International Conference on. vol. 3, pp. 1167–1172. IEEE (2007)
14. Kwon, J., Park, F.C.: Natural movement generation using hidden Markov models and principal components. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 38(5), 1184–1194 (2008)
15. Law, M.H., Figueiredo, M.A., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 26(9), 1154–1166 (2004)
16. Lord, D., Miranda-Moreno, L.F.: Effects of low sample mean values and small sample size on the estimation of the fixed dispersion parameter of Poisson-gamma models for modeling motor vehicle crashes: a Bayesian perspective. Safety Science 46(5), 751–770 (2008)
17. Lv, F., Nevatia, R.: Recognition and segmentation of 3-d human action using HMM and multi-class adaboost. In: Computer Vision–ECCV 2006, pp. 359–372. Springer (2006)
18. Min, F., Hu, Q., Zhu, W.: Feature selection with test cost constraint. International Journal of Approximate Reasoning 55(1), 167–179 (2014)
19. Min, F., Liu, Q.: A hierarchical model for test-cost-sensitive decision systems. Information Sciences 179(14), 2442–2452 (2009)
20. Montero, J.A., Sucar, L.E.: Feature selection for visual gesture recognition using hidden Markov models. In: Proc. 5th Int. Conf. Computer Science, 2004. ENC 2004. pp. 196–203. IEEE (2004)
21. Mukherjee, S., Speed, T.P.: Network inference using informative priors. Proceedings of the National Academy of Sciences 105(38), 14313–14318 (2008)
22. Nouza, J.: Feature selection methods for hidden Markov model-based speech recognition. Proc. 13th Int. Conf. Pattern Recognition 2, 186–190 (1996)
23. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (February 1989)
24. Rude, D., Adams, S., Beling, P.A.: A Benchmark Dataset for Depth Sensor-Based Activity Recognition in a Manufacturing Process. Ottawa, Canada (May to be presented May, 2015), `http://incom2015.org/`
25. Thomas, D.C., Witte, J.S., Greenland, S.: Dissecting effects of complex mixtures: whos afraid of informative priors? Epidemiology 18(2), 186–190 (2007)
26. Windridge, D., Bowden, R.: Hidden Markov chain estimation and parameterisation via ICA-based feature-selection. Pattern analysis and applications 8(1-2), 115–124 (2005)
27. Winkler, R.L., Smith, J.E., Fryback, D.G.: The role of informative priors in zero-numerator problems: being conservative versus being candid. The American Statistician 56(1), 1–4 (2002)
28. Yu, R., Abdel-Aty, M.: Investigating different approaches to develop informative priors in hierarchical Bayesian safety performance functions. Accident Analysis & Prevention 56, 51–58 (2013)
29. Yu, S.Z.: Hidden semi-Markov models. Artificial Intelligence 174(2), 215–243 (2010)
30. Zhu, H., He, Z., Leung, H.: Simultaneous feature and model selection for continuous hidden Markov models. IEEE Signal Processing Letters 19(5), 279–282 (May 2012)

Rankings of financial analysts as means to profits

Artur Aiguzhinov[1,2], Carlos Soares[2,3], and Ana Paula Serra[1]

[1] FEP & CEF.UP, University of Porto
[2] INESC TEC
[3] FEUP, University of Porto
artur.aguzhinov@inesctec.pt, csoares@fe.up.pt, aserra@fep.up.pt

Abstract. Financial analysts are evaluated based on the value they create for those who follow their recommendations and some institutions use these evaluations to rank the analysts. The prediction of the most accurate analysts is typically modeled in terms of individual analyst' characteristics. The disadvantage of this approach is that these data are hard to collect and often unreliable. In this paper, we follow a different approach in which we characterize the general behavior of the rankings of analysts based upon state variables rather than individual analyst's characteristics. We extend an existing adaptation of the naive Bayes algorithm for label ranking with two functions: 1) dealing with numerical attributes; and 2) dealing with a time series of label ranking data. The results show that it is possible to accurately model the relation between the selected attributes and the rankings of analysts. Additionally, we develop a trading strategy that combines the predicted rankings with the Black-Litterman model to form optimal portfolios. This strategy applied to US stocks generates higher returns than the benchmark (S&P500).

1 Introduction

In recent years, some institutions were very successful selling the rankings of analysts based on their relative performance. For example, Thomson Reuters publishes the StarMine rankings of the financial analysts on an annual basis identifying the top. The Institutional Investors magazine and Bloomberg have been publishing and selling these rankings for decades and these attract investor attention and broad media coverage. Aside from personal acknowledgment among the peers, it is still arguable if rankings of financial analysts provide valuable information to market participants and help them in selecting which analysts to follow.

Following analysts' recommendations, on average, brings value to investors [15]. Hence, following the recommendations of the top analysts should result in a profitable trading strategy. Since analysts do not make recommendations frequently, at any given moment in time, an investor may only have recommendations from analysts other than the top ones. Given that, identifying the top analysts ahead of time is beneficial for an investor. In this paper, we propose a method to predict the rankings of the analysts and use these rankings to develop a successful trading strategy.

We address the problem of rankings of analysts as a label ranking (LR) problem. Many different algorithms have been adapted to deal with LR such as: naive Bayes [1], decision-trees [5], k-nn [4,5]. However, none of these algorithms is prepared to deal with time series of rankings, which is an important characteristic of our problem. It is expected that the ranking of analysts on a given period is not independent from the ranking in the previous period. Thus, some adaptation of existing LR algorithms is required to solve this problem.

Once we have predicted rankings, we apply a trading strategy that works within the framework of the Black-Litterman (BL) model [2]. The model admits a Bayesian setting and allows to transform stock views into optimal portfolio weights. We use analysts' target prices to obtain expected returns. Using the predicted rankings, we compute analysts' views for a particular stock. These views are the input for the BL model. The resulting portfolio maximizes the Sharpe ratio [12]. We use S&P500 as a proxy for market returns. We show that 1) our LR model outperforms other forecasting models; 2) the resulting trading strategy generates superior returns.

The contributions of our paper are the following. We are able to adapt the existing LR algorithm and apply it to a real world problem of predicting the rankings of financial analysts. Using the predicted rankings as inputs, we design a profitable trading strategy based on the BL model

The paper is organized as follows: Section 2 reviews the rankings of the analysts in the finance literature; Section 3 formalizes the label ranking problem and introduces the adaptation of the algorithm to deal with time series of the rankings; Section 4 outlines the trading strategy that uses the predicted rankings; Section 5 describes the datasets used for the experiments; Section 6 analyzes the results; and Section 7 concludes.

2 Ranking of Financial Analysts

In the finance literature there has been a long debate over whether financial analysts produce valuable advice. Some argue that following the advice of financial analysts, translated as recommendations of buying, holding, or selling a particular stock, does not yield abnormal returns, i.e., returns that are above the required return to compensate for risk [8]. If financial markets are efficient then any information regarding a stock would be reflected in its current price; hence, it would be impossible to generate abnormal returns based upon publicly available information. This is the Efficient Market Hypothesis (EMH).

Yet there are information-gathering costs and the information is not immediately reflected on prices [9]. As such, prices could not reflect all the available information because if that was the case, those who spent resources to collect and analyze information would not receive a compensation for it.

For market participants, rankings could be useful because they signal the top analysts. Evidence shows that market response to analysts' recommendations is stronger when they are issued by analysts with good forecasting tracking record [11]. Yet the value of these rankings for investors is arguable as they are ex-post

and a good analyst in one year does not necessarily make equally good recommendations in the following year [7]. However, if we know the ranking of analysts ahead of time then it would be possible to create a successful trading strategy based upon that information. If we can, with reasonable accuracy, predict the rankings we can follow the recommendations of the analysts that are expected to be at the top and, in presence of contradictory recommendations, take the rank of the corresponding analysts into account.

3 Label ranking algorithm

The classical formalization of a label ranking problem is the following [13]. Let $\mathcal{X} = \{\mathcal{V}_1, \ldots, \mathcal{V}_m\}$ be an instance space of variables, such that $\mathcal{V}_a = \{v_{a,1}, \ldots, v_{a,n_a}\}$ is the domain of nominal variable a. Also, let $\mathcal{L} = \{\lambda_1, \ldots, \lambda_k\}$ be a set of labels, and $\mathcal{Y} = \Pi_{\mathcal{L}}$ be the output space of all possible total orders over $\mathcal{L}$ defined on the permutation space Π. The goal of a label ranking algorithm is to learn a mapping $h : \mathcal{X} \to \mathcal{Y}$, where h is chosen from a given hypothesis space $\mathcal{H}$, such that a predefined loss function $\ell : \mathcal{H} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is minimized. The algorithm learns h from a training set $\mathcal{T} = \{x_i, y_i\}_{i \in \{1,\ldots,n\}} \subseteq \mathcal{X} \times \mathcal{Y}$ of n examples, where $x_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\} \in \mathcal{X}$ and $y_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,k}\} \in \mathcal{Y}$. With time-dependent problem in rankings, we replace the i index with t; that is $y_t = \{y_{t,1}, y_{t,2}, \ldots, y_{t,k}\}$ is the ranking of k labels at time t described by $x_t = \{x_{t,1}, x_{t,2}, \ldots, x_{t,m}\}$ at time t.

Consider an example of a time-dependent ranking problem presented in Table 1. In this example, we have three brokers ($k = 3$), four independent variables ($m = 4$) and a period of 7 quarters. Our goal is to predict the rankings for period t, given the values of independent variables and rankings known up to period $t - 1$; that is, to predict the ranking for time $t = 7$, we use $n = 6$ ($t \in \{1 \ldots 6\}$) examples to train the ranking model.

Table 1. Example of label ranking problem

Period	$\mathcal{V}_1$	$\mathcal{V}_2$	$\mathcal{V}_3$	$\mathcal{V}_4$	Ranks		
					Alex	Brown	Credit
1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	1	2	3
2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	2	3	1
3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	1	2	3
4	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	3	2	1
5	$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$	3	2	1
6	$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$	2	1	3
7	$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$	1	2	3

3.1 Naive Bayes algorithm for label ranking

The naive Bayes for label ranking (NBLR) will output the ranking with the higher $P_{LR}(y|x)$ value [1]:

$$\hat{y} = \arg\max_{y \in \Pi_{\mathcal{L}}} P_{LR}(y|x) = \tag{1}$$
$$= \arg\max_{y \in \Pi_{\mathcal{L}}} P_{LR}(y) \prod_{i=1}^{m} P_{LR}(x_i|y)$$

where $P_{LR}(y)$ is the prior label ranking probability of ranking $y \in Y$ based on the similarity between rankings obtained from the Spearman ranking correlation Equation (2):

$$\rho(y, y_i) = 1 - \frac{6 \sum_{j=1}^{k} (y - y_{i,j})^2}{k^3 - k} \tag{2}$$

$$P_{LR}(y) = \frac{\sum_{i=1}^{n} \rho(y, y_i)}{n} \tag{3}$$

Similarity and probability are different concepts; however, a connection as been established between probabilities and the general Euclidean distance measure [14]. It states that maximizing the likelihood is equivalent to minimizing the distance (i.e., maximizing the similarity) in a Euclidean space.

$P_{LR}(x_i|y)$ in Equation (1) is the conditional label ranking probability of a nominal variable x of attribute a, (v_a):

$$P_{LR}(x_i|y) = \frac{\sum_{i:x_a = v_a} \rho(y, y_i)}{|\{i : x_a = v_a\}|} \tag{4}$$

The predicted ranking for an example x_i is the one that will receive the maximum posterior label ranking probability $P_{LR}(y|x_i)$.

Continuous independent variables In its most basic form, the naive Bayes algorithm cannot deal with continuous attributes. The same happens with its adaptation for label ranking [1]. However, there are versions of the naive Bayes algorithm for classification that support continuous variables [3]. The authors modify the conditional label ranking probability by utilizing the Gaussian distribution of the independent variables. We apply the same approach in defining the conditional probability of label rankings:

$$P_{LR}(x|y) = \frac{1}{\sqrt{2\pi}\sigma(x|y)} e^{\frac{(x - \mu(x|y))^2}{2\sigma^2(x|y)}} \tag{5}$$

where $\mu(x|y)$ and $\sigma^2(x|y)$ weighted mean and weighted variance, defined as follows:

$$\mu(x|y) = \frac{\sum_{i=1}^{n} \rho(y, y_i) x}{\sum_{i=1}^{n} \rho(y, y_i)} \tag{6}$$

$$\sigma^2(x|y) = \frac{\sum_{i=1}^{n} \rho(y, y_i)[x - \mu(x|y)]^2}{\sum_{i=1}^{n} \rho(y, y_i)} \tag{7}$$

3.2 Time series of rankings

The time dependent label ranking (TDLR) problem takes the intertemporal dependence between the rankings into account. That is, rankings that are similar to the most recent ones are more likely to appear. To capture this, we propose the weighted TDLR prior probability:

$$P_{TDLR}(y_t) = \frac{\sum_{t=1}^{n} w_t \rho(y, y_t)}{\sum_{t=1}^{n} w_t} \tag{8}$$

where $w = \{w_1, \ldots, w_n\} \rightarrow \mathbf{w}$ is the vector of weights calculated from the exponential function $\mathbf{w} = b^{\frac{1-\{t\}_1^n}{t}}$. Parameter $b \in \{1 \ldots \infty\}$ sets the degree of the "memory" for the past rankings, i.e., the larger b, the more weight is given to the last known ranking (i.e, at $t-1$) and the weight diminishes to the rankings known at $t = 1$.

As for the conditional label ranking probability, the equation for the weighted mean (Equation (5)) becomes:

$$\mu(x_t|y_t) = \frac{\sum_{t=1}^{n} w_t \rho(y, y_t) x_t}{\sum_{t=1}^{n} \rho(y, y_t)} \tag{9}$$

and σ:

$$\sigma^2(x_t|y_t) = \frac{\sum_{i=1}^{n} w_t \rho(y, y_t)[x_t - \mu(x_t|y)]^2}{\sum_{i=1}^{n} \rho(y, y_t)} \tag{10}$$

4 Trading Strategy

4.1 Independent variables

Several studies try to analyze factors that affect the performance of the analysts [6,10]. However, most of these papers look at the individual characteristics of analysts such as their job experience, their affiliation, education background, industry specializations. These variables are very important to characterize the relative performance of the analysts in general. Yet, our goal is to predict the rankings of the analysts over a series of quarters. We assume that the variation in rankings is due to the different ability of the analysts to interpret the informational environment (e.g., whether the market is bull or bear). We, thus, use variables that describe this environment. We select variables based on different levels of information: analyst-specific (analysts' dispersion; analysts' information asymmetry; analysts' uncertainty), stock-specific (stock return volatility, Book-to-Market ratio; accruals; Debt-to-Equity ratio), industry-specific (Sector index volatility) and general economy (interest rate; Gross National Product; inflation rate; S&P 500 volatility).

Given the time series of the rankings and independent variables, we also need to capture the dynamics of independent variables from one time period to another; that is, to find signals that affect brokers' forecasts accuracy. We propose the following methods of dynamics:

- `last`—no dynamics of x: $x_{t,m} = x_{t-1,m}$;
- `diff`—first-difference of x: $x_{\Delta t,m} = x_{t,m} - x_{t-1,m}$;
- `random`—an unobserved component of time series decomposition of x: $x_{\Delta t,m} = T(t) + S(t) + \epsilon(t)$, where $T(t)$- trend, $S(t)$ - seasonal part and $\epsilon(t)$ - random part of time series decomposition.
- `roll.sd`—moving 8 quarters standard deviation of x [16]:

$$\mu_{t(8),m} = \frac{1}{8}\sum_{j=0}^{7} x_{t-j,m} \qquad \sigma^2_{t(8),m} = \frac{1}{7}\sum_{j=0}^{7}\left(x_{t-j,m} - \mu_{t(8),m}\right)^2 \tag{11}$$

Each of these methods produces a different set of attributes. By using the algorithm on each one of them separately, we get different rankings. By evaluating them, we can get an idea of which one is the most informative.

4.2 Strategy setup

The Black-Litterman model [2] is a tool for active portfolio management. The objective of the model is to estimate expected returns and optimally allocate the stocks in a mean-variance setting, i.e., maximize the Sharpe ratio.

The BL model has established notations for the views part of the model and we use the same notations in this paper. The views are made of: Q—the expected stock return; Ω—the confidence of Q. For the market inputs, the model requires a vector of equilibrium returns.

The trading strategy is applied as follows:

1. For each stock s, at the beginning of quarter q, we predict the rankings of all analysts that we expect to be at the end of the quarter q;
2. Based on these predicted rankings and analysts' price targets, we define $Q_{q,s}$ and $\Omega_{q,s}$;
3. Using market information available at the last day of quarter $q-1$, we obtain the market inputs;
4. Apply BL model to get optimized portfolio weights and buy/sell stocks accordingly;

To measure the performance of our portfolio, we compare it to the baseline which is the market portfolio (S&P500). We compare the relative performance of our portfolio using the Sharpe ratio:

$$SR = \frac{r_p - r_f}{\sigma_p} \tag{12}$$

where r_p is the portfolio quarterly return and r_f is the risk-free rate; σ_p is the standard deviation of the portfolio returns.

5 Data and experimental setup

To implement the trading strategy, we focus on the S&P500 stocks. Given that we base stock views on the analysts' price target information, the period of the strategy experiment runs from the first quarter of 2001 until the last quarter of 2009. We get price target from ThomsonReuters. The list of S&P constituents and stock daily prices data is from DataStream as well as the market capitalization data. The total number of brokers in price target dataset includes 158 brokers covering 448 stocks all of which at some point in time were part of the S&P 500. Given the fact that analysts issue price targets annually, we assume that an analyst keeps her price target forecast valid for one calendar year until it either is revised or expire.

5.1 Target rankings

We build the target rankings of analysts based on the Proportional Mean Absolute Forecast Error (PMAFE) that measures the accuracy of a forecasted price target ξ. First, we define the forecast error (Δ) as an absolute value of the difference between actual price ξ_s and the price target made by an analyst k ($\hat{\xi}_{k,s}$):

$$\Delta_{t,k,s} = |\xi_{t,s} - \hat{\xi}_{t,k,s}| \tag{13}$$

Then, we calculate the average error across analysts as:

$$\overline{\Delta}_{t,s} = \frac{1}{k} \sum_{k=1}^{k} \Delta_{t,k,s} \tag{14}$$

Next, PMAFE is given as:

$$\tilde{\Delta}_{t,k,s} = \frac{\Delta_{t,k,s}}{\overline{\Delta}_{t,s}} \tag{15}$$

5.2 Information sets to define the views

To proceed with the trading strategy, we need to establish which information we will be using to build the rankings. These rankings will be the inputs to compute the weighted return estimates ("smart estimates"). Different analysts' ranks are obtained if we select different time horizons. If we use only the most recent information, we will capture the recent performance of the analysts. This, of course, is more sensitive to unique episodes (e.g., a quarter which has been surprisingly good or bad). If, alternatively, we opt to incorporate the entire analyst's performance, the ranking is less affected by such events, yet it may not reflect the current analyst's ability. We use two information sets: the first uses only the information about the analyst's performance in period $t-1$; the second, uses all the available information for that particular analyst. We call the former the *recent* set and the latter the *all-time* set. We use rankings based on these information sets as the baseline rankings.

In addition to these sets, we also create a hypothetical scenario that assumes we anticipate perfectly the future analyst accuracy performance that would only be available at the end of t. This represents the perfect foresight strategy. The perfect foresight refers to analysts' rankings not stock prices. Therefore, it serves a performance reference point to evaluate the other trading strategies. We call this the *true* set.

6 Experimental Results

The results of the trading strategy based on predicted analysts' rankings are presented in (Table 2).

Table 2. Trading strategy performance

Strategy	Annualized cum. return (in %)	Annualized Std. dev (in %)	Sharpe ratio	Average num. stock	Average turnover rate
Panel A					
Market	-3.032	16.654	-0.182	499	0.053
Panel B: TP					
true	1.785	15.312	0.117	240	0.272
recent	0.634	15.444	0.041	240	0.251
all-time	0.587	15.325	0.038	240	0.238
`last`	0.513	15.478	0.033	240	0.262
`diff`	0.779	15.507	0.050	240	0.269
`random`	0.671	15.474	0.043	240	0.258
`roll.sd`	0.634	15.464	0.041	240	0.264

Panel A reports the performance of *market* (passive strategy). This strategy showed annualized cumulative return of -3.03% and annualized Sharpe ratio of -0.18. The average number of stocks used per quarter is 499.98 and the turnover ratio of strategy is 0.05 which demonstrates the ins/outs of the S&P 500 constituents list.

Panel B of Table 2 demonstrates the results of trading with rankings based on price target. Consistent with our assumption, the *true* resulted in the maximum possible annual cumulative return and the Sharpe ratio (1.78% and 0.12 respectively). This implies that in the settings where analysts' expected returns and rankings are based on price targets, an investor can gain a maximum results from trading strategy. Given the hypothetical assumption of *true*, it is not feasible to implement. The next best strategy is `diff` which is based on our algorithm of predicting the rankings. This strategy resulted in annual cumulative return of 0.78% and the Sharpe ratio of 0.05. In addition, the average per quarter turnover ratio of this strategy of 0.27 implies relative low trading costs.

Figure 1 plots the graphical representation of the cumulative returns for all methods of trading strategy. We see that the *true* strategy is always on top of all the others. We observe that the best outcome was achieved for the strategy based on the first difference of the independent variables.

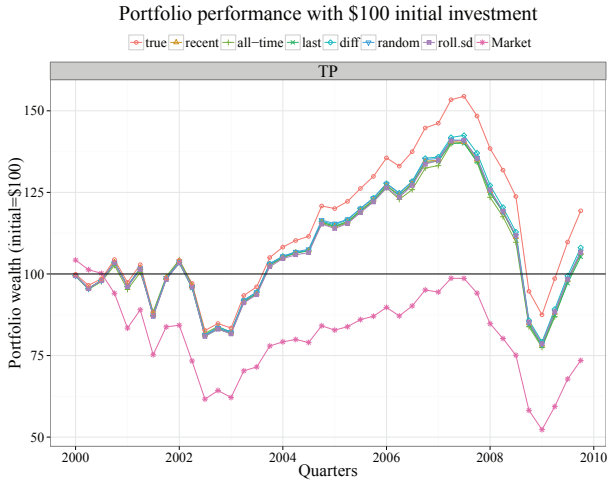


Fig. 1. Performance of the BL model

7 Conclusion

Some institutions, such as StarMine, rank financial analysts based on their accuracy and investment value performance. These rankings are published and are relevant: stocks favored by top-ranked analysts will probably receive more attention from investors. Therefore, there is a growing interest in understanding the relative performance of analysts. In this paper we developed an algorithm that is able to predict the rankings based on state variables that characterize the information environment of the analysts. Further, we designed and operationalized a trading strategy based on the Black-Litterman model with rankings as inputs. We obtained positive successful results from trading that out-performs both the market and the baseline ranking prediction.

References

1. Aiguzhinov, A., Soares, C., Serra, A.: A similarity-based adaptation of naive Bayes for label ranking: Application to the metalearning problem of algorithm recommendation. In: Discovery Science. Lecture Notes in Computer Science, vol. 6332, pp. 16–26 (2010)
2. Black, F., Litterman, R.: Global portfolio optimization. Financial Analysts Journal 48(5), 28–43 (1992)
3. Bouckaert, R.R.: Naive Bayes classifiers that perform well with continuous variables. In: AI 2004: Advances in Artificial Intelligence. Lecture Notes in Computer Science, vol. 3339, pp. 85–116 (2005)
4. Brazdil, P., Soares, C., Costa, J.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning 50(3), 251–277 (2003)
5. Cheng, W., Hühn, J., Hüllermeier, E.: Decision tree and instance-based learning for label ranking. In: ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 161–168. ACM, New York, NY, USA (2009)
6. Clement, M.: Analyst forecast accuracy: Do ability, resources, and portfolio complexity matter? Journal of Accounting and Economics 27(3), 285–303 (1999)
7. Emery, D., Li, X.: Are the Wall Street analyst rankings popularity contests? Journal of Financial and Quantitative Analysis 44(2), 411 (2009)
8. Fama, E.: Efficient capital markets: A review of empirical work. The Journal of Finance 25, 383–417 (1970)
9. Grossman, S., Stiglitz, J.: On the impossibility of informationally efficient prices. American Economic Review 70, 393–408 (1980)
10. Jegadeesh, N., Kim, J., Krische, S., Lee, C.: Analyzing the analysts: When do recommendations add value? The Journal of Finance 59(3), 1083–1124 (2004)
11. Park, C., Stice, E.: Analyst forecasting ability and the stock price reaction to forecast revisions. Review of Accounting Studies 5(3), 259–272 (2000)
12. Sharpe, W.: Mutual fund performance. The Journal of Business 39(1), 119–138 (1966)
13. Vembu, S., Gärtner, T.: Label ranking algorithms: A survey. In: Fürnkranz, J., Hüllermeier, E. (eds.) Preference Learning, pp. 45–64. Springer (2010)
14. Vogt, M., Godden, J., Bajorath, J.: Bayesian interpretation of a distance function for navigating high-dimensional descriptor spaces. Journal of chemical information and modeling 47(1), 39–46 (2007)
15. Womack, K.: Do brokerage analysts' recommendations have investment value? The Journal of Finance 51, 137–168 (1996)
16. Zivot, E., Wang, J.: Modeling financial time series with S-PLUS, vol. 191. Springer Verlag (2003)

Multi-Label Classification by Label Clustering based on Covariance

Reem Al-Otaibi[1,2], Meelis Kull[1], and Peter Flach[1]

[1]Intelligent System Laboratory, Computer Science, University of Bristol
Bristol, United Kingdom
[2]King Abdulaziz University, Saudi Arabia
{ra12404, meelis.kull, peter.flach}@bristol.ac.uk

Abstract. Multi-label classification is a supervised learning problem that predicts multiple labels simultaneously. One of the key challenges in such tasks is modelling the correlations between multiple labels. **LaCova** is a decision tree multi-label classifier, that interpolates between two baseline methods: Binary Relevance (BR), which assumes all labels independent; and Label Powerset (LP), which learns the joint label distribution. In this paper we introduce **LaCova-CLus** that clusters labels into several dependent subsets as an additional splitting criterion. Clusters are obtained locally by identifying the connected components in the thresholded absolute covariance matrix. The proposed algorithm is evaluated and compared to baseline and state-of-the-art approaches. Experimental results show that our method can improve the label exact-match.

Keywords: Multi-label learning; Decision trees; Covariance matrix; Splitting criteria; Clustering

1 Introduction

Classification is a learning task in which the goal is to categorise a new instance into one or more classes using the trained model. Single-label classification (binary and multiclass) is concerned with classifying a new instance into one and only one class. In binary classification, each training point can belong to one of two classes, whereas, in multiclass, the setting is more general, so that each training point can belong to one of more than two classes.

On the other hand, in multi-label classification, training points are associated with a set of classes (labels) simultaneously. There is a wide range of applications for multi-label data, such as text categorisation and image and movie tagging. For example, in a medical diagnosis, a patient may have multiple diseases at the same time.

Decision trees are one of the most popular algorithms for classification. They are built recursively by partitioning the training data. They consist of nodes: starting from the root node, the internal node represents an attribute, and the leaf node represents a class label or a probability distribution over classes. The internal node splits the training space into two or more nodes based on the chosen splitting criterion.

Among other advantages, decision trees are intuitive and comprehensible, such that the structure can be captured by inexpert users easily. Furthermore, they are selective,

which means they select the most discriminative features from the data to construct the trees, which usually number much less than the actual number of features. This has the advantage, especially in a high-dimensional feature space [11].

In fact, with regard to multi-label tasks, there are many strategies to apply the decision trees. On one hand, a single decision tree can be learnt for each label, ignoring the correlation between different labels. This approach is known as binary relevance (BR). On the other hand, a single tree can be learnt, which makes predictions for all labels together. it is known as label powerset (LP). In the work proposed in [1], authors have developed a tree based multi-label classifier using a label covariance as splitting criterion called **LaCova**. The key idea of **LaCova** is to use the label covariance matrix at every node of the tree in order to decide to treat labels independently (BR) or keep them all together (LP) and find a feature to split on.

The aim of this paper is to explore how to mediate between keeping all labels together or treating them independently, by studying ways in which the covariance matrix can suggest label clusters. Moreover, we incorporate ensemble models with **LaCova-CLus** and compare it with other ensemble algorithms.

The rest of this paper is organised as follows: Section 2 discusses the problem definition and possible approaches to learn multi-label decision trees; the **LaCova-CLus** is presented in Section 3; Section 4 gives an overview of existing approaches to introduce decision trees into multi-label classification; experimental results are presented in Section 5; and Section 6 concludes the paper.

2 Problem Definition

Before giving a formal definition, suppose the learning task is to classify a given movie into several genres so the task in this case is a multi-label classification problem. To build a decision tree for this dataset a number of methods can be used.

One of the most common approaches and perhaps the simplest way, is to build a single decision tree for each label and train it independently from the other labels' trees. So in this case, we create a tree for each movie genre. In order to do this, we need to transform the dataset into several subsets where each subset has information only about one movie category. Given a new movie, in order to predict its categories, all trees should be tested. The prediction for this movie will be the union of all binary decision trees. This approach learns number of trees equals to the number of labels, which can be hundreds or thousands in some domains.

The second approach is to learn one decision tree for all movie genres and predict all of them once. This method models the joint distributions and learns all the labels as one set. Although this approach is simple and effective, it might end up with few instances at a leaf when the number of labels is increased.

In summary, we will justify about our approach based on the advantages and disadvantages of each approach. The first method does not exploit dependencies among the labels, which is one of the main challenges in a multi-label classification [8]. Moreover, the number of learnt trees can be large, particularly in the high-dimensional space, which can be hundreds or thousands in some domains (it can be up to 174 in our experiment) . Finally, from the knowledge discovery point of view, the resulting trees of

this approach identify features relevant for one label, rather than identifying features with overall relevance [14]. In the second approach, over-fitting is an issue because it might end up with a few instances at a leaf. In addition, the exponential number of label combinations is a potential issue with this approach.

This paper proposes an algorithm that combines these two baseline approaches and produces a hybrid tree. **LaCova-CLus** uses the thresholded absolute covariance matrix in order to find the connected components among labels locally. These components are partitioned into clusters. For each cluster, we learn a single decision tree (LP).

3 The LaCova-CLus Algorithm

The area of multi-label classification has attracted many researchers. One of the main challenges is the large number of labels in real-world applications; importantly, these labels may present some correlation between them. Thus, exploiting dependencies among the labels without increasing complexity could improve the classifier performance [8,9]. Acknowledging the benefits of decision tree models, we focus our work on decision trees themselves.

In this paper, we propose **LaCova-CLus**, which is different from the previous methods as it clusters labels dynamically during the construction of the decision tree. It also tests label dependencies while growing the tree, which might change and lead to change in label clusters.

Standard decision tree algorithms use greedy searches to select splits that maximally decrease the impurity from a node to its children. There are many ways to measure the impurity of a set of instances, including entropy and Gini index. We note that the Gini index $p_j(1-p_j)$ for a binary class label j with proportion of positives p_j is the variance of a Bernoulli distribution with success probability pj. With multiple binary labels we can also consider label covariance, which for two Bernoulli variables j and k with success probabilities p_j and p_k and joint probability p_{jk} is $p_{jk} - p_j \cdot p_k$. For a set of $|L|$ labels we can form an L-by-L covariance matrix with label variances on the diagonal and pairwise covariances off the diagonal.

LaCova implemented a three-way splitting criterion, which can be summarised as follows. Firstly, if the trace of this matrix (the sum of the diagonal entries) is small, then the set of instances is nearly pure in a multi-label sense. Secondly, if the labels are actually independent, i.e., each pairwise covariance is low in magnitude, then apply BR at this point. We assessed this by calculating total absolute covariance, where the absolute value is taken to avoid positive and negative covariances cancelling out. Finally, learn all labels together and find a feature to split on. The main algorithms are given in [1].

The covariance threshold λ is required to decide whether there is a significant dependence between the labels or not. Authors in [1] derived a threshold λ that is also computed dynamically at each node of the tree. For more details on derivation λ, see [1].

The first two choices require a threshold on the sum of variances and the sum of absolute covariances, respectively. In experiments we found that, in combination with a minimum number of instances in a leaf, a variance threshold of 0 (i.e., all labels pure) works well. The covariance threshold requires a second innovation, which is presented

Algorithm 1 LaCova-CLus (D): Learn a tree-based multi-label classifier from training data.

Input: Dataset D; Labels set L, Minimum number m of instances to split
Output: Tree-based multi-label classifier
CM=Covariance Matrix
if SumOfVar$(CM) = 0$ or $|D| < m$ **then**
  Return Leaf with relative frequencies of labels
else if SumOfAbsCov$(CM) \leq \lambda$ **then**
  for each label j in D **do**
    T_j = Learn a decision tree for single label (BR)j
  end for
  Return Node with single-label decision tree T_j
else
  $clusters$=**CLUST**(L, CM)
  if $|clusters| > 1$ **then**
    for each set s in $clusters$ **do**
      T_s = Learn a decision tree for set of labels s (LP)
    end for
    Return Node with LP labels decision tree T_s
  else
    $f, \{D_i\}$ = FindBestSplit(D)
    for each child node D_i **do**
      T_i = **LaCova-CLus** (D_i)
    end for
    Return Node splitting on f with subtrees T_i
  end if
end if

in the next section. This leads to the main algorithm given in Algorithm 1, which implements the above three-way split.

In this work, we address how to mitigate between two options: learning a separate tree for each label or keep all labels together and learn one tree. The basic idea of **LaCova-CLus** is to find useful clusters using the thresholded absolute covariance matrix and decompose the set of labels into several subsets of dependent labels, build an LP classifier for each subset. It also combines BR for independent labels. In addition to the above mentioned three-way splitting criterion, **LaCova-CLus** incorporates a fourth option as follows (see Algorithm 1).

1. If the sum of variances is low, stop growing the tree.
2. Or, if the sum of absolute covariances is low, apply BR (vertical split).
3. Or, if there are label clusters, apply LP (vertical split) for each cluster (**LaCova-CLus**).
4. Or, find a good feature to split on and recurse (horizontal split).

3.1 Clustering

Algorithm 2 identifies label clusters based on the thresholded absolute covariance matrix. The first step is to assume that all labels are independent and in separate clusters. Then, it determines which pair of labels to merge in one cluster based on tunable parameter λ. The algorithm continues building the clusters by merging the clusters gradually. These clusters are generated dynamically within the tree.

Algorithm 2 CLUST(L,CM): Cluster labels based on the covariance matrix

Input: A set of labels $L = l_1, \cdots, l_{|L|}$; Covariance Matrix *CM*
Output: *n*ewClust - The final label clusters
Initialise *c*urrClust=null; *n*ewClust=null; pairList=null;
/* Build initial clusters by assuming each label is in a separate cluster. */
*c*urrClust=$\{l_1\},\{l_2\},\cdots,\{l_{|L|}\}$
/* Create a list of label pairs sorted in descending order of the absolute covariance value. */
pairList $\leftarrow$ sorted list
for each label pair(l_i, l_j), where $i = 0 \cdots |L|-1$ and $j = i+1 \cdots |L|$ **do**
  if all labels are in the same cluster **then**
    Stop clustering
  end if
  if AbsCov$(CM, l_i, l_j) > \lambda$ **then**
    /* Merge l_i and $l_j$to a new cluster and update the clusters. */
    *n*ewClust=*c*urrClust $\cup (l_i, l_j)$
  end if
end for
Return *n*ewClust

We use the same threshold to decide if a pair of labels are dependent or not to merge them in one cluster.

$$\lambda = \hat{\mu} + 2\hat{\sigma}$$

$$\hat{\mu} = \sqrt{\frac{2}{(n-1)\pi}} \sqrt{p_j p_k (1-p_j)(1-p_k)}$$

$$\hat{\sigma}^2 = \frac{1-\frac{2}{\pi}}{n-1} p_j p_k (1-p_j)(1-p_k)$$

where p_j and p_k are the probabilities of two labels j and k. n is the number of instances reaching a particular tree node.

4 Related Work

Many different directions have been taken in the literature to adapt decision trees for multi-label problems. We now summarise them into different approaches as follows.

A first approach transforms a multi-label problem into several binary classification tasks and builds a decision tree for each label separately which is known as BR. To classify a new instance, it outputs the union of the labels that are positively predicted by all the trees. Classifier Chains (CC) is similar to the BR concept, however, it considers labels correlation. It learns binary classifier for each label along the chain (labels ordering). Features of each classifier in the chain is incremented with the predictions of all previous classifiers along the chain [10]. Ensembles of Classifier Chains (ECC) [10] use CC as a base classifier by training a number of CC classifiers. Each classifier is trained with different order of labels in the chain and a random subset of the data.

The second method learns a single tree and predicts all the labels together. Such example is proposed by authors in [3], they adapted the C4.5 decision tree to deal with multi-label data, while the basic strategy was to define multi-label entropy over a set of multi-label examples separately. The modified entropy sums the entropies for each individual label. Although, this approach is able to identify features that are relevant to

all labels at once, splitting is not guided by label correlations. Another recent work is proposed in [6], which also builds a single tree for a multi-label dataset. They proposed a hybrid decision tree model that utilises support vector machines (SVMs) at its leaves. It is known as ML-SVMDT and it combines two models: ML-C4.5 and BR. It builds single decision trees similar to ML-C4.5, where the leaves contain BR classifiers that give multi-label predictions using SVM.

The final approach exploits the correlation between labels by applying clustering approaches. Hierarchy of multi-label classifiers (HOMER) organises all labels into a tree-shaped hierarchy with a smaller set of labels at each node [13]. In the training phase, a multi-label classifier is trained for each internal node to predict a set of labels called a meta-label. Then, it proceeds into the successor nodes of the meta-label if they belong to this set of labels. Leaf nodes construct a binary classifier to predict only a single label. Another recent work in [2] combines the LP and BR methods and is called LPBR. Its first step is to explore dependencies between labels and then to cluster these labels into several independent subsets according to the chi square χ^2 statistic. Second, a multi-label classifier is learnt: if the set contains only one label, BR is applied; otherwise, LP is used for a group of dependent labels.

5 Experimental Evaluation

LaCova, **LaCova-CLus** and ML-C4.5 have been implemented in Java using Meka[1]. Meka was used for BR, LP, and CC, whereas Mulan[2] was used for the LPBR algorithm. In all these algorithms, the trees are produced by J48 algorithm.

Initially, we performed experiments and compared **LaCova-CLus** to other baseline approaches: BR and LP. Then, we evaluated and compared the proposed model and its ensemble version to other state-of-the-art multi-label algorithms.

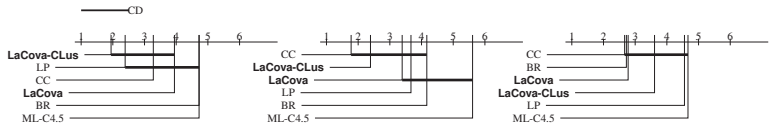
LPBR needs parameters configuration such as non-improving counter to stop clustering. The default parameters setting in Mulan were 10 for non-improving counter and 10-fold cross validation for testing the clustering performance. For large datasets it takes days to run the experiments and then cause out of memory error. Therefore, these parameters were set to 5 for both non-improving counter and 5-fold cross validation for clusters evaluation.

Nevertheless, the largest three datasets in terms of the number of labels: CAL500, Language log and Enron, LPBR takes hours to execute and then reports out of memory problem (as shown in Table 1). We report the results on others at least to see how it performs but we did not include its results in the significant test.

Considering ensemble models, all methods involve bootstrap sampling the training set. Ensemble iterations are set to 10. The predictions are averaged per label to produce the classifications confidence values.

We evaluate the algorithms on 9 commonly used benchmarks from the Meka and Mulan repositories. We used the most common multi-label metrics, namely multi-label

[1] http://meka.sourceforge.net/
[2] http://mulan.sourceforge.net/



(a) exact-match (b) hamming loss (c) f-measure

Fig. 1: Critical Difference diagrams using pairwise comparisons for those experiments where the Friedman test gives significance at 0.05. It considers only single classifier algorithms.

accuracy (Jaccard index), exact-match (subset accuracy), hamming loss and micro averaged f-measure [5,7,12]. Tables 1 and 2 show the average 10-fold cross validation of the single classifier and ensembles, respectively.

Table 1 also shows the average rank of each approach. Both **LaCova** and **LaCova-CLus** have the best average rank in terms of the multi-label accuracy. **LaCova** wins in five datasets out of nine, whereas **LaCova-CLus** wins in two datasets out of nine. In relation to exact-match, **LaCova-CLus** has the best average rank followed by LP, which suggests that clustering labels into dependent subsets may improve the exact-match. CC has the best average rank for both hamming loss and f-measure. **LaCova-CLus** and **LaCova** have the second best average rank for hamming loss and f-measure, respectively.

We conducted the Friedman test based on the average ranks for all datasets in order to verify whether the differences between algorithms are statistically significant [4]. For exact-match, hamming loss, f-measure the Friedman test gave a significant difference at 5% confidence so we proceed to a post-hoc analysis based on Nemenyi statistics as shown in Figure 1. Regarding the ensemble models, there is no significant difference between the algorithms.

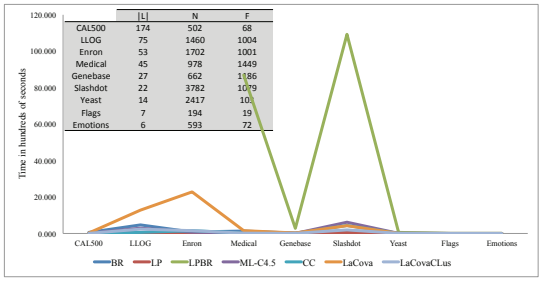


Fig. 2: Comparison of build time in hundreds of seconds. The table shows statistics of the datasets used in the experiments. $|L|$ is the number of labels, F is the number of features and N is the number of examples.

Table 1: Average 10-fold cross validation on 9 datasets comparing single classifier algorithms. The lower value of hamming loss is the better, and for the other metrics the higher value is the better.

	BR	LP	LPBR	ML-C4.5	CC	**LaCova**	**LaCova-CLus**
Multi-label Accuracy							
CAL500	0.201	0.204	-	0.225	0.208	**0.226**	0.206
LLOG	0.245	0.234	-	0.208	**0.247**	0.100	0.243
Enron	**0.296**	0.277	-	0.344	**0.315**	**0.315**	0.294
Medical	0.729	0.724	0.725	0.541	0.728	0.548	**0.736**
Genebase	0.901	0.980	0.955	0.922	**0.986**	0.918	0.982
Slashdot	0.425	0.420	0.376	0.312	0.429	0.444	**0.465**
Yeast	0.391	0.396	0.411	0.414	0.218	**0.431**	0.397
Flags	0.536	0.550	0.541	0.527	0.531	**0.574**	0.565
Emotions	0.402	0.416	0.417	0.411	0.388	**0.425**	0.405
average rank	4.3	4.1	-	4.1	3	**2.6**	2.8
exact-match							
CAL500	0	0	-	0	0	0	0
LLOG	0.184	0.205	-	0.172	0.203	0.063	**0.211**
Enron	0.022	**0.079**	-	0.064	0.076	0.058	0.077
Medical	0.607	0.642	0.655	0.389	0.641	0.393	**0.661**
Genebase	0.809	0.962	0.937	0.882	**0.971**	0.844	0.968
Slashdot	**0.425**	0.368	0.306	0.064	0.356	0.351	0.411
Yeast	0.035	**0.134**	0.122	0.096	0.118	0.120	0.122
Flags	0.098	0.139	0.145	0.108	0.150	0.160	**0.187**
Emotions	0.125	0.202	0.165	0.165	0.157	**0.219**	0.187
average rank	4.7	2.3	-	4.7	3.2	3.9	**1.9**
hamming loss							
CAL500	0.223	0.201	-	0.213	**0.189**	0.214	0.201
LLOG	0.031	0.026	-	0.033	**0.023**	0.035	0.025
Enron	0.082	0.078	-	0.084	**0.069**	0.075	0.076
Medical	0.013	0.014	0.017	0.025	**0.011**	0.025	0.013
Genebase	0.008	0.002	0.019	0.014	**0.001**	0.011	0.002
Slashdot	0.055	0.058	0.086	0.084	**0.045**	0.057	0.054
Yeast	0.296	0.288	0.363	0.297	0.307	**0.272**	0.285
Flags	0.317	0.297	0.309	0.334	0.295	**0.281**	0.285
Emotions	0.296	0.304	0.309	0.323	0.305	**0.287**	0.295
average rank	4.1	3.6	-	5.6	**1.7**	3.3	2.3
f-measure							
CAL500	0.334	0.332	-	0.363	0.338	**0.364**	0.334
LLOG	0.175	0.122	-	0.103	**0.177**	0.121	0.136
Enron	**0.428**	0.364	-	0.426	0.014	0.421	0.391
Medical	0.778	0.744	0.773	0.587	**0.786**	0.618	0.758
Genebase	0.921	0.982	0.853	0.870	**0.988**	0.896	0.981
Slashdot	0.506	0.429	0.381	0.416	**0.512**	0.476	0.473
Yeast	0.541	0.522	0.531	0.546	0.528	**0.560**	0.526
Flags	0.686	0.689	0.686	0.674	0.677	**0.711**	0.707
Emotions	0.539	0.521	0.540	0.523	0.499	**0.544**	0.527
average rank	2.7	4.5	-	4.6	**2.6**	2.7	3.6

5.1 Summary

The general conclusion of these experiments, which compare **LaCova** and **LaCova-CLus** with other different algorithms, are summarised in the following points:

- There is no algorithm that performs well in all evaluation measures. Multi-label classifiers can be selected depending on the dataset and the desired evaluation metrics.
- Classifiers that transform multi-label problems into several binary problems are good for both hamming loss and f-measure. A good example for this is binary relevance approach.

Table 2: Average 10-fold cross validation on 9 datasets comparing ensemble algorithms (10 iterations).

Multi-label Accuracy			
	ECC	**LaCova**	**LaCova-CLus**
CAL500	0.282	**0.288**	0.249
LLOG	**00.281**	0.084	0.084
Enron	**0.388**	0.384	0.372
Medical	**0.773**	0.439	0.744
Genebase	**0.974**	0.919	0.972
Slashdot	0.466	0.446	**0.484**
Yeast	0.505	**0.519**	0.496
Flags	0.565	0.579	**0.587**
Emotions	0.493	**0.500**	0.466
average rank	**01.66**	2.05	2.27

Exact-match			
	ECC	**LaCova**	**LaCova-CLus**
CAL500	0	0	0
LLOG	**0.199**	0.003	0.003
Enron	0.047	0.029	**0.055**
Medical	**0.671**	0.269	0.635
Genebase	**0.953**	0.844	0.949
Slashdot	0.320	0.341	**0.359**
Yeast	0.124	**0.159**	0.120
Flags	0.135	0.129	**0.187**
Emotions	**0.226**	0.261	**0.226**
average rank	**1.83**	2.27	1.88

hamming loss			
	ECC	**LaCova**	**LaCova-CLus**
CAL500	0.209	**0.167**	0.193
LLOG	**0.033**	0.037	0.037
Enron	0.068	**0.060**	0.071
Medical	**0.011**	0.025	0.013
Genebase	**0.002**	0.014	0.003
Slashdot	**0.051**	0.061	0.056
Yeast	0.239	**0.213**	0.245
Flags	0.294	0.271	**0.263**
Emotions	0.258	**0.234**	0.254
average rank	**01.88**	1.94	2.16

f-measure			
	ECC	**LaCova**	**LaCova-CLus**
CAL500	0.435	**0.442**	0.394
LLOG	**0.219**	0.126	0.126
Enron	0.519	**0.529**	0.502
Medical	**0.813**	0.570	0.772
Genebase	**0.980**	0.873	0.968
Slashdot	**0.548**	0.523	0.512
Yeast	0.634	**0.648**	0.627
Flags	0.707	0.719	**0.726**
Emotions	0.610	**0.622**	0.590
average rank	**1.66**	1.83	2.5

- There are some proposed solutions which combines binary relevance because of its effectiveness for the above mentioned metrics. To name a few, classifier chain and **LaCova**. We can see from the experiments that these methods have good results for hamming loss and f-measure.
- Exact-match is a strict measure. Considering correlation between labels can get higher exact-match. **LaCova-CLus** and LP achieve better exact-match. CC also considers label correlation, however, it depends on the labels order. For that reason, they propose ensemble of classifier chain thats tries different label orders.
- **LaCova** and **LaCova-CLus** have better multi-label accuracy among others.
- In case of high dimensional space in terms of number of labels, features, examples, **LaCova-CLus** is much faster than **LaCova**. Figure 2 shows the build time for all algorithms across datasets.

6 Conclusion

In this paper we have presented a novel algorithm for multi-label classification called **LaCova-CLus**. The key idea of this algorithm is to compute label covariance matrix at each node of the tree in order to measure the correlation and cluster labels dynamically. It interpolates between two well-known multi-label classifiers: LP and BR by introducing four splitting ways that enables local decisions.

To evaluate **LaCova-CLus** we first compared it to baseline and other state-of-the-art approaches. Then, we evaluated its ensemble to other ensemble methods. We used four common evaluation metrics and nine datasets. Experimental results indicate that it outperforms these methods for exact-match on the average rank.

Acknowledgment

Reem is a PhD student who is sponsored by King Abdulaziz University, Saudi Arabia. This work was partly supported by the REFRAME project granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences and Technologies ERA-Net (CHISTERA), and funded by the Engineering and Physical Sciences Research Council in the UK under grant EP/K018728/1.

References

1. Al-Otaibi, R., Kull, M., Flach, P.: Lacova: A tree-based multi-label classifier using label covariance as splitting criterion. In: Proceedings of the IEEE 13th International Conference on Machine Learning and Application (ICMLA-2014). pp. 74–79. IEEE, Detroit, USA (December 2014)
2. Chekina, L., Gutfreund, D., Kontorovich, A., Rokach, L., Shapira, B.: Exploiting label dependencies for improved sample complexity. Machine Learning 91(1), 1–42 (Apr 2013)
3. Clare, A., Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: In: Lecture Notes in Computer Science. pp. 42–53. Springer (2001)
4. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (Dec 2006)
5. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: Proceedings of the 14th ACM international conference on Information and knowledge management. pp. 195–200. CIKM '05, ACM, New York, NY, USA (2005)
6. Gjorgjevikj, D., Madjarov, G., Dzeroski, S.: Hybrid decision tree architecture utilizing local svms for efficient multi-label learning. IJPRAI 27(7) (2013)
7. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 22–30. Springer (2004)
8. Luaces, O., Díez, J., Barranquero, J., del Coz, J.J., Bahamonde, A.: Binary relevance efficacy for multilabel classification. Progress in AI 1(4), 303–313 (2012)
9. Read, J., Martino, L., Luengo, D.: Efficient monte carlo optimization for multi-label classifier chains. In: Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2013)
10. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II. pp. 254–269. ECML PKDD '09, Springer-Verlag, Berlin, Heidelberg (2009)
11. Rokach, L., Maimon, O.: Data Mining with Decision Trees: Theory and Applications. World Scientific Publishing Co., Inc., River Edge, NJ, USA (2008)
12. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. In: Machine Learning. pp. 297–336 (1999)
13. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In: ECML/PKDD 2008 Workshop on Mining Multidimensional Data (2008)
14. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Mach. Learn. 73(2), 185–214 (Nov 2008)

Yet Another Tool for Time Series Visualization and Analysis

Ilseyar Alimova

Kazan Federal University, Kazan, Russia,
AlimovaIlseyar@gmail.com
http://kpfu.ru/eng

Abstract. We present a computer software system for time series visualization and analysis. The visualization capabilities allow for plotting time series and applying various transformations on the plot. The time series analysis capabilities support several similarity measures such as Pearson correlation coefficient and a new measure based on moving approximation transformation. Plots of two time series can be combined on one graph in such a way that the chosen similarity measure between them is minimized. The system can be used for processing Google Books Ngram data.

Keywords: Time series, measures of association

1 Introduction

Significant volume of information in different domains is stored as time series – for example, prices of stocks of different companies in finance [1] and variation of temperature in meteorology [2]. Very often, there is a need to identify dependence between time series in systems in order to make a decision about their operations. For example, in the gas and oil domain, identifying possible relationships between wells can help figure out a possible location of new oil and injectors and increase the oil recovery [3].

Visualization plays an important role in time series analysis. Graphical representation of data sometimes allows the identification of relationships before the calculation of the measures of association, which requires tremendous computational resources when large volume of data comes to the input.

In this article, we describe a system that combines visualization of time series and methods of numerical analysis of the association measures. The system represents time series in the form of graphics and includes more advanced tools for data analysis, such as overlaying graphics on the same coordinate plane and putting message labels at a certain point of graphic or parallel displacement of graphic relative to the y-axis. Such transformations contribute to the optimization of finding useful relationships between the time series.

In the following section, we represent related work on time series visualization and motivation to develop our own system. In section 3, the system's architecture and main functions are described. Section 4 elaborates preliminary experiments with Google Books Ngram data. Section 5 gives an outlook for further work.

2 Background and Related Work

2.1 Association Measures for Time Series Analysis

One of the most popular association measures is the Euclidean distance and Pearson correlation coefficient. However, time series need to be processing before this measure is used. The values of time series have to be smooth and normalized, and all accidental releases of data have to be removed. Moreover, these measures can define only a linear relationship. In the papers [4–6] was described the measure of associations between time series, which is called Moving Approximation Transform and can work with raw values of time series. In [6], described the derivation of the formula to calculate this measure. The advantages of this measure of association were the reason for using it in the system.

2.2 Tools for Visual Analysis of Time Series

There are a lot of programming tools for visualization and analysis of time series. They represent data in the traditional form of line charts and using alternative methods. Below is a brief overview of such systems:

- TimeSearcher [7] represents data in the form of line charts and allows users to analyze time series by creating queries. The queries are formed with time boxes. The time boxes are rectangles that highlight the regions of interest to the user. In the second version of this system, developers added the ability to analyze long time series, and in the next version appeared tools for the prediction of future behavior. However, this program does not provide numerical estimates of the degree of similarity.
- KronoMiner [8] represents time series as a line chart and places them on a circle. The users can highlight the region of interest and get it enlarged image. If the user overlaps highlighted regions on each other, the program will change the color of the background depending on the degree of similarity. At the center of the circle system shows the connection between similar regions. Kronominer is great for analysis of a small number of time series, but not useful when it is necessary to analyze a large number of time series.
- One of the alternative methods of representing time series is spiral [9]. Weber et al. developed a tool, in which each periodic section of time series represents as a ring of spiral and the values of time are characterized by colors and thickness of line. The time period is set in advance by the user. In case the user cannot determine the time period, the program has a tool to help identify it. This tool animates the spiral by continuously changing the cycle length, and the user can stop the animation when the period is spotted. This system is useful for data visualization, with a clearly defined period, but it is difficult to use without periodical time series.
- The VizTree [10] visualization system represents time series as an augmenting suffix tree. It splits the original series on the segments, marks each segment with a symbol of a given alphabet using the special algorithm and

builds string suffix tree using that symbols. This suffix tree allows the extracting of repeated patterns, the calculation of the frequency of occurrence of interesting pattern and the discovery of anomaly patterns. The program is suitable for the analysis of long time series, but it is impossible to analyze several time series and count the measures of association using it.

So the main differences of our system from all these tools lie in using a new measure of association and integrating with Google Ngram.

3 System Description

The system is a desktop application that can work with data from files on a computer and with data on a remote server. It aims to automate data conversion and graphics of time series and provides an interface to make necessary changes manually.

3.1 System Architecture

The main components of the system include a graphical interface to display data and a data converter that performs calculations. The modular system architecture makes it scalable and expandable for a variety of input data.

3.2 Interface and Functionality

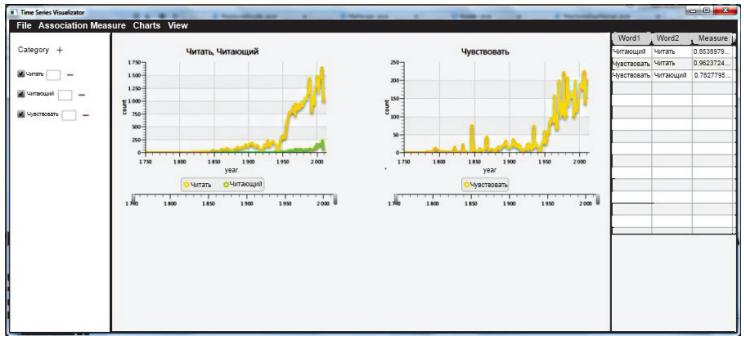


Fig. 1. Main program screen

Figure 1 shows the interface of the main program screen. The screen has three panes. The left pane displays the names of time series and shows to what category they belong. It is possible to hide a graphic or show it again with

checkboxes on the left side of a name or remove a chart by pressing minus. The field on the right side of a name is intended for the coefficient on which the system multiplies the value of the time series.

The central panel shows graphics and place them in the same order as the names on the left pane of the screen. The following are basic functions to work with graphics:

- Comparison charts on the same coordinate plane. It can be done with a mouse by a sample transfer of the coordinate plane with one chart to another coordinate plane with a second chart or by selecting one of the options in the menu on the top of the screen. Graphics can be also combined by categories. Combining graphs on one coordinate plane allows one to see the degree of similarity of these graphs and extract intervals of joint increase and decrease.
- Attachment of a text event to a point of graphic. In time series analysis, sometimes it is necessary to identify relationships between the growth of values or reduction of the values and the events that happened in the world. For example, a decrease in sales of Samsung devices can be caused by release of a new device from the Apple company. So, this function helps one remember such events.
- Parallel shift of a graphic along the y-axis. The values of time series may not match with each other, but they still depend on each other. This function puts graphs of time series on each other. It increases the value of the lower chart by multiplying them on a coefficient. Coefficient may be selected manually or automatically.
- Parallel shift of graphic along the x-axis. Sometimes, after this shift, the values of a graphic, which is deemed independent, begin to coincide with each other. The function is useful in cases where events related to the first graphic affect the second graphic with delay, and the delay time period repeats over time. Shift interval is selected manually or automatically.
- Average. This function lets down all charts of coordinate plane so that the x-axis becomes a medium line of charts.
- Marking certain time points in several charts with a line. To apply this function, the graphics must be placed under each other and the vertical line needs to be moved on the time point.
- Zoom. The interval that is displayed on a coordinate plane can be changed using clips under the graphic. The displayed interval can be moved along the x-axis.

The right pane displays a table that contains the following columns: the name of the first time series, the name of the second time series, and the association measure. It is possible to select similarity measures, which will be displayed on the table using the menu on the top of the screen.

3.3 Data Converter

This component of the system helps the graphic module and calculates values for data conversion. The main functions are as follows:

- Calculation of the coefficient for the parallel transfer graphic along the y-axis. This function takes each value of the coefficient, multiplies time series values on it, and count the association measure for new values. The association measure is selected optionally. Measure counting stops when graphics values no longer intersect. From the obtained values, the system chooses the minimum and the coefficient that corresponds to it. This coefficient is used to overlay the graphics' values with each other.
- Calculation of the interval for the parallel transfer graphic along the y-axis. On each iteration, the values of the graphic, which is going ahead, is shifted by a one-time measure to the right, and after that, the system calculate the measure of association. These iterations continue until the graphics are intersected. From the obtained values of the association measures, the system chooses the minimum and then moves the graphic to the appropriate time interval.
- The calculation of new values of time series to make x-axis their medium line is made using the following formula [11]:

$$F(x)_i = \frac{x_i - \overline{x}}{\sqrt{\sum_{i=1}^{n}(x_j - \overline{x})^2}}$$

- The calculation of the measures of association. This function calculate the measures of associations by standard formulas such as the moving approximation transform measure for x and y time series, which is calculated using the following formula [6]:

$$AM(y,x) = \frac{1}{|K|}\sum_{k\in K} coss_k(y,x)$$

Here $coss_k(y,x)$ is the cosine of the angle between vectors of moving approximation transform of y and x. K is the set of all possible windows, $K \subseteq \{2,\cdots n\}$.

4 Working with Google Books NGram Dataset

The system has additional functionality that can work with Google Ngram data [12]. Google Ngram Viewer is a service from Google Inc. that provides information about how much the word is used in print in a given year. This system has its own service to visualize data, but it represents all graphics on the same chart.

Working with this module, one could use the following scenario: the user inputs a word, the system generates the necessary request to a server, and the server returns as the resulting time series values, showing the frequency of use of the word for each year. Then these values are displayed as graphics, and after that, all the functions described in the previous chapter can be used with these graphics.

Google does not provide any built-in API to work with Ngram dataset; therefore, a server that gives necessary functions to system, has been developed. The

server stores all Russian n-grams and the frequency of each n-gram. The source files, provided by Google, have the following information: word, year, number of usages this year, number of pages and number of books. So it was necessary to calculate the frequency for each word in each year. For this purpose, it was used a file that contains the total number of processed data words. Thus, the frequency calculated by the standard formula is the number of occurrences of words in a given year divided by the total number of words for this year. The data obtained were stored in the server database. Then requests, which receive the input words and return the required data, were developed.

5 Conclusion and Future Work

The developed system has a convenient functionality for the display and analysis of time series and the calculation of measures of association. Now, the system is configured to work with Google Ngram. In the future, we plan to expand this functionality to work with data obtained from Google Finance. The Google Finance service provides access to financial information of many companies. This information makes it possible to analyze stock prices and exchange rates for a decision about their sales. The next step is planned for the implementation of the ability to display three-dimensional graphs in the coordinate plane. This allows taking into account two time-dependent indicators at once.

6 Acknowledgments

This study was financially supported by the Russian Foundation for Basic Research (project 15-01-06456) and by the subsidy of the Russian Government to support the Program of competitive growth of Kazan Federal University among world class academic centers and universities.

We thank Ildar Batyrshin for assistance with formulation of the problem, and Vladimir Ivanov for comments that greatly improved the manuscript.

References

1. Daniel A Keim, Tilo Nietzschmann, Norman Schelwies, Jörn Schneidewind, Tobias Schreck, and Hartmut Ziegler. A spectral visualization system for analyzing financial time series data. 2006.
2. Richard H Jones. Spectral analysis and linear prediction of meteorological time series. *Journal of Applied Meteorology*, 3(1):45–52, 1964.
3. Ildar Batyrshin. Up and down trend associations in analysis of time series shape association patterns. In *Pattern Recognition*, pages 246–254. Springer, 2012.
4. I Batyrshin, R Herrera-Avelar, L Sheremetov, and R Suarez. Moving approximations in time series data mining. In *Proceedings of International Conference on Fuzzy Sets and Soft Computing in Economics and Finance, FSSCEF 2004, St. Petersburg, Russia*, volume 1, pages 62–72, 2004.

5. Ildar Batyrshin, Raul Herrera-Avelar, Leonid Sheremetov, and Aleksandra Panova. Association networks in time series data mining. In *Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American*, pages 754–759. IEEE, 2005.
6. Ildar Batyrshin, Raul Herrera-Avelar, Leonid Sheremetov, and Aleksandra Panova. Moving approximation transform and local trend associations in time series data bases. In *Perception-based Data Mining and Decision Making in Economics and Finance*, pages 55–83. Springer, 2007.
7. Harry Hochheiser and Ben Shneiderman. Interactive exploration of time series data. In *Discovery Science*, pages 441–446. Springer, 2001.
8. Jian Zhao, Fanny Chevalier, and Ravin Balakrishnan. Kronominer: using multi-foci navigation for the visual exploration of time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1737–1746. ACM, 2011.
9. Marc Weber, Marc Alexa, and Wolfgang Müller. Visualizing time-series on spirals. In *Infovis*, page 7. IEEE, 2001.
10. Jessica Lin, Eamonn Keogh, and Stefano Lonardi. Visualizing and discovering non-trivial patterns in large time series databases. *Information visualization*, 4(2):61–82, 2005.
11. Ildar Batyrshin and Valery Solovyev. Positive and negative local trend association patterns in analysis of associations between time series. In *Pattern Recognition*, pages 92–101. Springer, 2014.
12. Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182, 2011.

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, classical 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 $S(t)$:

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

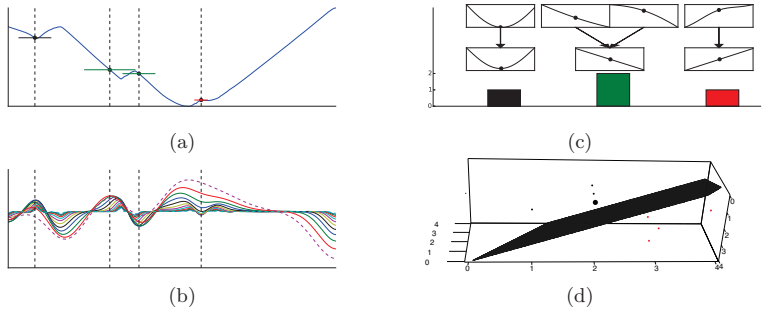


Fig. 1: Approach overview : (a) A time series and its extracted keypoints (the length of the horizontal lines for each point is proportional to the keypoint scale), (b) The Difference-of-Gaussians, computed at different scales, on which the keypoint extraction is built, (c) Keypoint description is based 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.

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

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

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.* 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 $\{2^i, \forall\ i \in \{2..10\}\}$ 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].

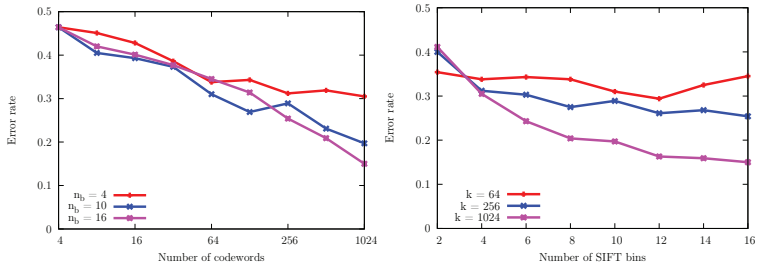


Fig. 2: Classification accuracy on dataset Yoga as a function of k and n_b.

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.

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).

	ED+ 1NN	DTW+ 1NN	TSBF[2]	SAX-VSM[12]	SMTS[1]	BoP[9]
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.

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.

Reducing Bit Error Rate of Optical Data Transmission with Neighboring Symbol Information Using a Linear Support Vector Machine

Weam M. Binjumah[1], Alexey Redyuk[2], Neil Davey[1], Rod Adams[1], Yi Sun[1]

[1]The School of Computer Science, University of Hertfordshire, Hatfield, AL10 9AB, UK, weam.m.j@gmail.com, [n.davey, R.G.Adams, y.2.sun]@herts.ac.uk.
[2]Novosibirsk State University, Novosibirsk, 2 Pirogova Street, 630090, Russia, alexey.redyuk@gmail.com.

Abstract. Improvement of bit error rate in optical transmission systems is a crucial and challenging problem. Whenever the distance travelled by the pulses increases, the difficulty of it being classified correctly also increases. We apply a linear support vector machine for classifying Wavelength-Division-Multiplexing Non-Return-to-Zero Dual-Polarization Quadrature-Phase-Shift-Keying (WDM NRZ DP QPSK) signals with neighboring information. We demonstrate a bit error rate (BER) improvement in comparison with the traditional threshold method.

Key words: Support Vector Machine (SVM), Classification, Machine Learning, Bit Error Rate (BER), Signal Processing, Optical Data Transmission.

1 Introduction

In optical communication systems, there are many different causes of loss in the quality of the signal [1]. Increasing the distance travelled by the pulses leads to an increase in the number of bit errors. As is normally the case the phase is measured at the mid point of the pulse because that represents the highest power level. Both linear and nonlinear distortions can be present in the signal. Pulse linear distortion can be modeled, and therefore factor it out. The same is not true for non-linear distortion. And so, we are using machine learning technique to detect and correct such distortions. Metaxas et al. demonstrates that linear Support Vector Machines (SVM) outperformed other trainable classifiers for error correction in optical data transmission, such as using neural networks [2].

In this paper, we investigate the most significant samples that can be used for training the linear SVM classifier to reduce the number of bit errors during the classification process. In particular, we take into account the neighboring information from each symbol.

2 Motivation

There is an on going need to increase global bandwidth. Due to its capacity and speed at long distances optical links are currently used and probably will also be used in the foreseeable future. The greater distance signals travel the more likely noise will corrupt the signal, giving rise to an ever increasing bit error rate (BER). Errors can be dealt with by adding check bits to the signal, but this uses bandwidth. In our work, we use an alternative approach in which we train a machine learning system to automatically detect and correct errors.

3 Background

3.1 Related Work

One technique that can be used to reduce the effect of signal distortion is using machine learning systems. In earlier works, we demonstrated the possibility of using simple artificial neural networks to help error correction and detection accurately at a high speed [3]. Since the system is trainable, it could cope with the change over the time of a channel's characteristics. The decoder used a perceptron which can classify at high speed. In fact, it could be built in hardware and give real time error correction even at bit rates of over 100 GHz. One problem of using a perceptron is to regularize the decision boundary to avoid over/under fitting. [2] demonstrated the efficiency of a trainable classifier at improving the bit error rate. It is known that a support vector machine (SVM) is a better classifier than perceptron. In the work reported here, we show how a linear SVM can be used to perform error detection and correction.

3.2 Computational Model

Fig. 1 shows the link configuration under investigation. In the numerical model we simulated a typical non-return-to zero (NRZ)-DP-QPSK transmitter which consisted of a laser, modulated at 30 Gbaud using a 2^{15} pseudorandom binary sequence (PRBS) and filtered by a 2^{nd} order super Gaussian filter with 30 GHz 3 dB bandwidth. The signal channel at 1550 nm was propagated over the fiber along with 10 50 GHz spaced similar crosstalk channels, with decorrelated PRBS sequences. In order to model signal propagation over the nonlinear fiber a system of coupled nonlinear Schrödinger equations (CNLSE) was used. CNLSE has been solved using the split-step Fourier method [4]. After each erbium doped fiber amplifier (EDFA), the signal was noise loaded with the white Gaussian noise, calculated using a 6 dB amplifier noise figure. At the receiver side the signal was filtered by a 2^{nd} order super Gaussian filter with 30 GHz 3 dB bandwidth. The chromatic dispersion was fully compensated by multiplying the Fourier transformed optical field with the reverse dispersion function. For phase estimation, an algorithm based on the 4^{th}-power Viterbi–Viterbi method has been used. The effects of signal digitization, polarization rotation and PMD have not been considered in the simulations.

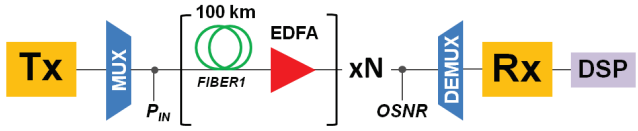
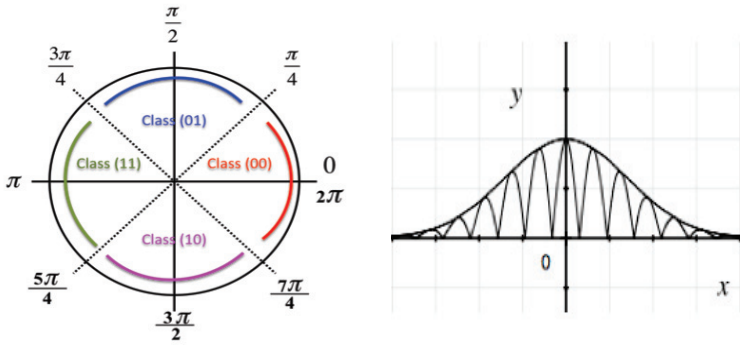


Fig. 1: The fiber link configuration.



(a) Four different variant of the phase for PI/4 QPSK modulation with Gray coding.

(b) Gaussian carrier wave superimposed on an sinusoidal light wave

Fig. 2: PI/4 QPSK modulation and Gaussian carrier wave.

The data we are analyzing consists of 32,768 symbols. Our data was generated by a dual-polarization optical communication system, X and Y polarization. The simulation process was repeated 10 times with different random realizations of the amplified spontaneous emission (ASE) noise and input PRBS, each run generates 32,768 symbols. The signal was detected at intervals of 1,000 km to a maximum distance 10,000 km.

Each pulse was decoded into one of four symbols; see Fig. 2(a), according to its phase. Each data point has a corresponding two-bit label for each run. Each run generates one data set. Fig. 2(b) shows a Gaussian carrier wave superimposed on an sinusoidal light wave. In this paper we focus on X-Polarization data and use Y-Polarization data for verification of our results. Each pulse is represented by 64 equally spaced phase samples. Fig. 3 shows the phase of central sample of one of the data sets at 10,000 km. As we can see from Fig. 3, the phase of some pulses is different from their actual encodings (provided by labels). This means these signals were distorted after traveling 10,000 km.

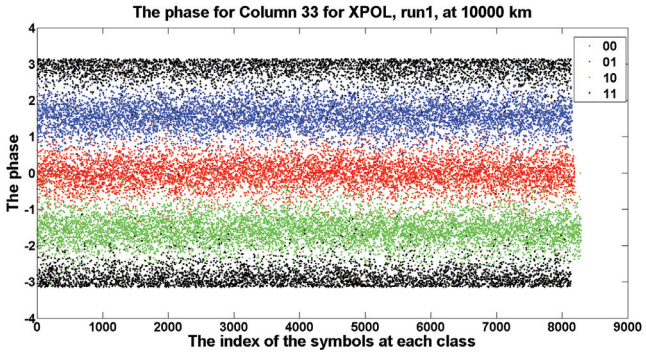


Fig. 3: The phase of the sample 33, XPOL, run1, at 10,000 km.

4 Machine Learning Decoder

In this work, we used a trainable classifier to help decode the received pulses. Note that the data used in this work is simulated data. Since this decoding must take place in real time, the classifier must be capable of being hardware based. To this end, we used a linear support vector machine (SVM) [2]. SVM is a soft maximum margin classifier, see Fig. 4. It has only one non-learnable parameter, which is the regularizing cost parameter C [2]. This parameter allows the cost of misclassification to be specified. The Linear SVM is arguably the most successful method in machine learning.

In order to have a continuously varying phase value, both the Sine and Cosine of the phase were used as input to our decoding (The phase angle has a discontinuity at 0 / 2π).

We had used a variety of inputs to our decoder as can be seen in Table 1. From Fig. 5, we see the reason behind using symbols on either side of the symbol being analyzed. Fig. 5 shows three consecutive symbols at 0 km and at 10,000 km. At 10,000 km the middle symbol was incorrectly decoded (the dotted line) when using the threshold method. As we can see from Fig.5, the first symbol has a phase of (π) whereas the phase of the middle symbol is (0) or (2π). However, at 10,000 km the central symbol has been degraded. At the distance of 10,000 km, the first symbol tries to pull the second symbol from (2π) to (π), which led to the prediction of the middle symbol at the middle point as ($3\pi/2$). From the above observation, our hypothesis is that the neighboring symbols can affect the target symbol, for which we want to predict the label. Therefore, in this work we investigate the effect of using the symbol either side of the target in an attempt to reduce the bit error rate. Table 1 shows a description of some experiments that had done on our data in terms of the samples used and features considered.

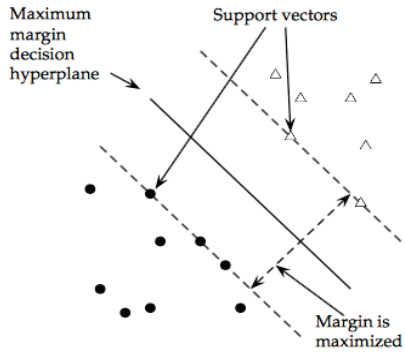


Fig. 4: The margin and support vectors[5].

Table 1: A description of the experiment that were implemented on our data. In the experiment D we used the central sample from the target symbol and symbol either side. In the experiment E we used the central sample from the target symbol and two symbols either side. In the experiment F we used the central sample from the target symbol and two preceding symbols.

Exp.	Method	Sample	Feature's type	Symbol
A	Threshold	Central	Phase value (ϑ)	One symbol
B	Linear SVM	Central	sin(ϑ),cos(ϑ)	One symbol
C	Linear SVM	64 samples	sin(ϑ),cos(ϑ)	One symbol
D	Linear SVM	3 samples	sin(ϑ),cos(ϑ)	Three symbols
E	Linear SVM	5 samples	sin(ϑ),cos(ϑ)	Five symbols
F	Linear SVM	3 samples	sin(ϑ),cos(ϑ)	Three symbols

5 Experiments Setup and Results

In each experiment, we divided each data set into 2/3 training and 1/3 testing. LIBSVM [6] program was used to classify our data sets linearly. We divided the data in the training set into 5 parts and used 5-fold cross validation to find a perfect value for C.

Table 2 shows a comparison between the number of bit errors that obtained from using the threshold method (column A), and the linear SVM using different numbers of samples and symbols (columns B, C, D, E and F, (see Table 1 for details)). Each number in Table 2 from column A to F is the average of the number of bit errors over 10 data sets; and the best result in each distance has been shown in bold font. As we can see from Table 2, the best results obtained so far is from the linear SVM when using 3 samples, the central sample from the target symbol and symbols either side. And also when using 5 samples, the

central sample from the target symbol and two symbols either side. Compared with the result obtained from using the traditional threshold method, the linear SVM has showed a useful improvement when using more than one sample. Especially, when those samples were taken from more than one symbol. For example, the experiment D correctly classified the middle symbol shown in Fig. 5, whereas experiment A, B and C, which did not involve neighboring information misclassified the symbol.

Fig. 6 shows the percentage of the improvement over the threshold method against the distance for the experiments D in Fig. 6 (a) and E in Fig. 6 (b). As can be seen using the neighboring information, an improvement can be obtained from the distance of 2,000 km; with the best improvement was obtained at the distance 3,000 km.

Table 2: How the number of bit errors varies with distance (Note that each number in columns A to F is the average of the number of bit errors over 10 data sets).

Distance	A	B	C	D	E	F
0 km	0	0	0	0	0	0
1,000 km	0	0	0	0	0	0
2,000 km	2.3	3	3.2	2	**1.7**	1.8
3,000 km	16.7	16.3	16.5	**10.6**	11.2	12.9
4,000 km	50.9	50.7	46.9	**37.6**	39.2	40.3
5,000 km	103.7	103.3	98.3	89.9	**88.9**	92.2
6,000 km	185.5	184.9	172.3	165.3	**163.3**	165.2
7,000 km	284.4	284.7	273.1	**260.5**	262.1	262.3
8,000 km	403.3	403.6	386.6	**372.5**	377.9	378.2
9,000 km	533.5	534.2	517.4	511.9	**509.3**	509.9
10,000 km	666.6	670.2	642.1	639.8	**632.4**	634

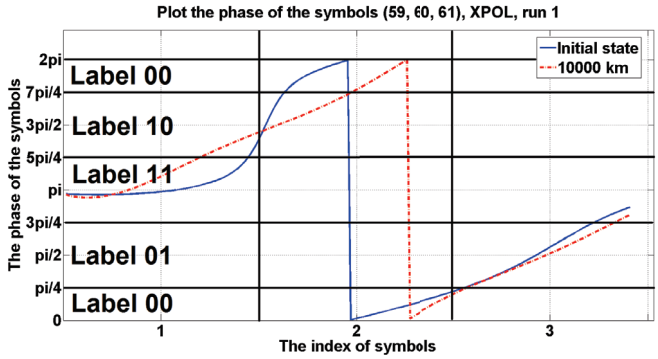
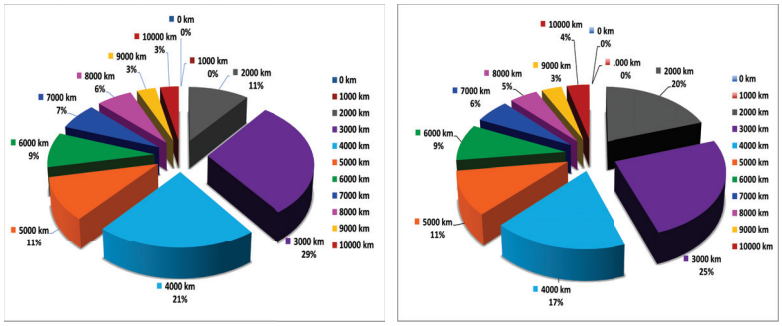


Fig. 5: Three contiguous symbols to show the effect of the symbols either side.



(a) Experiment D. (b) Experiment E.

Fig. 6: The improvement over the threshold method (%). (a) Experiment D: applying linear SVM using 3 samples (the 33^{rd}central sample from the target symbol and symbol either side). (b) Experiment E: applying linear SVM using linear SVM with 5 samples (the 33^{rd}central sample from the target symbol and two symbols either side).

6 Summary and Conclusions

In this work we demonstrated the bit error rate can be reduced using machine-learning techniques. So far, the best results has been obtained for all distances using a linear SVM trained on data from the target symbol with either the symbol either side, or two symbols either side.

We are investigating that how many neighboring symbols should be used as inputs of our machine learning decoder. At the current stage, the target symbol

with three, four, five symbols either side respectively are being investigated. Furthermore, since essentially, the sequence of pulses is time series, we shall apply embedding dimension [7] as a guide to find out a suitable number of neighbors.

We expect that our investigations with nonlinear SVM allow us to obtain further BER improvement along all distances. In addition, features extracted from the signal wave will be investigated in the future work as well. Moreover, we will investigate our methods on different kinds of modulation.

References

1. Bernstein, G., Rajagopalan, B., Saha, D.: Optical network control: architecture, protocols, and standards. Addison-Wesley Longman Publishing Co., Inc. (2003)
2. Metaxas, A., Redyuk, A., Sun, Y., Shafarenko, A., Davey, N., Adams, R.: Linear support vector machines for error correction in optical data transmission. In: Adaptive and Natural Computing Algorithms. Springer (2013) 438–445
3. Hunt, S., Sun, Y., Shafarenko, A., Adams, R., Davey, N., Slater, B., Bhamber, R., Boscolo, S., Turitsyn, S.K.: Adaptive electrical signal post-processing with varying representations in optical communication systems. In: Engineering Applications of Neural Networks. Springer (2009) 235–245
4. Agrawal, G.: Applications of nonlinear fiber optics. Academic press (2001)
5. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Volume 1.
6. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. acm transactions on intelligent systems and technology, 2: 27: 1–27: 27, 2011. Software available at http://www. csie. ntu. edu. tw/cjlin/libsvm (2011)
7. Cao, L.: Practical method for determining the minimum embedding dimension of a scalar time series. Physica D: Nonlinear Phenomena **110**(1) (1997) 43–50

Structure Learning with Distributed Parameter Learning for Probabilistic Ontologies

Giuseppe Cota[1], Riccardo Zese[1], Elena Bellodi[1], Evelina Lamma[1], and Fabrizio Riguzzi[2]

[1] Dipartimento di Ingegneria – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy
[2] Dipartimento di Matematica e Informatica – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy
[giuseppe.cota,riccardo.zese,elena.bellodi,evelina.lamma,
fabrizio.riguzzi]@unife.it

Abstract. We consider the problem of learning both the structure and the parameters of Probabilistic Description Logics under DISPONTE. DISPONTE ("DIstribution Semantics for Probabilistic ONTologiEs") adapts the distribution semantics for Probabilistic Logic Programming to Description Logics. The system LEAP for "LEArning Probabilistic description logics" learns both the structure and the parameters of DISPONTE knowledge bases (KBs) by exploiting the algorithms CELOE and EDGE. The former stands for "Class Expression Learning for Ontology Engineering" and it is used to generate good candidate axioms to add to the KB, while the latter learns the probabilistic parameters and evaluates the KB. EDGE for "Em over bDds for description loGics paramEter learning" is an algorithm for learning the parameters of probabilistic ontologies from data. In order to contain the computational cost, a distributed version of EDGE called EDGE^{MR} was developed. EDGE^{MR} exploits the MapReduce (MR) strategy by means of the Message Passing Interface. In this paper we propose the system LEAP^{MR}. It is a re-engineered version of LEAP which is able to use distributed parallel parameter learning algorithms such as EDGE^{MR}.

Keywords: Probabilistic Description Logics, Structure Learning, Parameter Learning, MapReduce, Message Passing Interface.

1 Introduction

In real world domains the information is often uncertain, hence it is of foremost importance to model uncertainty in representations of the world, including Description Logics (DLs).

In [11, 19, 7, 12] the authors studied the use of probabilistic DLs and various approaches for representing uncertainty in DLs.

Moreover, some works have started to appear about learning the probabilistic parameters or the whole structure of probabilistic ontologies. These are motivated, on one hand, from the fact that specifying the values of the probabilities is

a difficult task for humans and data is usually available that could be leveraged for tuning them, and, on the other hand, from the fact that in some domains there exist poor-structured knowledge bases which could be improved [11, 10]. A knowledge base with a refined structure and instance data coherent with it permits more powerful reasoning, better consistency checking and improved querying possibilities.

In Probabilistic Logic Programming (PLP) various proposals for representing uncertainty have been presented. One of the most successful approaches is the distribution semantics [17]. In [3, 16, 13] the authors proposed an approach to represent probabilistic axioms in DLs called DISPONTE ("DIstribution Semantics for Probabilistic ONTologiEs"), which adapts the distribution semantics for Probabilistic Logic Programming to DLs.

In the field of Probabilistic Inductive Logic Programming the reasoning task is composed by three main issues: 1) *inference*: we want to compute the probability of a query, 2) *parameter learning*: we know the structure (the logic formulas) of the KB but we want to know the parameters (weights) of the logic formulas and 3) *structure learning*: we want to learn both the structure and the parameters.

LEAP [15] for "LEArning Probabilistic description logics" is an algorithm for learning the structure and the parameters of probabilistic DLs following DISPONTE. It combines the learning system CELOE [9] with EDGE [14]. The former, CELOE ("Class Expression Learning for Ontology Engineering"), provides a method to build new (subsumption) axioms that can be added to the KB, while the latter is used to learn the parameters of these probabilistic axioms.

EDGE stands for "Em over bDds for description loGics paramEter learning" and learns the parameters of a probabilistic theory starting from examples of instances and non-instances of concepts. EDGE builds Binary Decision Diagrams (BDDs) for representing the explanations of the examples from the theory. The parameters are then tuned using an EM algorithm [6] in which the required expectations are computed directly on the BDDs. This algorithm is rather expensive from a computational point of view. In order to efficiently manage larger datasets in the era of Big Data, it is crucial to develop approaches for reducing the learning time. One solution is to distribute the algorithm using modern computing infrastructure such as clusters and clouds.

In order to reduce EDGE running time, we developed EDGE$^{\mathrm{MR}}$ [5]. It represents a distributed implementation of EDGE and uses a simple MapReduce approach based on the Message Passing Interface (MPI).

In this paper we present an evolution of LEAP called LEAP$^{\mathrm{MR}}$ which adapts the LEAP algorithm to use EDGE$^{\mathrm{MR}}$. In addition, due to a software re-engineering effort, it was possible to remove the RMI module used by LEAP. Compared with LEAP, the quality of the solutions found with LEAP$^{\mathrm{MR}}$ does not change, the difference consists in the running time which is reduced thanks to EDGE$^{\mathrm{MR}}$ and to the removal of the RMI module to a lesser extent. To the best of our knowledge there are no other algorithms that perform distributed structure learning of probabilistic DLs.

Implementing learning algorithms able to elaborate data in a distributed way paves the way to the development of useful tools for Semantic Web and Data Mining in the context of Big Data.

The paper is structured as follows. Section 2 introduces Description Logics and summarizes DISPONTE. Sections 3 and 4 briefly describe the EDGE and EDGE$^{\mathrm{MR}}$ algorithms. Section 5 presents LEAP$^{\mathrm{MR}}$. Finally, Section 7 draws conclusions.

2 Description Logics and DISPONTE

Description Logics (DLs) are a family of logic based knowledge representation formalisms which are of particular interest for representing ontologies and for the Semantic Web. For an extensive introduction to DLs we refer to [1, 2].

While DLs are a fragment of first order logic, they are usually represented using a syntax based on concepts and roles. A concept corresponds to a set of individuals while a role corresponds to a set of couples of individuals of the domain. For the sake of simplicity we consider and describe $\mathcal{ALC}$, but the proposed algorithm can work with $\mathcal{SROIQ}(\mathbf{D})$ DLs.

We use $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{I}$ to indicate *atomic concepts*, *atomic roles* and *individuals*, respectively. A *role* is an atomic role $R \in \mathbf{R}$. *Concepts* are defined as follows. Each $A \in \mathbf{A}$, $\bot$ and $\top$ are concepts. If C, C_1 and C_2 are concepts and $R \in \mathbf{R}$, then $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$ and $\neg C$ are concepts, as well as $\exists R.C$ and $\forall R.C$.

Let C and D be concepts, R be a role and a and b be individuals, a *TBox* $\mathcal{T}$ is a finite set of *concept inclusion axioms* $C \sqsubseteq D$, while an *ABox* $\mathcal{A}$ is a finite set of *concept membership axioms* $a : C$ and *role membership axioms* $(a, b) : R$. A *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

A KB is usually assigned a semantics using interpretations of the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty *domain* and $\cdot^{\mathcal{I}}$ is the *interpretation function* that assigns an element in $\Delta^{\mathcal{I}}$ to each individual a, a subset of $\Delta^{\mathcal{I}}$ to each concept C and a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each role R. The mapping $\cdot^{\mathcal{I}}$ is extended to all concepts as:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad \bot^{\mathcal{I}} = \emptyset$$
$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$$
$$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \qquad (\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | R^{\mathcal{I}}(x) \subseteq C^{\mathcal{I}}\}$$
$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | R^{\mathcal{I}}(x) \cap C^{\mathcal{I}} \neq \emptyset\}$$

A query over a KB is usually an axiom for which we want to test the entailment from the KB. The entailment test may be reduced to checking the unsatisfiability of a concept in the KB, i.e., the emptiness of the concept.

DISPONTE [3] ("DIstribution Semantics for Probabilistic ONTologiEs") applies the distribution semantics to probabilistic ontologies [17]. In DISPONTE a *probabilistic knowledge base* $\mathcal{K}$ is a set of certain and probabilistic axioms. *Certain axioms* take the form of regular DL axioms. *Probabilistic axioms* take the form $p :: E$, where p is a real number in $[0, 1]$ and E is a DL axiom. A

DISPONTE KB defines a distribution over DL KBs called *worlds* assuming that the axioms are independent. Each world w is obtained by including every certain axiom plus a subset of chosen probabilistic axioms.

For each probabilistic axiom $p :: E$, we decide whether or not to include E in w. The probability of this choice is p if the probabilistic axiom is included in the world, $1 - p$ otherwise. A world therefore is a non probabilistic KB that can be handled in the usual way. By multiplying the probability of the choices made to obtain a world, we can assign a probability to it. The probability of a query is then the sum of the probabilities of the worlds where the query is true.

3 Parameter Learning for Probabilistic DLs

EDGE [14] is a parameter learning algorithm which adapts the algorithm EMBLEM [4], developed for learning the parameters for probabilistic logic programs, to the case of probabilistic DLs under DISPONTE. Inspired by [8], it performs an Expectation-Maximization cycle over Binary Decision Diagrams (BDDs).

EDGE performs supervised parameter learning. It takes as input a DISPONTE KB and a number of positive and negative examples that represent the queries in the form of concept membership axioms, i.e., in the form $a : C$ for an individual a and a class C. Positive examples represent information that we regard as true and for which we would like to get high probability while negative examples represent information that we regard as false and for which we would like to get low probability.

First, EDGE generates, for each query, the BDD encoding its explanations using BUNDLE [16]. For a positive example of the form $a : C$, EDGE looks for the explanations of $a : C$ and encodes them in a BDD. For a negative example of the form $a : \neg C$, EDGE first looks for the explanations of $a : \neg C$, if one or more are found it encodes them into a BDD, otherwise it looks for the explanations of $a : C$, encodes them in a BDD and negates it with the NOT BDD operator. Then, EDGE starts the EM cycle in which the steps of Expectation and Maximization are iterated until a local maximum of the log-likelihood (LL) of the examples is reached. The LL of the examples is guaranteed to increase at each iteration. EDGE stops when the difference between the LL of the current iteration and that of the previous one drops below a threshold ϵ or when this difference is below a fraction δ of the previous LL. Finally, EDGE returns the reached LL and the new probabilities π_i for the probabilistic axioms. EDGE's main procedure is illustrated in Alg. 1.

Procedure Expectation takes as input a list of BDDs, one for each example Q, and computes the expectations $P(X_i = x|Q)$ for all the random Boolean variables X_i in the BDD and for $x \in \{0, 1\}$. According to DISPONTE, each variable X_i is associated with the probabilistic axioms E_i and has value 1 if the axiom E_i is included in the world, 0 otherwise.

Function Maximization computes the parameters' values for the next EM iteration by relative frequency.

Algorithm 1 Procedure EDGE.

function EDGE($\mathcal{K}, P_E, N_E, \epsilon, \delta$) ▷ P_E, N_E: positive and negative examples
  Build $BDDs$ ▷ performed by BUNDLE
  $LL = -\infty$
  repeat
    $LL_0 = LL$
    $LL =$ Expectation($BDDs$)
    Maximization
  until $LL - LL_0 < \epsilon \vee LL - LL_0 < -LL_0 \cdot \delta$
  return LL, p_i for all i ▷ p_i: learned probability of the i-th probabilistic axiom
end function

Building BDDs is #P-hard [18]. However, BUNDLE is able to handle domains of significant size. The EM phase, instead, has a linear cost in the number of nodes since the Expectation requires two traversals of the diagrams.

EDGE is written in Java, hence it is highly portable. For further information about EDGE please refer to [14].

4 Distributed Parameter Learning for Probabilistic DLs

In this section we briefly describe a parallel version of EDGE that exploits the MapReduce approach in order to compute the parameters. We called this algorithm EDGE$^{\text{MR}}$ [5].

4.1 Architecture and Scheduling

Like most MapReduce frameworks, EDGE$^{\text{MR}}$'s architecture follows a master-slave model. The communication between the master and the slaves is done by means of the Message Passing Interface (MPI), specifically we use the OpenMPI[3] library which provides a Java interface to the native library.

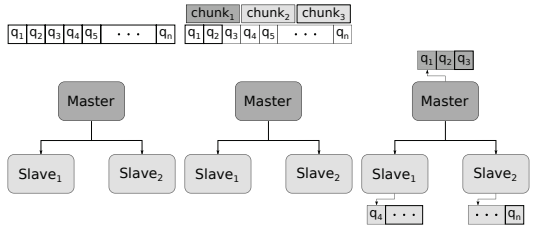
In a distributed context, the performances depend on the scheduling strategy. In order to evaluate different methods, we developed two scheduling strategies: *single-step scheduling* and *dynamic scheduling*. These are used during the queries computation phase.

Single-step Scheduling if N is the number of the slaves, the master divides the total number of queries into $N + 1$ chunks, i.e. the number of slaves plus the master. Then the master begins to compute its queries while, for the other chunks of queries, the master starts a thread for sending each chunk to the corresponding slave. After the master has terminated dealing with its queries, it waits for the results from the slaves. When the slowest slave returns its results to the master, EDGE$^{\text{MR}}$ proceeds to the EM cycle. Figure 1(a) shows an example of single-step scheduling with two slaves.

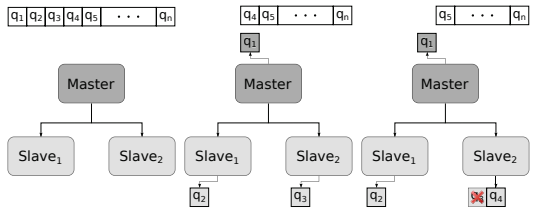
Dynamic Scheduling is more flexible and adaptive than single-step scheduling. Handling each query may require a different amount of time. Therefore, with single-step scheduling, it could happen that a slave takes much more

[3] `http://www.open-mpi.org/`

time than another slave to deal with its chunk of queries. Hence the master and some slaves could be idle. Dynamic scheduling mitigates this issue. The user can establish a chunk dimension, i.e. the number of examples in each chunk. At first, each machine is assigned a chunk of queries in order. Then, if the master ends handling its chunk it just takes the next one, instead, if a slave ends handling its chunk, it asks the master for another one and the master replies by sending a new chunk of queries to the slave. During this phase the master runs a thread listener that waits for the slaves' requests of new chunks and for each request the listener starts a new thread that sends a chunk to the slave which has done the request (to improve the performances this is done through a thread pool). When all the queries are evaluated, EDGE$^{\mathrm{MR}}$ starts the EM cycle. An example of dynamic scheduling with two slaves and a chunk dimension of one example is displayed in Fig. 1(b).



(a) Single-step scheduling



(b) Dynamic scheduling

Fig. 1. Scheduling techniques of EDGE$^{\mathrm{MR}}$.

Experimental results conducted in [5] show that dynamic scheduling has usually better performances than single-step.

It is obvious that for large sizes of the chunk the dynamic scheduling tends to have the same behavior of single-step. Nevertheless the use of chunks containing only one query can introduce a lot of overhead and therefore reduce the speedup. In order to maximize the speedup it is necessary to find an optimal size of the query chunk.

5 Structure Learning with Distributed Parameter Learning

LEAP^{MR} is an evolution of the LEAP system [15]. While the latter exploits EDGE, the first was adapted to be able to perform EDGE^{MR}. Moreover, after a process of software re-engineering it was possible to remove the RMI communication module used by LEAP and therefore reduce some communication overhead.

It performs structure and parameter learning of probabilistic ontologies under DISPONTE by exploiting: (1) CELOE [9] for the structure, and (2) EDGE^{MR} (Section 4) for the parameters. Figure 2 shows the architecture of LEAP^{MR}.

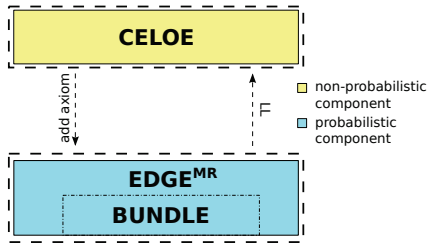


Fig. 2. LEAP^{MR} architecture.

CELOE [9] was implemented in Java and belongs to the open-source framework DL-Learner[4]. Let us consider a knowledge base $\mathcal{K}$ and a concept name `Target` whose formal description, i.e. class description, we want to learn. It learns a set of n class expressions C_i $(1 \leq i \leq n)$ from a set of positive and negative examples. Let $\mathcal{K}' = \mathcal{K} \cup \{C\}$ where $\mathcal{K}$ is the background knowledge, we say that a concept C *covers* an example e if $\mathcal{K}' \models e$. The class expressions found are sorted according to a heuristic. Such expressions can be used to generate candidate axioms of the form $C_i \sqsubseteq$ `Target`.

In order to learn an ontology, LEAP^{MR} first searches for good candidate probabilistic subsumption axioms by means of CELOE, then it performs a greedy search in the space of theories using EDGE^{MR} to evaluate the theories using the log-likelihood as heuristic.

Algorithm 2 shows LEAP^{MR}'s main procedure: it takes as input the knowledge base $\mathcal{K}$ and the configuration settings for CELOE and EDGE^{MR}, then generates $NumC$ class expressions by exploiting CELOE and the sets of positive and negative examples which will be the queries (concept membership axioms) for EDGE^{MR}. A first execution of EDGE^{MR} is applied to $\mathcal{K}$ to compute the initial value of the parameters and of the LL. Then LEAP^{MR} adds to $\mathcal{K}$ one probabilistic subsumption axiom generated from the class expression set at a time. After

[4] `http://dl-learner.org/`

each addition, EDGE$^{\mathrm{MR}}$ is performed on the extended KB to compute the LL of the data and the parameters. If the LL is better than the current best, the new axiom is kept in the knowledge base and the parameter of the probabilistic axiom are updated, otherwise the learned axiom is removed from the ontology and the previous parameters are restored. The final theory is obtained from the union of the initial ontology and the probabilistic axioms learned.

Algorithm 2 Function LEAP$^{\mathrm{MR}}$.

function LEAP$^{\mathrm{MR}}(\mathcal{K}, LP_{type}, NumC, \epsilon, \delta, Schedul)$
  $ClassExpressions =$ up to $NumC$ ▷ generated by CELOE
  $(P_I, N_I) =$ ExtractIndividuals(LP_{type}) ▷ LP_{type}: specifies how to extract (P_I, N_I)
  for all $ind \in P_I$ **do** ▷ P_I: set of positive individuals
    Add ind : Target to P_E ▷ P_E: set of positive examples
  end for
  for all $ind \in N_I$ **do** ▷ N_I: set of negative individuals
    Add ind : Target to N_E ▷ N_E: set of negative examples
  end for
  $(LL_0, \mathcal{K}) =$ EDGE$^{\mathrm{MR}}(\mathcal{K}, P_E, N_E, \epsilon, \delta, Schedul)$ ▷ $Schedul$: scheduling strategy
  for all $CE \in ClassExpressions$ **do**
    $Axiom = p :: CE \sqsubseteq$ Target
    $\mathcal{K}' = \mathcal{K} \cup \{Axiom\}$
    $(LL, \mathcal{K}') =$ EDGE$^{\mathrm{MR}}(\mathcal{K}', P_E, N_E, \epsilon, \delta, Schedul)$
    if $LL > LL_0$ **then**
      $\mathcal{K} = \mathcal{K}'$
      $LL_0 = LL$
    end if
  end for
  return $\mathcal{K}$
end function

6 Experiments

In order to test how much the exploitation of EDGE$^{\mathrm{MR}}$ can improve the performances of LEAP$^{\mathrm{MR}}$, we did a preliminary test where we considered the Moral[5] KB which qualitatively simulates moral reasoning. It contains 202 individuals and 4710 axioms (22 axioms are probabilistic).

We performed the experiments on a cluster of 64-bit Linux machines with 8-cores Intel Haswell 2.40 GHz CPUs and 2 GB (max) memory allotted to Java per node. We allotted 1, 3, 5, 9 and 17 nodes, where the execution with 1 node corresponds to the execution of LEAP, while for the other configurations we used the dynamic scheduling with chunks containing 3 queries. For each experiment 2 candidate probabilistic axioms are generated by using CELOE and a maximum of 3 explanations per query was set for EDGE$^{\mathrm{MR}}$. Figure 3 shows the speedup obtained as a function of the number of machines (nodes). The speedup is the ratio of the running time of 1 worker to the one of n workers. We can note that the speedup is significant even if it is sublinear, showing that a certain amount of overhead (the resources, and thereby the time, spent for the MPI communications) is present.

[5] `https://archive.ics.uci.edu/ml/datasets/Moral+Reasoner`

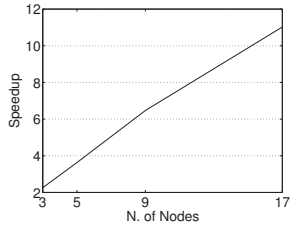


Fig. 3. Speedup of LEAP$^{\mathrm{MR}}$ relative to LEAP for Moral KB.

7 Conclusions

The paper presents the algorithm LEAP$^{\mathrm{MR}}$ for learning the structure of probabilistic description logics under DISPONTE. LEAP$^{\mathrm{MR}}$ performs EDGE$^{\mathrm{MR}}$ which is a MapReduce implementation of EDGE, exploiting modern computing infrastructures for performing distributed parameter learning.

We are currently working for distributing both the structure and the parameter learning of probabilistic knowledge bases by exploiting EDGE$^{\mathrm{MR}}$ also during the building of the class expressions. We would like to distribute the scoring function used to evaluate the obtained refinements. In this function EDGE$^{\mathrm{MR}}$ take as input a KB containing only the individuals and the class expression to test. Finally, the class expressions found are sorted according to the LL returned by EDGE$^{\mathrm{MR}}$ and their initial probability are the probability learned during the execution of EDGE$^{\mathrm{MR}}$.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York, NY, USA (2003)
2. Baader, F., Horrocks, I., Sattler, U.: Description Logics, chap. 3, pp. 135–179. Elsevier Science (2008)
3. Bellodi, E., Lamma, E., Riguzzi, F., Albani, S.: A distribution semantics for probabilistic ontologies. CEUR Workshop Proceedings, vol. 778, pp. 75–86. Sun SITE Central Europe (2011)
4. Bellodi, E., Riguzzi, F.: Expectation Maximization over Binary Decision Diagrams for probabilistic logic programs. Intell. Data Anal. 17(2), 343–363 (2013)
5. Cota, G., Zese, R., Bellodi, E., Lamma, E., Riguzzi, F.: Distributed parameter learning for probabilistic ontologies (2015), to appear
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. J. Roy. Stat. Soc. B Met. pp. 1–38 (1977)
7. Fleischhacker, D., Völker, J.: Inductive learning of disjointness axioms. In: On the Move to Meaningful Internet Systems: OTM 2011, pp. 680–697. Springer (2011)

8. Ishihata, M., Kameya, Y., Sato, T., Minato, S.: Propositionalizing the EM algorithm by BDDs. In: Late Breaking Papers of the International Conference on Inductive Logic Programming. pp. 44–49 (2008)
9. Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering. J. Web Semant. 9(1), 71–81 (2011)
10. Minervini, P., d'Amato, C., Fanizzi, N.: Learning probabilistic description logic concepts: Under different assumptions on missing knowledge. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 378–383. ACM (2012)
11. Ochoa-Luna, J.E., Revoredo, K., Cozman, F.G.: Learning probabilistic description logics: A framework and algorithms. In: Advances in Artificial Intelligence, pp. 28–39. Springer (2011)
12. Riguzzi, F., Bellodi, E., Lamma, E.: Probabilistic Ontologies in Datalog+/-. In: Proceedings of the 9th Italian Convention on Computational Logic, Rome, Italy, June 6-7, 2012. CEUR Workshop Proceedings, vol. 857, pp. 221–235. Sun SITE Central Europe (2012)
13. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R.: Epistemic and statistical probabilistic ontologies. In: Uncertainty Reasoning for the Semantic Web. CEUR Workshop Proceedings, vol. 900, pp. 3–14. Sun SITE Central Europe (2012)
14. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R.: Parameter learning for probabilistic ontologies. In: Faber, W., Lembo, D. (eds.) RR 2013. LNCS, vol. 7994, pp. 265–270. Springer Berlin Heidelberg (2013)
15. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R., Cota, G.: Learning probabilistic description logics. In: Bobillo, F., Carvalho, R.N., Costa, P.C., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) Uncertainty Reasoning for the Semantic Web III, pp. 63–78. LNCS, Springer International Publishing (2014)
16. Riguzzi, F., Lamma, E., Bellodi, E., Zese, R.: BUNDLE: A reasoner for probabilistic ontologies. In: Faber, W., Lembo, D. (eds.) RR 2013. LNCS, vol. 7994, pp. 183–197. Springer Berlin Heidelberg (2013)
17. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: Proceedings of the 12th International Conference on Logic Programming. pp. 715–729. MIT Press (1995)
18. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM J. Comput. 8(3), 410–421 (1979)
19. Völker, J., Niepert, M.: Statistical schema induction. In: The Semantic Web: Research and Applications, pp. 124–138. Springer Berlin Heidelberg (2011)

Chronicles mining in a database of drugs exposures

Yann Dauxais[1], David Gross-Amblard[1], Thomas Guyet[2], and André Happe[3]

[1] Université Rennes-1/IRISA - UMR6074
[2] AGROCAMPUS-OUEST/IRISA - UMR6074
[3] Plateforme PEPS/CHRU Brest

Abstract. Pharmaco-epidemiology is the study of uses and effects of health products (medical devices and drugs) on population. A new approach consists in using large administrative databases to perform such studies on care pathways which contain drugs exposures and medical problems, like hospitalizations. In this context, knowledge discovery techniques becomes mandatory to support clinicians in formulating new hypotheses. Since care-pathways are based on timestamped events and can be complex, we choose a temporal pattern mining approach. In this paper, we adapt existing chronicle mining algorithms in order to mine care-pathways. We present our method to extract all the frequent chronicles and the challenges we encountered. Finally, we present our first experimental results and our perspectives.

Keywords: Sequences mining, temporal data mining, care-pathway

1 Introduction

In classical pharmaco-epidemiology studies, people who share common characteristics are recruited to build a cohort. Then, meaningful data (drug exposures, diseases, etc.) are collected from people of the cohort within a defined period. Finally, a statistical analysis highlights the links (or the lack of links) between drug exposures and adverse effects. The main drawback of cohort studies is the time required to collect the data. Indeed, in some cases of health safety, health authorities have to answer quickly to pharmaco-epidemiology questions.

Using medico-administrative databases is an alternative to classical pharmaco-epidemiology studies. Data is immediately available and it concerns a wide population. Medico-administrative databases have been build primary to ensure health reimbursements. They record with some level of details, for all insured, all drug delivery and all medical procedure. In France, the SNIIRAM national database contains such data for more than 60 millions of insured within a sliding period of 3 years.

The challenges of making pharmaco-epidemiology studies from medico-administrative databases are 1) to abstract the administrative data into meaningful information for a specific study, and 2) to support the clinicians in their analysis of this large amount of data.

This article is focused on the second challenge and deals more specially with the extraction of frequent temporal patterns in a database of care-pathways. In a preliminary step, a dedicated method enables to translate medico-administrative data into patient care-pathways. A care-pathway is a sequence of drug exposures and medical procedures. Each element of the sequence is timestamped and each drug exposure has a time period. We propose to use sequential pattern mining to extract frequent behaviours in the patient care-pathways.

Among all the temporal patterns, *chronicles* [3] appear to be interesting to extract meaningful patterns from timestamped events. A chronicle can be briefly defined as a set of events linked by constraints indicating the minimum and maximum time elapsed between to events. A care-pathway contains point-based events and interval-based events (*e.g.* drug exposures) and a chronicle can express a complex temporal behaviour, for instance: "*The patient was exposed to a drug X between 1 and 2 years, he met his doctor between 400 to 600 days after the beginning of the exposure and, finally, he was hospitalized.*".

In this article, we propose a new algorithm to extract frequent chronicles from a database of sequences of point-based events and interval-based events in which events can be repeated.

2 Events, sequences and chronicles

In this section, we introduce some formal definitions of sequential data, chronicle pattern and of the chronicle mining task.

Definition 1. Let $\mathbb{E}$ be a set of event types and $\mathbb{T}$ a time domain where $\mathbb{T} \subseteq \mathbb{R}$, an **event** is a pair (e, t) where $e \in \mathbb{E}$ and $t \in \mathbb{T}$. We assume that $\mathbb{E}$ is totally ordered and we denote its order by $\leq_{\mathbb{E}}$.

An **event sequence** S is a tuple $\langle SID, \langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n)\rangle\rangle$ where SID is the sequence identifier in the database and $\langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n)\rangle$ is a finite sequence of events. For all i, j, $i < j \Rightarrow t_i \leq t_j$. If $t_i = t_j$, then $e_i <_{\mathbb{E}} e_j$.

In pharmaco-epidemiology studies, a sequence is the care-pathway of a patient identified by SID. A care-pathway consists of point-based events and interval-based events. A point-based event (e, t) represents a medical consultation or a delivery of a drug where e is its event type (consultation or drug name). For drug exposures, which are commonly represented by interval-based events, we use two point-based events (e_s, t_s) and (e_f, t_f) where e_s (resp. e_f) is an event type corresponding to the interval beginning (resp. ending) of an event e.

Example 1 (Database of sequences, $\mathcal{S}$)**.**

SID	sequence
1	$(A_s, 1)$, $(B, 3)$, $(A_f, 4)$, $(C, 5)$, $(B, 20)$
2	$(B, 1)$, $(A_s, 4)$, $(B, 5)$, $(D, 6)$, $(A_f, 8)$, $(C, 9)$
3	$(C, 1)$, $(D, 2)$, $(C, 2)$, $(B, 7)$
4	$(B, 1)$, $(B, 3)$, $(A_s, 7)$, $(A_f, 9)$, $(C, 11)$, $(D, 12)$

The database contains four sequences. There are one type of interval-based event (A) and three types of point-based events (B, C and D).

We will now define the notion of chronicle, which is a pattern of events and a set of temporal constraints. We begin by defining the latter:

Definition 2. A **temporal constraint**, denoted $e_1[t^-, t^+]e_2$, is a tuple where $(e_1, e_2) \in \mathbb{E}$, $e_1 \leq_{\mathbb{E}} e_2$ and $(t^-, t^+) \in \mathbb{T}$, $t^- \leq t^+$. A temporal constraint is satisfied by a pair of events $((e, t_1), (e', t_2))$, $e \leq_{\mathbb{E}} e'$ iff $e = e_1$, $e' = e_2$ and $t^- \leq t_2 - t_1 \leq t^+$. We say that $e_1[a, b]e_2 \subseteq e'_1[a', b']e'_2$ iff $e_1 = e'_1$ and $e_2 = e'_2$ and $[a, b] \subseteq [a', b']$. Hence $\subseteq$ is a partial order on the set of the temporal constraints.

Definition 3. A **chronicle** is a pair $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ such that $\mathcal{E} = \{e_1, \ldots, e_n\}$, $e_i \in \mathbb{E}$, where $\forall i, j$, $1 \leq i < j \leq n$, $e_i \leq_{\mathbb{E}} e_j$; and such that $\mathcal{T}$ is a set of temporal constraints where there is at most one temporal constraint between two events of the chronicle, *i.e.* $\forall e, e' \in \mathcal{E}$, $|\{e[a, b]e' \mid e[a, b]e' \in \mathcal{T}\}| \leq 1$. $\mathcal{E}$ is called a multiset. It is a set of events allowing repetitions.

Example 2. Figure 1 illustrates the chronicle $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ where $\mathcal{E} = \{e_1 = A_s, e_2 = A_f, e_3 = B, e_4 = B, e_5 = C\}$ and $\mathcal{T} = \{e_1[2, 4]e_2, e_1[-4, 2]e_3, e_2[-8, 1]e_3, e_2[1, 2]e_5, e_3[2, 17]e_4, e_4[-15, 8]e_5\}$. (A_s, A_f) can be seen as a pair of events representing an interval event A that starts with event A_s and that finishes with event A_f.

We can notice that the graph is not complete. The lack of arc between two nodes can be interpreted as a $[-\infty, +\infty]$ constraint. But, in most case, a more restrictive constraint can be deduced from the other constraints. For instance, a temporal constraint $A_s[3, 6]C$ can be deduced from constraints between A_s and A_f, and between A_f and C.

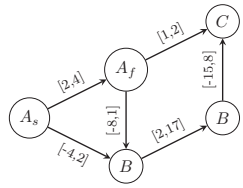


Fig. 1: Chronicle example.

Given two chronicles $\mathcal{C}_1 = (\mathcal{E}_1, \mathcal{T}_1)$ and $\mathcal{C}_2 = (\mathcal{E}_2, \mathcal{T}_2)$, we define the partial order $\preceq$ where $\mathcal{C}_1 \preceq \mathcal{C}_2$ if $\mathcal{E}_2 \subseteq \mathcal{E}_1$ and there is a strictly increasing function f where $\forall i, j$, $1 \leq i < j \leq |\mathcal{E}_2|$, $1 \leq f(i) < f(j) \leq |\mathcal{E}_1|$, $e_i, e_j \in \mathcal{E}_2$, $e_{f(i)}, e_{f(j)} \in \mathcal{E}_1$, $e_{f(i)}[a, b]e_{f(j)} \in \mathcal{T}_1$, $e_i[a', b']e_j \in \mathcal{T}_2$, $e_{f(i)}[a, b]e_{f(j)} \subseteq e_i[a', b']e_j$. If $\mathcal{C}_1 \preceq \mathcal{C}_2$ and $\mathcal{C}_1 \neq \mathcal{C}_2$, we say that $\mathcal{C}_1$ is **more specific** than $\mathcal{C}_2$ or is a child of $\mathcal{C}_2$. On the contrary, $\mathcal{C}_2$ is **more general** than $\mathcal{C}_1$ or is a parent of $\mathcal{C}_1$. An **extended child**

$\mathcal{C}'$ of a chronicle $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ is $\mathcal{C}' = (\mathcal{E} \cup \{e\}, \mathcal{T}')$ where $\mathcal{T}'$ is the union of $\mathcal{T}$ and of a set of temporal constraints between e and e_i for all e_i in $\mathcal{E}$. A **specialized child** $\mathcal{C}'$ of a chronicle $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ is $\mathcal{C}' = (\mathcal{E}, \mathcal{T} \backslash \{\tau\} \cup \{\tau'\})$ where $\tau' \subset \tau$.

Definition 4. Let $s = \langle (e_1, t_1),\ ...,\ (e_n, t_n) \rangle$ be a sequence and $\mathcal{C} = (\mathcal{E} = \{e'_1, \ldots, e'_m\}, \mathcal{T})$ a chronicle. An **occurrence** of the chronicle $\mathcal{C}$ in s is a subsequence $\tilde{s} = \langle (e_{f(1)}, t_{f(1)}),\ ...,\ (e_{f(m)}, t_{f(m)}) \rangle$ such that there exists a function f where $\forall i, j,\ 1 \leq i < j \leq m,\ 1 \leq f(i) \leq n,\ 1 \leq f(j) \leq n,\ f(i) \neq f(j)$ such that 1) $e'_i = e_{f(i)}$, $e'_j = e_{f(j)}$ and 2) $t_{f(j)} - t_{f(i)} \in [a, b]$ where $e'_i[a, b]e'_j \in \mathcal{T}$. $\mathcal{C}$ **occurs** in s, denoted by $\mathcal{C} \in s$, iff there is at least one occurrence of $\mathcal{C}$ in s.

Definition 5. The **support** of a chronicle $\mathcal{C}$ in a database of sequences $\mathcal{S}$ is the number of sequences in which $\mathcal{C}$ occurs: $support(\mathcal{C}, \mathcal{S}) = |\{S \mid S \in \mathcal{S} \text{ and } \mathcal{C} \in S\}|$. Given a minimal threshold $\sigma_{min} \in \mathbb{N}$, a chronicle $\mathcal{C}$ is said **frequent** in $\mathcal{S}$ iff $support(\mathcal{C}, \mathcal{S}) \geq \sigma_{min}$.

According to the anti-monotony property, if a chronicle $\mathcal{C}$ is frequent then all chronicles more general than $\mathcal{C}$ are frequent. One can easily be convinced of this by observing that, if a chronicle $\mathcal{C}'$ is more general than a chronicle $\mathcal{C}$, then $\mathcal{C}'$ has at least the same support as $\mathcal{C}$ because any occurrences of $\mathcal{C}$ is necessarily an occurrence of $\mathcal{C}'$. The proof is omitted for space reason.

3 Related works

The first algorithm dedicated to chronicle mining was proposed by Dousson and Duong [3]. This algorithm was originally designed for chronicle discovery in journal logs of telecommunication alarms. Recently, several improvements have been proposed [1, 2, 4]. All of these approaches are based on the anti-monotonicity property of frequency on the chronicle set.

Cram [2] proposed the *HCDA* algorithm which improves the first algorithm by mining the complete set of frequent chronicles. Those two approaches start with the extraction of frequent temporal constraints between pairs of events and then chronicles are generated by combining these constraints. The method of Dousson and Duong chooses only a representative of each type of temporal constraint while *HCDA* keeps all frequent temporal constraints in a graph of temporal constraints. These two methods process journal logs, *i.e.* a single long sequence. In our mining task, we have a database of sequences and the definition of the pattern support is based on a number of supported sequences. As a consequence, we can not apply these algorithms for our task. For this reason, we propose an adaptation of them.

In *CCP-Miner*, Huang [4] proposed to use chronicle mining on clinical pathways. Their data comes from inpatient electronic health record. Contrary to journal logs, a set of clinical pathways is a database of sequences. To simplify the evaluation of the support, *CCP-Miner* considers that an event type occurs at most one time in a pathway. Moreover, *CCP-Miner* is not complete. Chronicles are not obtained from the complete set of frequent multisets of event types but only those containing by frequent closed sequences.

Subias et *al.* [6] recently proposed an alternative support evaluation which is the number of sequences in which the number of occurrences of a chronicle in one sequence is above a given threshold. This support measure is not relevant in our application.

In parallel to alternative support, several approaches have been proposed to extract chronicles with simpler temporal constraints. For instance, Àlvarez et *al.* [1] use a similarity criterion to cluster together different temporal arrangements between events. Quiniou et *al.* [5] proposed an inductive logic programming approach to mine chronicles with quantified temporal constraints.

To the best of our knowledge, there is no algorithm that can extract a complete set of chronicles from a database in which sequences may contain duplicated events. The proposed method tackles this specific issue.

4 Complete chronicle mining in a database of sequences

The chronicle mining task is a classical pattern mining task. The search space is structured by a partial order, $\preceq$, (see section 2) and the frequency is an anti-monotonic measure in this space. As a consequence, the classical "generate and test" strategy can be applied: candidate k-patterns are generated from $(k-1)$-frequent patterns, frequency of candidates is evaluated. Then, the two main problems to tackle are 1) how to efficiently browse the search space and 2) how to evaluate the frequency of a pattern.

In this article, we propose an algorithm to extract the frequent chronicles in a database of sequences. This algorithm combines the approaches of $HCDA$ [2] and of $CCP\text{-}Miner$ [4]. We use the $CCP\text{-}Miner$ strategy that first extracts the multisets of event types and then add temporal constraints over those multisets. The generation of the temporal constraint is adapted from $HCDA$ in order to deal with databases of sequences. This two improvements are explained in the following section but before that, we detail the support evaluation process.

Enumerating sequences of a database that support a chronicle is simpler than the original chronicle enumeration of $HCDA$. In fact, evaluating the number of occurrences of a chronicle in a single sequence is very combinatorics because of the repetition of the events. Our support measure corresponds to the number of sequences where a chronicle occurs. For each sequence, we just have to search for one occurrence of this chronicle. Moreover, this support definition simplifies the construction of bases of constraints (see section 4.3).

4.1 A two steps strategy

Our algorithm is illustrated in Algorithm 1. Let $\mathcal{S}$ be a set of event sequences and σ_{min} be the minimal support threshold. Firstly, the $extractMultisets$ function generates ES, the set of all frequent multisets of event types $\mathcal{E}$ accordingly to σ_{min}. On the contrary to the $CCP\text{-}Miner$ approach, the algorithm does not generate chronicles from closed sequences. Multisets mining is an easy extension of itemsets mining and we do not detail with step of the algorithm.

Algorithm 1 Main algorithm for chronicle mining

1: $CS \leftarrow \emptyset$
2: $ES \leftarrow extractMultisets(\mathcal{S}, \sigma_{min})$
3: **for** each $e \in ES$ **do**
4: $\quad CS \leftarrow CS \cup extendChronicles(e, \sigma_{min})$
5: **return** CS

Then, multisets are extended in frequent chronicles and their temporal constraints are specialized. This step is performed for each multiset by the function *extendChronicles*. The set CS corresponds to the frequent chronicles. We detail this part of the algorithm in the following sections.

4.2 From multisets to chronicles

This section presents the generation of frequent chronicles from frequent multisets. Given a multiset $\mathcal{E}$, the exploration consists in generating all combinations of temporal constraints on $\mathcal{E}$, such that corresponding chronicles are frequent.

Temporal constraint bases To ensure the efficiency and the completeness of candidate generation, we use **temporal constraint bases** (TCB). A TCB is a set of graphs of temporal constraints (one per pair of events). Figure 2 illustrates a graph of temporal constraints.

Definition 6. A graph of temporal constraints $\mathcal{G}$ is a directed acyclic graph in which a node is a temporal constraint, τ, and children of τ are temporal constraint included in τ. The root of the graph is called the **top-constraint**. In our algorithm, we consider that a temporal constraint $\tau = e_1[a, b]e_2$ has at most two children, $\tau_{left} = e_1[a, b']e_2$ and $\tau_{right} = e_1[a', b]e_2$ where $b' < b$ and $a' > a$.

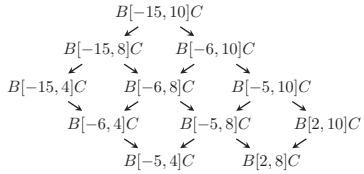


Fig. 2: A graph of temporal constraints for the pair of events (B, C).

Specialization of multisets Let $\mathcal{E}$ be a multiset. The "top-chronicle" $(\mathcal{E}, \mathcal{T})$ is generated such that for each $\tau \in \mathcal{T}$, τ is a top-constraint. Then, function

extendChronicles generates the complete set of frequent chronicles from $(\mathcal{E}, \mathcal{T})$ by specializing the temporal constraints on the multiset $\mathcal{E}$. The specialization of a chronicle consists in specializing a temporal constraint τ according to the specialization defined in its graph of temporal constraints.

The generation of chronicles is a "generate and test" approach. The specialization is done recursively to extract the complete set of frequent chronicles. Each time a chronicle $\mathcal{C}$ is specialized, its frequency in the database is evaluated. If its support is not above the minimal support, the chronicle is pruned. According to the anti-monotony property, we know that not any chronicle $\mathcal{C}'$, $\mathcal{C}' \preceq \mathcal{C}$, will be frequent.

The enumeration of specialized chronicles is done without redundancy. It is ensured thanks to the definition of a relation order $\lhd_{\mathcal{E}}$ amongst all chronicles sharing the same multiset $\mathcal{E}$. Let $\mathcal{C} = (\mathcal{E}, \mathcal{T} = \{t_1, \ldots, t_n\})$ and $\mathcal{C}' = (\mathcal{E}, \mathcal{T}' = \{t'_1, \ldots, t'_n\})$ be chronicles, $\mathcal{C}' \lhd_{\mathcal{E}} \mathcal{C}$ iff $\exists k$, $1 \leq k < n$ such that 1) $\forall i$, $1 \leq i < k$, $t_i = t'_i$, 2) $\forall j$, $k < j < n$, $t_j = t'_j$ and t_j is a top-constraint and 3) $t'_k = t_{k_{right}}$ otherwise $t'_k = t_{k_{left}}$ if $\nexists \tau$, $\tau_{left} = t'_k$.

It can be shown that $\mathcal{C}' \lhd_{\mathcal{E}} \mathcal{C} \Rightarrow \mathcal{C}' \preceq \mathcal{C}$ and that there exists at most one chronicle $\mathcal{C}$ such that $\mathcal{C}' \lhd_{\mathcal{E}} \mathcal{C}$. These properties ensure a unique and complete traversal of the search space. For space reason, we omit the proof of these properties.

4.3 Generation of the Base of Temporal Constraints

This section presents the generation of the TCB. The smaller are the TCB, the more efficient is the multiset specification. On the other hand, these bases must be complete, *i.e.* the chronicle mining algorithm of the section 4.1 must extract the complete set of frequent chronicles.

In these objectives, the algorithm generates the smaller complete TCB from the sequence database. A first algorithm has been proposed in *HCDA*. Our algorithm improves it by considering two specificities of our dataset:

1. the enumeration of chronicle occurrences in a database of sequences
2. the specificities of events that encode the period of an interval event with a pair of point-based events

Let $(e, e') \in \mathbb{E}^2, e \leq_{\mathbb{E}} e'$ be a pair of point-based events. We denote by $\mathcal{A}_{ee'} \subset \mathbb{R}$, the list of pairs (a, SID) for each co-occurrence $((e, t), (e', t'))$ in a sequence where a is the duration $t' - t$ and SID is the identifier of the sequence. The lists corresponding to all the pairs (e, e') in $\mathbb{E}^2$ can be filled in one pass of the database by generating all co-occurrences present in each sequence. We can notice that the duration can be negative if e' occurs before e. After this step, the lists are sorted by duration in ascending order and the duplicates of couple are removed. Similar lists are built from start/finish events of intervals.

The temporal constraint graph generation is given in the Algorithm 2. Each list $\mathcal{A}_{ee'}$ corresponds to a graph $\mathcal{G}_{ee'}$. To respect our support measure we check whether elements of $\mathcal{A}_{ee'}$ correspond to at least σ_{min} sequences. Otherwise $\mathcal{G}_{ee'}$ is empty. In the other case, the root of $\mathcal{G}_{ee'}$ is $e[a, b]e'$ where a is the duration of

Algorithm 2 Temporal constraint graph generation

```
1: function ConstructGraph(A_ee', σ_min)
2:     τ ← ∅
3:     if |{SID | (a, SID) ∈ A_ee'}| ≥ σ_min then
4:         (a, s) ← first(A_ee')
5:         (b, t) ← last(A_ee')
6:         τ ← e[a, b]e'
7:         τ_left ← ConstructGraph(A_ee'\{(b, t)})
8:         τ_right ← ConstructGraph(A_ee'\{(a, s)})
9:     return τ
```

the first element of $\mathcal{A}_{ee'}$ and b that of the last one. Then we built $\mathcal{G}_{ee'}$ recursively by defining the left child of a node as the graph corresponding to $\mathcal{A}_{ee'}$ without its last element and right child to $\mathcal{A}_{ee'}$ without its first element.

Finally, we can notice that our algorithm can take into account some classical constraints of the sequential pattern mining task. These constraints require additional parameters given by the expert. For example, it is possible to define a maximal/minimal size of chronicles, *i.e.* the number of events in their multiset. We can also use a maximal window constraint mwc to constraint events of chronicles to occurs together in a temporal window of maximal size mwc.

5 Experiments and results

We implemented a first prototype of our algorithm in C++ and we evaluate its efficiency on a real dataset of care-pathways.

5.1 Rational

The objective of our pharmaco-epidemiological study is to assess whether or not brand-to-generic antiepileptic drugs substitution is associated with seizure-related hospitalization. Our data represents 1,810,600 deliveries of 7,693 different drugs for 8,378 patients treated for epilepsy within a period from 03/02/2007 to 12/29/2011. We collected also 12,347 seizure-related hospitalizations on the same period concerning 7,754 patients.

In a first step, a naive algorithm abstracts drug deliveries into drug exposures. The algorithm transforms several point-based events $(e_1, \ldots, e_n)$, some drug deliveries, in a single interval-based event e, a drug exposure if 1) $n \geq rep_{min}$ and 2) two successive events are not spaced with more than gap_{max} time units. rep_{min} and gap_{max} are input parameters. We arbitrary choose to set gap_{max} to 30 and rep_{min} to 2 for our experiments.

To reduce the computing time of the TCB generation, we prefer to test our prototype on 10% percent of the original dataset corresponding to 839 care-pathways. In fact, the number of chronicles generated is not disturbed because

minimal support (σ_{min})	**mwc**	**#frequent chronicles**	**CPU time (s)**
84	90	463,006	440
84	91	917,952	537
84	92	1,506,243	730
84	93	1,868,923	808
84	94	2,596,401	1,053
84	95	3,878,436	1,479
85	90	342,598	231
86	90	246,724	209
87	90	173,872	167
167	180	1,861,607	958

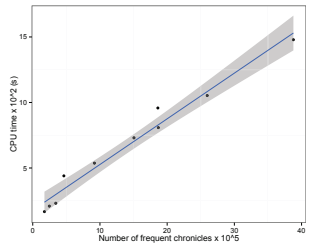


Fig. 3: Execution time results of our prototype. On the left, table of results ; On the right, CPU time wrt. number of frequent chronicles.

the minimal support constraint is defined as a percentage of the number of care-pathways. We constraint the generated chronicles to be ended by a specific event corresponding to an hospitalization for epileptic crisis. We are interested in the number of frequent chronicles generated (containing more than 2 events) and in the computing times.

5.2 Results

To distinguish the time to generate TCB and the time to extract frequent chronicles, we started to run this generation on our dataset with different couples of parameters. Parameters of this generation are a minimal support threshold σ_{min} and a maximal window constraint mwc. We ran 3 generations, one for $f_{min} = 10\%$ ($\sigma_{min} = 84$) and $mwc = 90$ days which generates 76,503 temporal constraints, an other for $f_{min} = 20\%$ and $mwc = 180$ days which generates 236,942 temporal constraints and a last one for $f_{min} = 20\%$ and $mwc = 20$ days which generates 0 temporal constraint. The computing time of the three generations is about 135 seconds. We can conclude that the generation time of the TCB only depends on the number and on the size of sequences but not on the parameters of the algorithm.

The Figure 3 illustrates computing times for different settings of our experiment. We can first notice that our algorithm generates millions of chronicles. Moreover, we precise that, for this dataset, all frequent chronicles have a multiset of events containing 3 events and that they are mainly specialization of the same multiset. By setting the minimal support threshold, we notice that the number of returned patterns is very sensitive to the maximal window constraint parameter. We next remark that the computing time is linear with the number of chronicles. If we only look at the settings which extract more than one million of chronicles, we observe that our algorithm can extract about 2300 chronicles per second.

6 Conclusion

Chronicles seem to be relevant to represent interesting patterns for pharmaco-epidemiology studies. Their expressiveness enables to model complex temporal behaviours of patients in the health care system (*e.g.* consultation, hospitalization and drugs delivery). In this article, we proposed a chronicle mining algorithms to the specificities of our database of sequences: sequences with interval-based events and sequences with repeated events. Our algorithm extracts the complete set of chronicles from a database of sequences. It has been implemented and evaluated on a real dataset of care-pathways. The experiments shown that our algorithm was able to generate very large numbers of chronicles.

We are now facing a classical pattern mining issue: the deluge of frequent patterns. Our main perspective is to tackle this issue. Several research directions can by studied, for instance, a heuristic to explore the search space or a method to extract a smaller set of chronicles like closed chronicles. Another way to reduce the number of frequent chronicles could be to consider as similar the chronicles with same multisets of event types and "similar" temporal constraint sets. Finally, visualization could help clinicians to define interesting patterns during the extraction, and the clinician's feedback could pilot the algorithm to the patterns he/she considers as more interesting.

Acknowledgements

This work is a part of the PEPS (Pharmaco-epidemiology of health products) funded by the French national agency for medicines and health products safety.

References

1. Álvarez, M.R., Félix, P., Cariñena, P.: Discovering metric temporal constraint networks on temporal databases. Artificial Intelligence in Medicine 58(3), 139–154 (2013)
2. Cram, D., Mathern, B., Mille, A.: A complete chronicle discovery approach: application to activity analysis. Expert Systems 29(4), 321–346 (2012)
3. Dousson, C., Duong, T.V.: Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 620–626 (1999)
4. Huang, Z., Lu, X., Duan, H.: On mining clinical pathway patterns from medical behaviors. Artificial Intelligence in Medicine 56(1), 35–50 (2012)
5. Quiniou, R., Cordier, M.O., Carrault, G., Wang, F.: Application of ILP to cardiac arrhythmia characterization for chronicle recognition. In: Proceeding of the conference on Inductive Logic Programming. pp. 220–227 (2001)
6. Subias, A., Travé-Massuyès, L., Le Corronc, E.: Learning chronicles signing multiple scenario instances. In: Proceedings of the 19th World Congress of the International Federation of Automatic Control. pp. 397–402 (2014)

Sequential Pattern Mining and its application to Document Classification

José Kadir Febrer-Hernández[1], Raudel Hernández-León[1], José Hernández-Palancar[1], and Claudia Feregrino-Uribe[2]

[1]Centro de Aplicaciones de Tecnologías de Avanzada
7ma A ♯21406 e/ 214 y 216, Rpto. Siboney, Playa, CP: 12200, La Habana, Cuba.
{jfebrer,rhernandez,jpalancar}@cenatav.co.cu
[2]Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro #1, Sta. María de Tonantzintla, Puebla, CP: 72840, México.
cferegrino@ccc.inaoep.mx

Abstract. In this paper, we present the main research ideas of my doctoral studies, which aims to contribute in the field of sequential pattern mining and sequential patterns-based classification. Among our results, we propose an algorithm to compute frequent sequences and its extension to compute sequential patterns-based rules. Additionally, we introduce some improvements to the Sequential Patterns-based Classifiers.

Keywords: data mining, sequential patterns mining, document classification.

1 Motivation and problem description

Sequential pattern mining is a well-known data mining technique that aims to compute all frequent (or interesting) sequences from a transactional dataset. Unlike an itemset, in which an item can occur at most once, in a sequence an itemset can occur multiple times. Additionally, in itemset mining, $(abc) = (cba)$ but in sequence mining, $\langle\ (ab)\ c\ \rangle \neq \langle\ c\ (ab)\ \rangle$.

In the last decades, some works have used sequential patterns to increase the accuracy of classifiers. An important part of the Sequential Patterns-based Classification (SPaC) is the process of mining the set of classification rules, called SPaRs (Sequential Patterns-based Rules). A SPaR describes an implicative co-occurring relationship between a sequence α and a class c. Notice that the algorithms for SPaRs generation can be easily extended from the sequential pattern mining algorithms.

Both sequential pattern mining and sequential patterns-based classification have been used in several application areas, for example in web access analysis [1], text mining [2], disease treatments [3], document-specific keyphrase extraction [4], among others.

The main reported algorithms obtain the interesting sequential patterns either using a depth first search strategy or generating the sequences of size k by combining the sequences of size $(k-1)$ with a common $(k-2)$-length prefix. In

order to reach better performance, we proposed (and published in [5]) a novel strategy that generates the sequences of size k by combining the sequences of size $(k-1)$ with the sequences of size 2. Also in [5], we introduced a new data structure to store the interesting sequences and a new pruning strategy to reduce the number of candidate sequences.

In general, the accuracy of the sequential patterns-based classifiers depends on four main elements: (1) the quality measure used to generate the SPaRs, (2) the pruning strategy used to reduce the number of candidate rules, (3) the rule ordering strategy and (4) the mechanism used for classifying unseen transactions. Therefore, any of the main sequential pattern mining algorithms (GSP [6], PrefixSpan [7], LAPIN [8] and PRISM [9]) can be adapted to generate the set of interesting SPaRs.

Currently, all classifiers based on sequential patterns use the Support and Confidence measures for computing and ordering the set of SPaRs. However, several authors have pointed out some drawbacks of these measures [10], for example, Confidence detects neither statistical independence nor negative dependence among items (misleading rules).

On the other hand, many studies [6, 11] have indicated the high number of rules that could be generated using a small Support threshold. To address this problem, recent works [12] prune the rules search space each time that a rule satisfies both Support and Confidence thresholds, it means that rules satisfying both thresholds are not extended anymore. Using this strategy, it is more frequent the generation of general (short) rules than the generation of specific (large) rules, some of which could be more interesting.

The existence of these drawbacks have been the main motivation of our research and, in order to overcome them, we have proposed some general improvements to the sequential patterns-based classifiers. The rest of this paper is organized as follows. In the next subsection a formal definition of both problems is presented. Related work is described in Section two. Our proposal are presented in section three. In the fourth section the experimental results and the work in progress are shown. Finally, the conclusions are given in section five.

1.1 Describing the problem

As it can be see above, we have two main objectives in this research:

- to develop a novel algorithm (heuristic, strategy) to improve the efficiency of the sequence mining process.
- to propose new improvements to the sequential patterns-based classifiers.

These two objectives are very related because the sequential patterns-based classifiers need to compute a set of rules (SPaRs) in a first stage and, the algorithms to compute the SPaRs are easily extended from the sequence mining algorithms. In this subsection, we will offer a formal definition of both problems.

In sequence mining, it is assumed that a set of items $I = \{i_1,\ i_2,\ ...,\ i_l\}$ and a set of transactions T are given, where each transaction $t \in T$ consists of a

sequence $\langle \alpha_1\ \alpha_2\ ...\ \alpha_n \rangle$, so that $\alpha_i \subseteq I$. The Support of a sequence α, denoted as $Sup(\alpha)$, is the fraction of transactions in T containing α (see Eq. 1).

$$Sup(\alpha) = \frac{|T_\alpha|}{|T|} \tag{1}$$

where T_α is the set of transactions in T containing α (see Def. 1) and $|\cdot|$ is the cardinality operator.

Definition 1 *Let $\alpha = \langle \alpha_1\ \alpha_2\ ...\ \alpha_n \rangle$ and $\beta = \langle \beta_1\ \beta_2\ ...\ \beta_m \rangle$ be sequences, we will say that α is contained in β if there exists integers $1 \leq j_1 < j_2 < ... < j_n \leq m$ such that $\alpha_1 \subseteq \beta_{j_1}$, $\alpha_2 \subseteq \beta_{j_2}$, ..., $\alpha_n \subseteq \beta_{j_n}$, with $\beta_{j_i} \in \beta$.*

Let *minSup* be a minimum Support threshold previously defined, an algorithm for interesting sequence mining computes all the sequences α such that $Sup(\alpha) > minSup$; when Support is the used measure the interesting sequences are called frequent sequences.

On the other hand, in sequential patterns-based classification we also have a set of classes C, and each transaction $t \in T$ consists of a sequence $\langle \alpha_1\ \alpha_2\ ...\ \alpha_n \rangle$, so that $\alpha_i \subseteq I$, and a class $c \in C$. A SPaR is an implication of the form $\alpha \Rightarrow c$ where α is a sequence and $c \in C$. The size of a SPaR is defined as its cardinality, a SPaR containing k itemsets (including the class) is called a k-SPaR. The rule $\alpha \Rightarrow c$ is held in T with certain Support and Confidence (see Eqs. 2 and 3).

$$Sup(\alpha \Rightarrow c) = Sup(\alpha \otimes \langle c \rangle) \tag{2}$$

where $\otimes$ is the concatenation operator (see Def. 2).

$$Conf(\alpha \Rightarrow c) = \frac{Sup(\alpha \Rightarrow c)}{Sup(\alpha)} \tag{3}$$

Definition 2 *Let $\alpha = \langle \alpha_1\ \alpha_2\ ...\ \alpha_n \rangle$ and $\beta = \langle \beta_1\ \beta_2\ ...\ \beta_m \rangle$, we will call the sequence $\langle \alpha_1\ \alpha_2\ ...\ \alpha_n\ \beta_1\ \beta_2\ ...\ \beta_m \rangle$ the concatenation of α and β, and we will use the operator $\otimes$ to indicate it.*

In general, a classifier based on this approach usually consists of an ordered SPaR list l, and a mechanism for classifying new transactions using l.

2 Related work

The sequential pattern mining algorithms can be split into two main groups: (1) apriori-like algorithms (AprioriAll and AprioriSome [13], GSP [6] and (2) pattern-growth based algorithms (PrefixSpan [7], LAPIN [8], PRISM [9]).

In [6], the authors proposed the GSP algorithm, which includes time constrains and taxonomies in the mining process. The PrefixSpan algorithm, proposed in [7], is based on recursively constructing the patterns by growing on the prefix, and simultaneously, restricting the search to projected datasets and reducing the search space at each step.

PRISM, the algorithm introduced by Karam Gouda in [9], uses a vertical approach for enumeration and support counting, based on the novel notion of primal block encoding, which is based on prime factorization theory. The LAPIN (LAst Position INduction) algorithm [8] uses an item-last-position list and a prefix border position set instead of the tree projection or candidate generate-and-test techniques introduced so far.

As we mentioned in Section 1, the sequential pattern mining algorithms can be easily adapted to generate the set of SPaRs. Once the SPaRs are generated, these are ordered. For this task there are six main strategies reported in the literature: Confidence-Support-Antecedent, Antecedent-Confidence-Support, Weighted Relative Accuracy, Laplace Expected Error Estimate, Chi-Square and L^3. In [14], the authors show that the L^3 rule ordering strategy obtains the best results of all strategies mentioned above. However, all these ordering strategies are based on Confidence measure.

Once a SPaR-based classifier has been built, usually presented as a list of sorted SPaRs, there are three main mechanisms for classifying unseen data [12].

- **Best rule**: This mechanism assigns the class of the first ("best") rule in the order that satisfies the transaction to be classified.
- **Best K rules**: This mechanism selects the best K rules (for each class) that satisfy the transaction to be classified and then the class is determined using these K rules, according to different criteria.
- **All rules**: This mechanism selects all rules that satisfy the unseen transaction and then these rules are used to determine their class.

Since the "Best K rules" mechanism has been the most widely used for rule-based classification, reporting the best results, it was used in our experiments.

3 Our proposal

In this section, we present the main contributions of our research. First, in subsection 3.1 we introduce the new data structure used to store the frequent sequences. Later, the algorithm to compute the frequent sequences and its extension to compute the set of SPaRs are described in subsections 3.2 and 3.3.

3.1 Storing useful information

Let α be a frequent sequence and T be a transactional dataset, the proposed data structure stores, for each $t \in T$, a list L_t with the occurrence positions of α in t (see Def. 3). Additionally, a bit-vector of 1's and 0's is stored representing the presence or absence of α in each transaction of T.

Definition 3 *Let $\alpha = \langle \alpha_1\ \alpha_2\ ...\ \alpha_n \rangle$ and $\beta = \langle \beta_1\ \beta_2\ ...\ \beta_m \rangle$ be sequences such that α is contained in β (i.e. exists integers $1 \leq j_1 < j_2 < ... < j_n \leq m$ such that $\alpha_1 \subseteq \beta_{j_1}$, $\alpha_2 \subseteq \beta_{j_2}$, ..., $\alpha_n \subseteq \beta_{j_n}$), we will call occurrence position of α in β ($occP(\alpha, \beta)$) to the least position of all possible β_{j_n} in β, if $\mid \alpha \mid > 2$, and the set of positions of all possible β_{j_n} in β, if $\mid \alpha \mid \leq 2$.*

In Table 1, five transactions and the occurrence positions of three sequences of different sizes are shown. Notice that when $\mid \alpha \mid > 2$ (*e.g.* $\langle a\ f\ b \rangle$ in transaction 2) we could also have several β_{j_n} (*e.g.* $(b:4)$ and $(b:6)$) but the proposed strategy to generate the candidate, only require the least of all.

Table 1. Example of five transactions and the occurrence positions of three sequences of different sizes.

Tid	Sequence	$\langle$b$\rangle$	$\langle$a f$\rangle$	$\langle$a f b$\rangle$
1	$\langle a\ b \rangle$	(b:2)		
2	$\langle cd\ a\ ef\ b\ cd\ ab \rangle$	(b:4), (b:6)	(f:3)	(b:4)
3	$\langle af\ f \rangle$		(f:2)	
4	$\langle af\ ef\ bf \rangle$	(b:3)	(f:2), (f:3)	(b:3)
5	$\langle b \rangle$	(b:1)		

3.2 Algorithm for mining the frequent sequences

In this subsection we describe a novel algorithm, called SPaMi-FTS and published by us in [5], to compute the set of frequent sequences. In a first step, SPaMi-FTS computes all frequent 1-sequences storing for each frequent sequence α (of any size) and for each transaction $t \in T$, a list with the occurrence positions of α in t (see Def. 3 in Section 3.1). Also a bit-vector representing the presence or absence in each transaction is stored.

In a second step, SPaMi-FTS computes the frequent 2-sequences; for this, it first generates the candidate 2-sequences by combining the frequent 1-sequences obtained in the first step and later, it applies a pruning strategy to reduce the number of Support counting. This pruning strategy intersects, using AND operations, the bit-vectors of two sequences obtaining the highest possible Support, which is used to decide if it is required (or not) to compute the real Support.

The pseudo code of SPaMi-FTS is shown in Algorithm 1, where the method $pruningMethod(\langle i \rangle, \langle j \rangle)$ work as follows: the bit-vectors of $\langle i \rangle$ and $\langle j \rangle$ are intersected to compute the highest possible Support value of $\langle i\ j \rangle$, named $highPosSup$. If $highPosSup \leq minSup$ then the sequence $\langle i\ j \rangle$ is pruning.

Finally, in a third stage, the procedure used to compute the frequent 2-sequences is extended to compute the frequent k-sequences ($k > 2$). For this, the candidate k-sequences are obtained by combining each frequent $(k-1)$-sequence $\alpha = \langle \alpha_1\ \alpha_2\ ...\ \alpha_k \rangle$ with all frequent 2-sequences of the form $\beta = \langle \alpha_k\ \beta_1 \rangle$.

3.3 Algorithm for mining the interesting SPaRs

As we mentioned in Section 1, all classifiers based on sequential patterns use the Support and Confidence measures for computing the set of SPaRs. However, several authors have pointed out some drawbacks of these measures that could lead us to discover many more rules than it should [10]. In particular, items with high Support can lead us to obtain misleading rules (see Ex. 1) because they appear in many transactions and they could be predicted by any itemset.

Algorithm 1: Pseudo code to compute the frequent k-sequences.

```
Input: Transactional dataset T, set of frequent (k − 1)-sequences kFreq, set of frequent
       2-sequences twoFreq and Support threshold minSup.
Output: Set of frequent k-sequences.
L1 ← ∅
foreach α : ⟨α1 α2 ... αk⟩ ∈ kFreq do
    foreach β : ⟨αk β1⟩ ∈ twoFreq do
        prune ← pruningMethod(α, β)
        if not prune then
            Sup ← 0
            foreach t ∈ T do
                if occPos(β, t) > occPos(α, t) then
                    Sup ← Sup + 1
                end
            end
            if Sup > minSup then
                L1 ← L1 ∪ {α ⊗ β}
            end
        end
    end
end
return L1
```

Example 1 *Without loss of generality, let us assume that* $Sup(X) = 0.4$, $Sup(Y) = 0.8$ *and* $Sup(X \Rightarrow Y) = 0.3$, *therefore* $Sup(\neg X) = 1$ - $Sup(X) = 0.6$ *and* $Sup(\neg X \Rightarrow Y) = Sup(Y)$ - $Sup(X \Rightarrow Y) = 0.5$. *If we compute* $Conf(X \Rightarrow Y)$ *we obtain 0.75 (a high Confidence value) but* Y *occurs in* 80 % *of the transactions, therefore the rule* $X \Rightarrow Y$ *does worse than just randomly guessing. In this case,* $X \Rightarrow Y$ *is a misleading rule.*

On the other hand, in [15] the authors proposed a measure, called Netconf (see Eq. 4), to estimate the strength of a rule. In general, this measure solves the main drawbacks of the Confidence measure, reported in other works [10].

$$Netconf(X \Rightarrow Y) = \frac{Sup(X \Rightarrow Y) - Sup(X)Sup(Y)}{Sup(X)(1 - Sup(X))} \tag{4}$$

The Netconf has among its main advantages that it detects the misleading rules obtained by the Confidence. For the Ex. 1, $Netconf(X \Rightarrow Y) = -0.083$ showing a negative dependence between the antecedent and the consequent. Therefore, in this research we propose to use the Netconf measure instead of Support and Confidence for computing and ordering the set of SPaRs.

Most of the algorithms in SPaR-based classification [6] prune the SPaRs search space each time a SPaR satisfying the defined thresholds is found, it produces general (small) rules reducing the possibility of obtain specific (large) rules, some of which could be more interesting. Besides, since the defined threshold(s) must be satisfied, many branches of the rules search space could be explored in vain.

In our proposal, instead of pruning the SPaR search space when a SPaR satisfies the Netconf threshold, we propose the following pruning strategy:

- If a SPaR r does not satisfy the Netconf threshold $minNF$ ($r.NF \leq minNF$) we do not extend it anymore avoiding to explore this part of the SPaR search space in vain.
- Let $r_1 : \alpha \Rightarrow c$ and $r_2 : \beta \Rightarrow c$ be SPaRs, if the SPaR $r : \langle \alpha \otimes \beta \rangle \Rightarrow c$ satisfies the Netconf threshold but $r.NF < r_1.NF$ and $r.NF < r_2.NF$ then we prune r avoiding to generate SPaRs with less quality than their parents.

The intuitive idea (or hypothesis) behind this pruning strategy is that specific rules with high Netconf values are better to classify than general rules with high Netconf values. Taking into account the advantages of the Netconf measure and the novel pruning strategy, we extend the SPaMi-FTS algorithm to generate the set of SPaRs. This extension, called SPaR-NF, does not apply the pruning strategy used by SPaMi-FTS because the Netconf measure is used inside of Support measure. Since the pseudo code of SPaR-NF is similar to the pseudo code of SPaMi-FTS, and considering the space limitations, we do not describe in this paper.

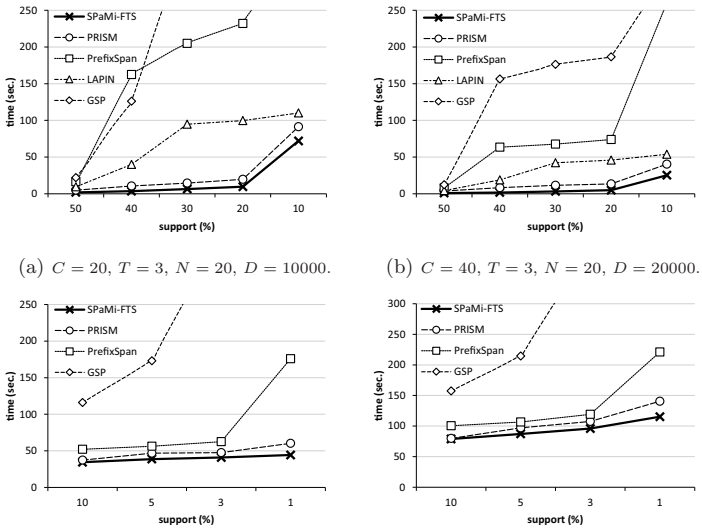
Once the set of SPaRs has been generated, using the SPaR-NF algorithm, the SPaR list is sorted. For this purpose, we propose sorting the set of SPaRs in a descending order according to their sizes (the largest first) and in case of tie, we sort the tied SPaRs in a descending order according to their Netconf (the highest values first). For classifying unseen transactions, we decided to follow the "Best K rules" mechanism, because, as it was explained above, the "Best rule" mechanism could suffer biased classification or overfitting since the classification is based on only one rule; and the "All rules" mechanism takes into account rules with low ranking, which affects the accuracy of the classifier.

4 Experimental results

In this section, we present some experimental results in order to evaluate the efficiency of the SPaMi-FTS algorithm and the accuracy of the proposed classifier, called SPaC-NF, which uses the SPaR-NF algorithm to compute the SPaRs.

In case of SPaMi-FTS, we show the result of the comparison between it and the main sequence mining algorithms reported in the literature: GSP [6], PrefixSpan [7], LAPIN [8], PRISM [9]. All codes (implemented in ANSI C standard) were provided by their authors. Our experiments were done over seven datasets, built with a synthetic dataset generator developed by the Data Mining Research Group at the Department of Computer Science, University of Illinois. Several experiments were conducted to evaluate the performance of the algorithms when these parameters change. In the first experiment we used the parameter values most employed in the literature (see their parameter values in the Fig. 1(a)). As it can see in Fig. 1(a), SPaMi-FTS obtains the best result, followed by PRISM.

To the second experiment, we simultaneously increased D (from 10000 to 20000) and C (from 20 to 40). As result, GSP and PrefixSpan were the most affected algorithms, in both cases the runtime were over 250 seconds (see Fig. 1(b)). In the third and fourth experiments, we increased N (from 20 to 50)



(a) $C = 20$, $T = 3$, $N = 20$, $D = 10000$. (b) $C = 40$, $T = 3$, $N = 20$, $D = 20000$.

(c) $C = 40$, $T = 3$, $N = 50$, $D = 5000$. (d) $C = 40$, $T = 3$, $N = 50$, $D = 10000$.

Fig. 1. Runtime comparison using DS_1, DS_2, DS_3 and DS_4 datasets.

keeping C set to 40 and varying D from 5000 in Fig. 1(c) to 10000 in Fig. 1(d). In both figures, we show the runtime of all evaluated algorithms with the exception of LAPIN algorithm because of it crashed with support thresholds under 10 %. In general, the SPaMi-FTS algorithm has the best performance of all tested algorithms both in scalability and in runtime.

In case of SPaC-NF classifier, we use three document collections, and we compare it against other classifiers as NaiveBayes, PART [16], J48 [17], Support Vector Machines [18] and against a classifier (SPaC-MR) built with the Main Results obtained in SPaR-based classification. All these classifiers, with the exception of SPaC-NF and SPaC-MR, were evaluated using Weka. The experiments were done using ten-fold cross-validation, reporting the average over the ten folds. Similar to other works, experiments were conducted using several document collections, three in our case: AFP (http://trec.nist.gov), TDT (http:/www.nist.gov) and Reuter (http://kdd.ics.uci.edu).

In the same way as in other works [13], for all used datasets, sentences are distinguished and ordered in each document. This means that the document is considered as being an ordered list of sentences. Each sentence is considered as being an unordered set of words. Therefore, we represented the document as a sequence of itemsets where each one corresponds with the set of words of each sentence.

In Table 2, the results show that SPaC-NF yields an average accuracy higher than the other evaluated classifiers, having in average a difference in accuracy of 3.2 % with respect to the classifier in the second place (SVM classifier).

Table 2. Comparison against other Sequential-patterns based Classifiers.

Dataset	SVM	J48	NaiveBayes	PART	SPaC-MR	SPaC-NF
AFP	88.7	81.5	83.6	78.3	89.5	93.8
TDT	89.6	86.2	80.8	75.4	87.1	91.9
Reuter	82.5	79.3	78.2	75.7	80.3	84.7
Average	86.9	82.3	80.8	76.4	85.6	90.1

In Table 3, we show the impact of our improvements. For this, we compare our approach (SPaC-NF) that uses the Netconf measure and obtains large rules against a SPaR-based classifier (SPaC-MR) that uses the Confidence measure and obtains short rules. Additionally, for both classifiers, we evaluate the best rule ordering strategy reported (L^3) and the strategy proposed by us, based on their rule sizes (largest first) and Netconf values.

Table 3. Impact of the different improvements in a general SPaC-based classifier.

	SPaC-MR		SPaC-NF	
Dataset	L^3	Size & NF	L^3	Size & NF
AFP	89.5	90.9	92.4	93.8
TDT	87.1	88.6	90.3	91.9
Reuter	80.3	81.8	83.5	84.7
Average	85.6	87.1	88.7	90.1

5 Work in progress

As we mentioned in Section 2, the "Best K rules" mechanism has been the most widely used for rule-based classification. However, using this mechanism could affect the classification accuracy. Ever more when most of the best K rules were obtained extending the same item (or itemset), or when there is an imbalance among the numbers of SPaRs with high quality measure values, per each class, that cover the new transaction.

Taking into account these limitations, we are working in the development of a new satisfaction mechanism that selects the value of K in a dynamic way. Additionally, we are evaluating the use of different quality measures in the SPaRs generation process.

In case of SPaMi-FTS algorithm, we are testing its scalability with respect to the number of transactions and number of itemsets per transaction. Furthermore, we plan to improve our pruning strategy to make the SPaMi-FTS algorithm even more efficient. In order to make the algorithms suitable for real life applications (*e.g.* Web access patterns, customer purchase behavior) we will consider some constrains like time-windows and gaps among elements.

References

1. Haleem, H., Kumar, P. and Beg, S.: Novel frequent sequential patterns based probabilistic model for effective classification of web documents. In Computer and Communication Technology, 2014 International Conference on, pp. 361-371, 2014.
2. Cesario, E., Folino, F. and Locane, A.: Boosting text segmentation via progressive classification. In Knowl. Inf. Syst., Vol. 15, Number 3, pp. 285-320, 2008.
3. Liao, V. and Chen, M.: An efficient sequential pattern mining algorithm for motifs with gap constraints. In Proceedings of the BIBM, Vol. 0, Number 1, 2012.
4. Xei, F., Wu, X. and Zhu, X.: Document-Specific Keyphrase Extraction Using Sequential Patterns with Wildcards. In Proceedings of the ICDM, 2014.
5. Febrer, J. K., Hernández, J., Hernández, R. and Feregrino, C.: SPaMi-FTS: An Efficient Algorithm for Mining Frequent Sequential Patterns. In LNCS, Vol. 8827, pp 470-477, 2014.
6. Srikant, R. and Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In Proceeding in the 5th International Conference Extending Database Technology, pp. 3-17, 1996.
7. Pei, J., Han, J., Mortazavi-asl, B., Pinto, H. and Chen, Q.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In Proceedings of the 17th International Conference on Data Engineering, pp. 215-224, 2001.
8. Yang, Z., Wang, Y. and Kitsuregawa, M.: LAPIN: Effective Sequential Pattern Mining Algorithms by Last Position Induction for Dense Databases. In LNCS, Vol. 4443, pp. 1020-1023, 2007.
9. Gouda, K., Hassaan, M. and Zaki, M. J.: Prism: An effective approach for frequent sequence mining via prime-block encoding. In J. Comput. Syst. Sci., Vol. 76, Number 1, pp. 88-102, 2010.
10. Steinbach, M. and Kumar, V.: Generalizing the Notion of Confidence. In Proceedings of the ICDM, pp. 402-409, 2005.
11. Mannila, H. and Toivonen, H.: Discovery of Frequent Episodes in Event Sequences. In Data Min. Knowl. Discov., Vol. 1, Number 3, pp. 258-289, 1997.
12. Wang, Y., Xin, Q. and Coenen, F.: Hybrid Rule Ordering in Classification Association Rule Mining. In Trans. MLDM, Vol. 1, Number 1, pp. 1-15, 2008.
13. Agrawal, R. and Srikant, R.: Mining Sequential Patterns. In Proceedings of the Eleventh International Conference on Data Engineering, pp. 3-14, 1995.
14. Hernández, R., Carrasco, J. A., Martínez, J. Fco. and Hernández, J.: Combining Hybrid Rule Ordering Strategies Based on Netconf and a Novel Satisfaction Mechanism for CAR-based Classifiers. In Intell. Data Anal., Vol. 18, Number 6S, pp. S89-S100, 2014.
15. Ahn, K. I. and Kim, J. Y.: Efficient Mining of Frequent Itemsets and a Measure of Interest for Association Rule Mining. In Information and Knowledge Management, Vol. 3, Number 3, pp. 245-257, 2004.
16. Frank, E. and Witten, I. H.: Generating Accurate Rule Sets Without Global Optimization. In Proceedings of the 15th ICML, pp. 144-151, 1998.
17. Quinlan, J. R.: C4.5: Programs for Machine Learning. Published by Morgan Kaufmann Publishers Inc., 1993
18. Cortes, C. and Vapnik, V.: Support-Vector Networks. In Mach. Learn., Vol. 20. Number 3, pp. 273-297, 1995.

Unsupervised Image Analysis & Galaxy Categorisation in Multi-Wavelength Hubble Space Telescope Images

Alex Hocking[1], J. E. Geach[2], Yi Sun[1], Neil Davey[1], and Nancy Hine[2]

[1] Computer Science & Informatics Research Centre, University of Hertfordshire, UK
[2] Centre for Astrophysics Research, University of Hertfordshire, UK
a.hocking3@herts.ac.uk

Abstract. A new generation of astronomical surveys will produce petabytes of data requiring new automated methods to categorise galaxies. We propose a novel unsupervised learning approach to categorise galaxies within multi-wavelength Hubble Space Telescope images. We demonstrate the efficient categorisation of galaxy structure into star-forming and passive regions and present the key steps required to produce catalogues of galaxies.

Keywords: unsupervised learning, growing neural gas, astronomical image data, image processing

1 Introduction

Astronomers seek to understand how galaxies form and evolve over time, from galaxies with very high star formation rates in the early universe, to passive galaxies producing few if any new stars today. As an initial step astronomical objects need to be identified in images, this is typically done using software such as SExtractor [1] to produce a catalogue of objects including their positions in the sky. These objects are next identified as stars or galaxies and then further categorised as different types of galaxies, e.g. star-forming or passive.

There are many existing approaches to identifying and categorising galaxies. However, the introduction of a new generation of telescopes, that will generate vast amounts of data, requires a new automated approach. Machine learning is a possible solution to this problem. Machine learning techniques are already applied to astronomical images, however these predominantly use supervised learning. Recent examples are the use of multiple layers of convolutional neural networks to classify galaxies [2] (Galaxy Zoo) and random forests used on images from the Pan-STARRS telescope to identify temporary or transient features [3]. Both of these approaches require preprocessed and labelled images. In the case of Galaxy Zoo input data consisted of 67,000 galaxy images classified by citizen scientists over a period of years.

However, in order to cope with orders of magnitude more data from next generation surveys, we need to adopt an unsupervised approach which removes

the human element from the categorisation process by identifying the natural structure within data. There is little published research on the application of unsupervised machine learning techniques to astronomical data, however, one recent example [4] uses computer vision techniques to identify galaxy types. This research still requires a pre-created catalogue of galaxy images, where each image contains a single, centred galaxy. In our work we apply unsupervised algorithms directly to large survey images containing thousands of galaxies, thus avoiding the need for preprocessed single galaxy images.

This paper is organised as follows, in Section 2 we introduce the Hubble Space Telescope survey image data. In Section 3 we introduce our methodology which includes the use of digitisation for partitioning feature data, the Growing Neural Gas algorithm, and agglomerative clustering. In Sections 4 and 5 we present and discuss our initial results. In Section 6 we describe our plans for future work and finally in Section 7 we present our conclusions.

2 Data

We used Hubble Space Telescope (HST) data from the Frontier Fields initiative [5]. These observations are deep exposures of distant clusters of galaxies. They are freely available for download from the Hubble Frontier Fields section of the Space Telescope Science Institute website.[1]

2.1 Hubble Space Telescope Frontier Fields Images

The Frontier Fields (FF) initiative uses the HST to image six strong lensing galaxy clusters. Massive clusters of galaxies act as strong gravitational lenses, magnifying and distorting the images of more distant galaxies along the same line of sight. Often these distortions result in characteristic 'arcs' around the cluster. Lensing provides astronomers with the opportunity to study the properties of very distant galaxies in far greater detail than would otherwise be possible.

The FF images contain many types of galaxies including passive ellipticals, star-forming spiral and bar galaxies and lensed galaxies. The wavelength of light we detect from a galaxy will depend on a number of physical processes. For example, star-forming galaxies emit lots of ultraviolet and blue optical light from young massive stars. Passive galaxies on the other hand emit the bulk of their stellar light at longer optical and near-infrared wavelengths. In the FF images we typically see blue star-forming spiral galaxies and red passive elliptical galaxies in the galaxy cluster, as well as the more distant blue lensed galaxies. However, this is an over simplification as the spiral galaxies also contain red, passive regions from their central bulge. Fig 1. shows a false colour composite image that was produced using three HST image files of galaxy cluster MACSJ0416. Three major types of galaxy are annotated on the image.

The HST takes multiple images of each region of the sky using seven filters. Each filter allows only a particular section of the electromagnetic spectrum to

[1] `http://www.stsci.edu/hst/campaigns/frontier-fields/`

be imaged. Our work uses images taken using the Advanced Camera for Surveys (ACS) and the following three filters:[1]

- F435W, central wavelength of 4317.4 angstroms, range of 8780 angstroms;
- F606W, central wavelength of 5917.7 angstroms, range of 2570 angstroms;
- F814W, central wavelength of 8059.8 angstroms, range of 2870 angstroms.

Each image file contains the light detected using one filter. The HST images range in size from 10000×10000 pixels to 12300×8300 pixels.

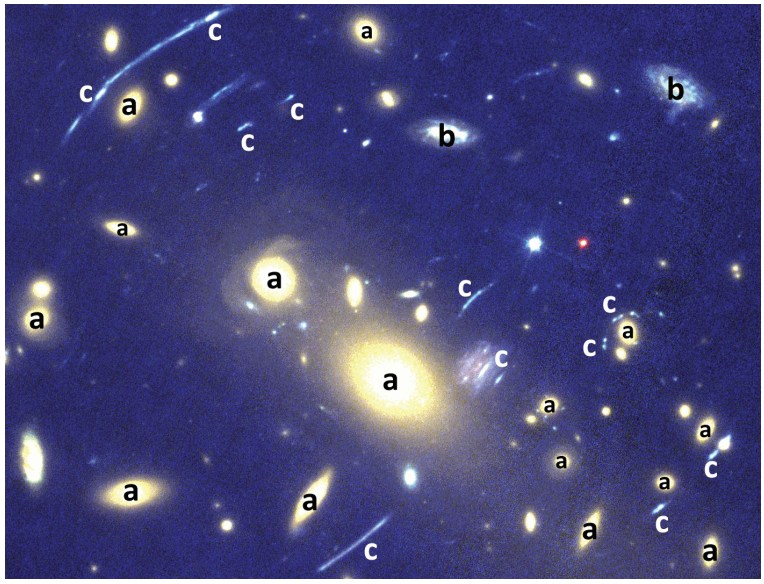


Fig. 1: This is a section of a false colour composite image of the MACS0416 strong lensing galaxy cluster. The larger objects have been annotated to identify their type. Galaxies annotated with the letter **a** are passive ellipticals, those annotated with the letter **b** are spiral galaxies, with active star-forming regions, and galaxies annotated with the letter **c** are lensed galaxies, also with active star-forming regions. In contrast to the elliptical and spiral galaxies many of the lensed galaxies have unusual morphology, appearing as lines and streaks. This is due to the galaxy cluster gravitational potential distorting space-time and altering the paths of light rays from distant galaxies. The lensed galaxy positions were identified using data from Jauzac et al [11]. This image was created using the HST image files for the F435W, F606W and the F814W filters.

[1] `http://etc.stsci.edu/etcstatic/users_guide/appendix_b_acs.html`

3 Methodology

3.1 Preprocessing

The preprocessing of the HST image data involved the creation and normalisation of the feature matrix. Details of the preprocessing are given below.

1. **Background removal** We removed the background from the three images to simplify the data and concentrate on the foreground objects (these are referred to in this paper as thresholded images). This was achieved by identifying a fixed threshold value, 5% of the root mean square of all pixel values in the image, and setting the pixel values below this threshold to zero.
2. **Feature extraction** The feature matrix was created by extracting 218,872 random sub images of a fixed size (8×8 pixels) from the red image and extracting corresponding sub images with the same location and size from the blue and green images. The Fast Fourier Transform (FFT)[6] was applied to each sub image. We then obtained the power spectrum by multiplying each complex number in the FFT output by its conjugate. The power spectrum encodes information about the distribution of power on different spatial scales. We then calculated the radial average of the pixel values by using five radial bins. This produced five values for each sub image. The feature sample was then created by concatenating the five radial average values from each sub image to form a single sample of fifteen values. Thus each sample contains features from all three images.
3. **Data normalisation** We found extreme outliers in the data, which we identified as the sub images at the centres of elliptical galaxies. These regions of the image are extremely bright relative to all other regions. On production of histograms of each feature it was clear that the features had a log normal distribution which characteristically contains extreme values. In order to convert each feature to a normal distribution, thus creating a better clustering outcome, the natural log function was applied to all values in the feature matrix. Each feature within the feature matrix was then normalised by subtracting the mean and dividing by the unit of standard deviation.
4. **Digitisation/Binning** We created a histogram of the feature matrix using twelve bins. Each value in the feature matrix was replaced with its nearest left bin edge value. This effectively 'digitised' the data. In Section 5 we discuss the Growing Neural Gas algorithm and the reason for processing the data in this way.

3.2 Model Creation Using Unsupervised Learning

The Growing Neural Gas (GNG) algorithm [7] is considered to be a good unsupervised algorithm for finding natural structure within data that does not require a preset number of clusters.[1] Initially, a graph of two nodes is created. Each of the two nodes is initialised with the values of a random sample taken

[1] `http://www.demogng.de/`

from the feature matrix. Nodes are added as the input data is processed. During this process the nodes move to map the topology of the data and the graph splits to form disconnected sub graphs, each of which represents a cluster within the feature matrix. The process continues until a stopping criteria has been met, such as the number of nodes within the graphs, or the number of the times the input data has been processed.

Applying the GNG algorithm resulted in over 7,000 clusters, making it difficult to understand the underlying structure. We therefore used agglomerative clustering [8] to merge the clusters into a more manageable number. This produced a tree structure that represents a hierarchy of merged clusters. Each node in the tree structure represents a new cluster consisting of the two hierarchical clusters with the greatest similarity. Cluster similarity was measured using average linkage and the Pearson correlation distance with an additional penalty. The Pearson correlation distance measures the similarity between the centroids of two GNG clusters. If the centroids are equivalent over the majority of the fifteen sample values then the distance is small. However, analysis of merging errors revealed that specific features were more important than others in creating meaningful clusters. Therefore clusters were only merged if the Pearson correlation distance was small and the difference in the normalised values of the first, sixth and eleventh features were less than ± 0.2.

The recursive clustering process was continued until all clusters had merged. A top down search was performed on the tree structure to identify the hierarchical clusters with a similarity greater than a threshold value. This resulted in 253 clusters with the largest 40 clusters representing over 97% of the samples.

3.3 Post Processing

In order to analyse the clusters we started with a blank image and added patches of colour (a different colour corresponding to each cluster) at the original positions of the samples. This image was then compared to a false colour RGB image, which was created by combining the original F814W, F606W and F435W HST thresholded images. This confirmed that the clusters identified by the machine learning process correspond to the distinct star-forming and passive areas in the RGB image, as illustrated in Fig 2.

4 Results

The model created by applying the unsupervised learning steps in Section 3.2 to Abell2744 was tested for its ability to generalise to other galaxy clusters, using MACSJ0416 as an example. Preprocessing steps 1 and 2 (see Section 3.1) were applied to the MACSJ0416 images as before, but in steps 3 and 4 the mean, standard deviation and bins derived from Abell2744 were used, instead of recalculating these values for MACSJ0416. The model created in Section 3.2 was then applied to the new feature matrix by using a nearest neighbour calculation with the Euclidean distance metric to identify the nearest cluster to each sample.

6

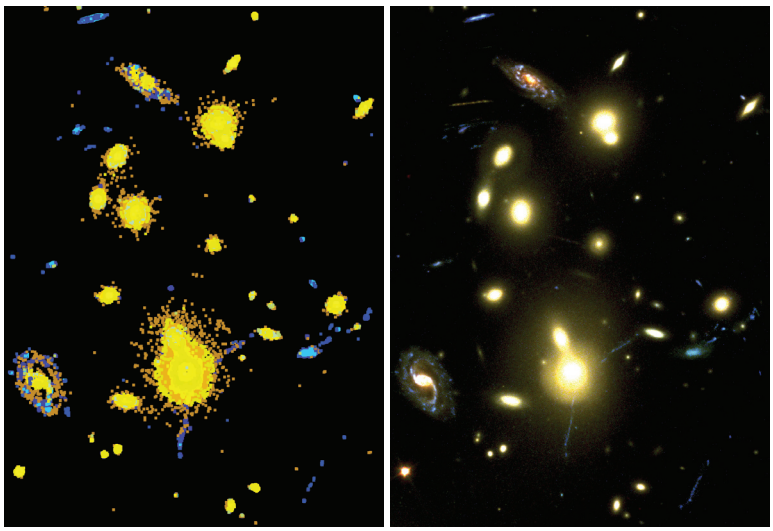


Fig. 2: A sub section of the image representing the model (left) and the thresholded HST image (right) of the galaxy cluster Abell2744. Blue colours in the processed image highlight the unsupervised clusters that represent star-forming regions in the spiral and lensed galaxies. Yellow colours correspond to the unsupervised clusters that represent passive elliptical galaxies and the central passive regions of spiral galaxies.

The results were used to create a processed image of clusters for the MACSJ0416 galaxy cluster using the same process as in Section 3.3. Fig 3. displays the results. The processed image was accurate and correctly categorised each sub image with the same colour (or cluster) that appeared in the processed image for Abell2744. Each star-forming and passive region in the processed image shows the identified clusters representing the star-forming and passive regions in the HST image.

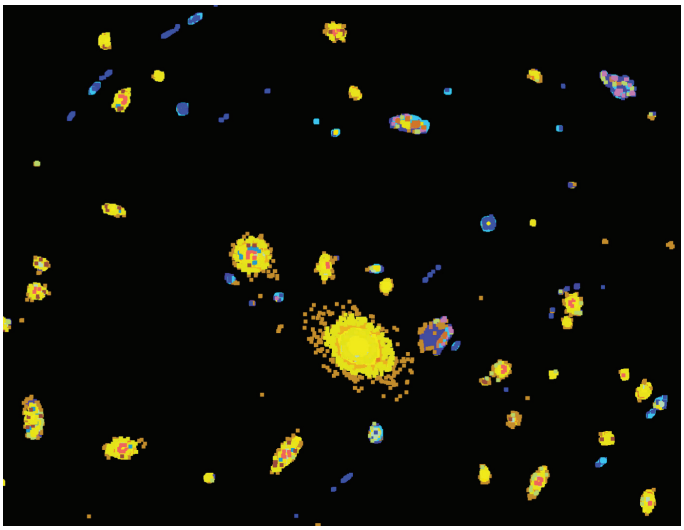
5 Discussion

Initially our preprocessing steps did not include thresholding the HST images, converting the feature matrix from log normal to a normal distribution, or the binning of the feature matrix. Without these three steps the clusters identified by the GNG algorithm were limited to one large cluster representing over 95% of the samples and a large number of very small clusters representing the remaining samples. Upon investigation we discovered that this was the result of the features having a log normal distribution, combined with the use of z-score normalisation. We applied the natural log function to the feature matrix which resulted in an improved distribution of values, however, the GNG algorithm continued to produce a similar clustering output.

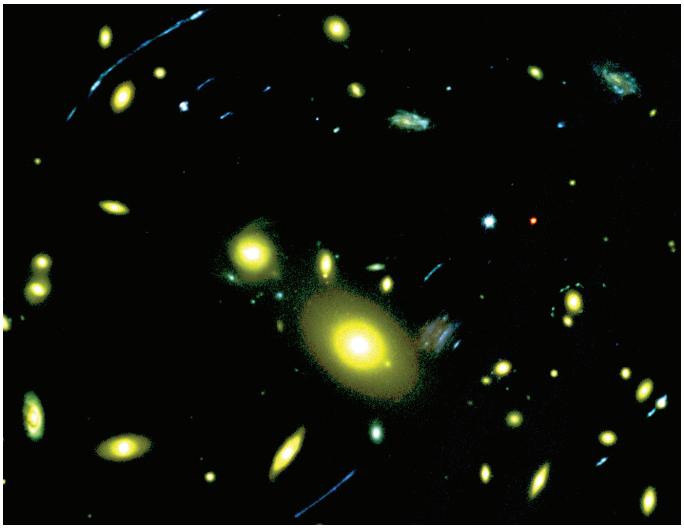
Further investigation, using Principle Component Analysis to project the data to a three dimensional space, showed that the data were distributed without any clear, spatially separated clusters. To make the data more discrete and thus help the GNG algorithm find separate clusters, the spectral values were binned into twelve bins. This effectively digitised the feature matrix into small, spatially separated clusters of samples, thus effectively partitioning the data. We found that the optimum number of bins, producing the most even distribution of clusters, was twelve. The final result, after applying the natural log function and adding the binning process, enabled GNG to effectively cluster the feature matrix, but resulted in a very large number of clusters (this was addressed using agglomerative clustering as described in Section 3.2). The application of the binning process appears to remove data by homogenising values in the feature matrix that are similar. The impact of doing this is not clear and will be investigated in future work.

We originally used average linkage and the Euclidean distance metric in our agglomerative clustering. However, we found that many of the GNG clusters were incorrectly merged and we therefore adopted the Pearson correlation distance with a penalty, as discussed in Section 3.2.

We experimented with a range of sub image sizes. When using the larger sizes GNG was not able to distinguish between small galaxies and the background. We found that using smaller sub images solved this problem, however if they were too small the number of samples became unmanageable. Sub images of 8×8 pixels provided the optimum solution.



(a) Processed image of the MACSJ0416 galaxy cluster.



(b) HST composite RGB image of the MACSJ0416 galaxy cluster.

Fig. 3: The processed image at the top displays the result of applying the model to HST images of galaxy cluster MACS0416. The bottom image shows the equivalent RGB image for comparison.

6 Future Work

We have successfully identified clusters of sub images that are meaningful. For example, we find that the centres of elliptical galaxies and the extended stellar halos of elliptical galaxies are represented by different clusters and therefore the clusters represent components of galaxies. However, we have not been able to identify individual clusters that exclusively represent lensed galaxies. The clusters that form lensed galaxies also form parts of spiral and small star-forming galaxies. This is not unexpected as these are all star-forming regions. Future work will add features based on galaxy shape to assist the identification of clusters that form lensed galaxies.

In this preliminary study we have not fully evaluated the effects of varying parameter values. In future work we will perform a more thorough evaluation of parameter values to identify how robust the model is to parameter changes. Additional future work will also involve combining sub images to achieve object detection and galaxy categorisation. To achieve this we will use a connected component labelling technique [10] adapted to use the cluster data generated in Section 4. This approach uses the sample positions from Section 3.1 to identify groups of overlapping sub images. Each group is given a unique label. For some types of galaxies this process is straightforward as they consist predominantly of a unique combination of clusters. For example, elliptical galaxies consist of a brighter, passive central region and a fainter, passive outer region. However, where there are overlapping galaxies, they will be separated by using the modified connected component labelling algorithm a second time, applying it to selected clusters within these galaxies. We will then be able to produce a catalogue of the galaxies in the HST images, including type, shape and position. Such catalogues are an important tool for astronomers.

We aim to apply our method to additional astronomical data sets, in particular test data from the next generation Large Synoptic Sky Telescope (LSST). This telescope will image the entire sky every few nights producing petabytes of multi-wavelength image data over the course of its lifetime [9] and so requires an automated process for classifying galaxies.

7 Conclusion

We have successfully used unsupervised machine learning to create a model that identifies star-forming and passive areas in HST Frontier Field images. We have also successfully validated the model's ability to generalise to one other HST image. Further work is required to combine the identified clusters to detect galaxies and catalogue their positions and to perform a more thorough evaluation of parameter values to identify how robust the model is to parameter changes.

References

1. Bertin, E. Arnout, S.: SExtractor: Software for source extraction. In *Astronomy and Astrophysics Supplement.* 117, 393-404 (1996)
2. Dieleman, S., Willet K., Dambre, J.: Rotation-invariant convolutional neural networks for galaxy morphology prediction. In *Monthly Notices of the Royal Astronomical Society.* 450, 1441-1459 (2015)
3. Wright, D. E. et al.: Machine learning for transient discovery in Pan-STARRS1 difference imaging. In *Monthly Notices of the Royal Astronomical Society.* 449, 451-466 (2015)
4. Schutter, A., Shamir, L.: Galaxy morphology - an unsupervised machine learning approach. In *Astronomy and Computing.* [arxiv 1505.04876] (2015)
5. Atek, H. et al.: Probing the z>6 Universe with the First Hubble Frontier Fields Cluster A2744. In *The Astrophysical Journal.* 786, Issue 1 Article 60 (2014)
6. Cooley, James W. Tukey, John W.: An algorithm for the machine calculation of complex Fourier series. In *Mathematics of Computation.* 19, 297-301 (1965)
7. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. In *Advances in Neural Information Processing Systems.* 7, (1995)
8. Hastie, T., Tibshirani, R., Friedman, J.: *Elements of Statistical Learning.* 520-528 (2009)
9. Ivezic, Z. et al.: LSST: from Science Drivers to Reference Design and Anticipated Data Products. [arXiv 0805.2366v4] (2014)
10. Shapiro, L., and Stockman, G.: *Computer Vision* 69-73 (2002)
11. Jauzac, M. et al.: Hubble Fronter Fields: A High-Precision Strong-Lensing Analysis of Galaxy Cluster MACSJ0416.1-2403 Using 200 Multiple Images. In *Monthly Notices of the Royal Astronomical Society* 443 (2), 1549-1554 (2014)

Web User Short-term Behavior Prediction: The Attrition Rate in User Sessions

Ondrej Kaššák, Michal Kompan, Mária Bieliková

Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

Ilkovičova 2, Bratislava, Slovakia

{name.surname}@stuba.sk

Abstract. User behavior on the Web is subject to regular patterns on various levels of granularity and periods of time, which means that it can be to some scale predicted. Most current approaches focus on predicting the user long-term behavior, which is quite stable and essentially not a subject to the short-term perceptions. In some tasks, however we are primarily interested in the user's short-term behavior, which requires considering different characteristics as in standard long-term tasks. An example of such a task is the prediction of attrition rate in a user browsing session. In our work we focus on online prediction with considering the behavior change in short time. We aim at predicting the user session end. In our actual work we proposed a classification approach that considers various user behavioral characteristics. The classifier is based on personalized prediction models trained individually for every user. We evaluated proposed approach in the e-learning domain, which is characteristic with short to middle-time browsing sessions. We present three open problems, which further work up our proposed classifier. We aim to reduce the cold start problem or more specifically to focus on prediction of browsing session end for occasional users and also experiment in different domains.

Keywords: user behavior prediction, short-term behavior, browsing session, attrition rate, cold start problem

1. Introduction and Related Work

User behavior on the Web is more than a random browsing between sites and their pages. As a user acts to fulfill his information needs subject to his actual context (e.g., knowledge, habits), his future behavior is in some scale predicable. This provides a great opportunity to improve a Web site and user browsing experience for example by personalization of the site.

The user behavior on the Web is typically mined from his actions and feedback, while it can be extracted on various levels according to its future usage. There are two levels of user behavior recognized – the long- and the short-term. The long-term be-

havior represents user stabile interests, preferences or regular behavior patterns, which are typically stored in user models [1]. As the long-term behavior changes in time only gradually it is nowadays widely used for the personalization.

In the opposite, the short-term behavior represents the user's actual interest, subject to actual task, context etc. As these factors are very difficult to describe, the short-term behavior is only hardly predictable, which makes the prediction task challenging. Short-term user's behavior changes quite dynamically so it is obviously handled on the level of raw user actions (e.g. Web page visits).

The long-term behavior is often used for the personalization in tasks such as personalized recommendation [2] (sometimes combined with short-term behavior [3]) or user attrition rate [4]. Except these task, often it is suitable to primarily use the short-term behavior (e.g., the chance that the user will end the session in next few actions or will buy premium content after hitting the paywall).

Our aim is to predict the user browsing session end, we formulated the hypothesis as '*User browsing session end can be predicted on time*'. This can be considered as the short-term behavior prediction task. For such tasks methods of supervised machine learning are typically used (e.g., classification, decision trees or neural networks) [5]. As there is often a need for processing of large data volumes, it is suitable to process them as a data stream [6]. Nowadays there can be seen a trend of usage of machine learning approaches processing data streams [7].

More specifically, in our research we focus on the task of prediction whether a user will end the browsing session within the next few actions [8]. Similarly – an attrition [9] or a churn rate [4] are typically predicted for the long-term user activities (e.g. customer loss, course dropout) in domains such as telecommunication [10], retail banking [9] or e-learning [11].

In our research, we deal with short time user activities - the browsing session. Browsing session is defined as a set of user actions (e.g. page visits) which he performs to reach one concrete task on the Web (e.g., find some information) [12]. The actions relate together, they are accomplished in similar circumstances and context. Such session meets the conditions to be a short-term behavior. To our best knowledge, there are no works on the scale of short-time behavior prediction.

2. User Browsing Session End Prediction Approach

To be able to predict the session end it is necessary to consider also domain characteristics. Our first proposal is within e-learning domain [8]. Whenever an e-learning system enrolls large number of students, there is needed to process a high volume of data that come in the form of data stream, which specifies the task to the sequential data processing.

The data stream is represented by users' visits of the Web pages, while these actions were described by various attributes (directly logged and also derived by us). An examples of such attributes are Web page details (e.g., course, type, difficulty), timestamp (e.g., day in week, hour in day), attributes describing the session (e.g., order of visited page in session, time spent in session before visit of current page), attributes

describing user habits (e.g., average session length) or attributes describing anomalies (e.g., flag if is current session longer than average, information how much is session longer or shorter than average).

As the user's preferences and behavior change in time (course beginning, before/ after the exam), it is important to be able to continuously react to these changes [13]. In the case of binary classification task (leave the site in next action vs. stay on the site), often a problem of unbalanced classes occurs [14], while the most often used techniques to reduce it are the oversampling of a rarer class, undersampling a majority class and assigning the different importance to observations [15].

To reduce these problems, we proposed a classification approach using polynomial classifier with stochastic gradient descent algorithm (as a representative of learning algorithms) to learn the attributes importance (Fig. 1).

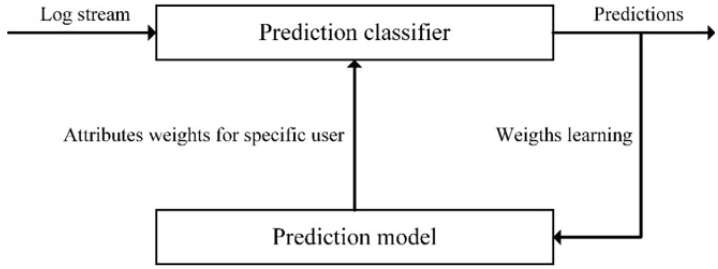


Fig. 1. Principle of used polynomial classifier [8]

The input of the classifier is a *stream of logs* represented by user actions (visits of Web pages/learning objects) described by 12 attributes as a visit timestamp, learning object (page) type, course, difficulty etc. In addition, we derived another 76 attributes as a session describing attributes (e.g., order of visit in session, time spent), advanced time attributes (e.g., week or weekend, hour, flag if is semester or exams time) and behavior describing attributes (e.g., user's average session length, flag if is current session longer than average, number of learning objects visited in last week, month). For every attribute there are considered also its squared and cubic powers, which makes 264 attributes in total.

The *predictions* are realized by a polynomial stochastic *prediction classifier*, which for every user action decides if it is the last one in user browsing session or not. The classifier processes data as a stream in one iteration, which enables an online processing of large data volumes. The classification process bases on attributes describing user actions. For the prediction we consider these attributes each supplemented with its *weight* (expressing its importance). In case of a positive result, the user action is classified as an end of the session, otherwise as continuous browsing.

The *attributes weights* are persisted in the *prediction model*. For a new user, they are initialized to zero and they are adjusted every time the classifier makes an incorrect prediction. This process of *weights learning* is controlled by a stochastic gradient descent algorithm, which allows to response to possible changes in the data character-

istics in time and keeps the *attributes weights* in the *prediction model* always up-to-date.

The data are composed of unbalanced amount of observations in individual classes (leave the session in next action vs. stay on the Web site). In our e-learning system the users in average visit 14 Web pages in a session, which gives the ratio around 13 observations of session continue to 1 session leave action. To balance this inequality we assigned the different importance for user actions based on their class (leave vs. stay) in the process of attributes weights learning.

The process output is for every user action a prediction if the user will leave the session in his next action or if he will stay and go to another learning object (Web page).

Evaluation of proposed approach was realized in e-learning system ALEF [16] on dataset consisting of 452,000 user actions from 882 users (average of 512 actions per user) captured in 5 courses for 3 years. In the first step proposed approach did not overcame the random classification significantly due to an extensive heterogeneity in users' behavior. After identification of this fact we have extended the original model storing classifier attributes weights globally by the multiple models considering weights individually for every user. It showed up that for the different users the different attributes are important, what was the reason why global variant did not worked very well. After this personalizing step it was possible to predict the user session end more precise (prediction precision = 66.5%).

The last action before the leave can be however too late for offering some changes for the user. For this reason we explored also possibilities for the prediction of leaving the session within few steps in advance. At first, we experimented with the time aspect in the mean of predicting if the user will leave the session within next 5, 10, 15 or 30 seconds. This increased the precision (precision = 78.3% for 30 seconds window), but as we did not have an information about pages content and we do not know how much time will the user spend on individual pages, we were not able to use all potential of the time aspect. For this reason we focused on "Last-N" user actions remaining until the session end. This showed as the promising way, we reached the precision = 83.4% for prediction that the user will end the session within next two actions and even 93.5% for next three actions.

The information that the user will leave the Web site within few next steps is more useful than the information about the leave in the nearest step. It gives us the chance to offer the user reasons why to stay on the site longer or to return in near future (in e-learning system it can be offering of the learning materials user did not read yet, in e-shop the discount coupon or interesting goods to buy).

3. Open problems

In our actual work [8] we proposed the approach of Web user session end prediction, which we evaluated in domain of e-learning. We showed that the end of the user browsing session can be predicted with reasonable accuracy. There however occurs a problem that users do not perform sufficient amount of actions on the Web site [17].

The reason is that a majority of Web site users are occasional or new ones, while both these types are characteristic by insufficient amount of actions made. Both of these user types are common in domains such as news domain where we plan to evaluate the approach next.

3.1. User Session End Prediction – occasional and new users

For occasional and also for new users it is difficult to predict their future behavior, because a classifier typically needs to process some amount of actions to train the prediction model used by the classifier for the user and to be able to predict his behavior satisfactorily.

The reason why focus on the occasional and new users and not only the active ones, is that they represent a great potential for the site and its visit traffic. The users who already visit the site often will not start to visit it rapidly more regardless to the site improvements. In the opposite, attracting occasional and new users (e.g., by personalized recommendation offered in a moment of the predicted session end) can persuade them to stay on site or visit it in the near future again.

Our idea is to train the classifier model for these users based on actions of similar users, who made sufficient amount of actions or based on external data source. Finding the similar users is for occasional and new users possible sooner than the training of the classifier models, because it requires the lower amount of actions. The training using external data sources is based on an assumption that if users (occasional and active one with sufficient amount of activities) behave similarly on some other Web site, their behavior can be similar also on the Web site where we actually predict.

To be able to evaluate this idea of comparing the user similarity based on data from external sources, we plan to use the dataset describing the behavior of e-learning system students on the wide Web [19]. As this data contain the behavior of the same users as we used in previous evaluations (users of e-learning system ALEF [16]), the results will be appropriate to compare with our actual results described in [8]. To generalize the results there is available another dataset from system BrUmo describing user behavior on the Web [20]. We plan to use it for comparison of user behavior similarity across multiple Web sites.

3.2. Classifier Model Training – The Cold Start Problem Overcome

An estimation of number of actions required for new user to overcome the cold start problem is subject to the two contradictory requirements. The first one is that it is useful to collect as much of the user's activity as possible. The second one is that if the user has to make too many actions before he get some benefit in return (e.g., well-tailored recommendation) it is likely that we will lose him, while the chance of losing the user grows with every next action made. For this reason it is important to reward the user as soon as possible.

To find out a balance between these two aspects, we explored the information value gain caused by users' activity increase [18]. The gain was quantified as the percentage of users, who were assigned, after some amount of their actions considered,

into the same cluster of similar users as after all their actions (available in the MovieLens dataset[1]). As we found out on the datasets from various domains (movies, short texts[2]), the very first actions bring an exponential growth of information gain, but after several actions considered it slows down to the linear growth. We consider this point, for the moment of cold start overcame (Fig. 2).

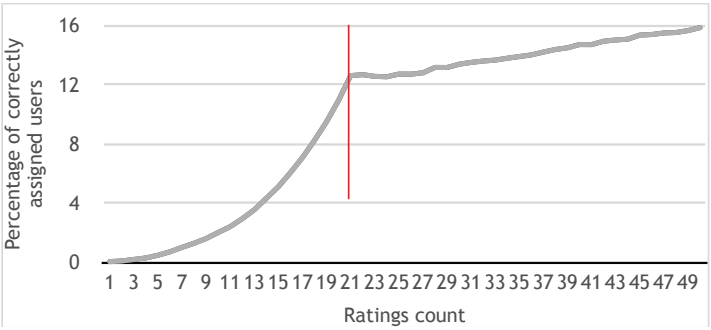


Fig. 2. An example of information gain (percentage of correctly assigned users) reached from first user actions - Movielens dataset [18]

3.3. Classifier Model Training – External Source User Similarity

User behavior on the Web site is highly influenced by the site structure [21]. This means that we suppose that the reason for similar behavior of two users on a Web site can be caused by the fact they have only limited possibilities of how to perform some activities, not by their similar preferences. For this reason it is suitable to include external data sources and compare users' behavior on the multiple Web sites for better similar users search. This kind of information can help for a new and for an occasional user. As the new user will come on Web site where the classifier predicts his behavior in sessions, its model can be immediately trained as to the user who behaves similarly on the external Web site.

The situation is in case of occasional users similar with an only exception that the model would not be initially trained but only updated. As the user reach the Web site only occasionally and the data characteristics change in time, his classifier model weights can become outdated. For this reason we propose to update user's classification model as over the time updated models of his similar users.

[1] ttp://grouplens.org/datasets/movielens/

[2] http://www.ieor.berkeley.edu/ goldberg/jester-data/

4. Conclusions

In our research in the area of user short-term behavior analysis we focus on the prediction of user session end. We aim at prediction of user browsing session end within the next few actions (the attrition rate for the Web sessions). Our proposed approach considers various user behavioral characteristics. Due to the extensive heterogeneity identified in user behavior we proposed the classifier based on personalized prediction models trained individually for every user.

We evaluated our proposed approach in the e-learning domain, where it was able to predict browsing session, especially within the next 3 actions (precision = 93.5%). To be able to generalize and validate proposed approach results to multiple different Web sites (e.g., with majority of occasional users), there is need of considering the multiple characteristics as for example the prediction model training (cold start problem for new users) or the prediction model update (problem of occasional users). We plan to evaluate our idea of comparing the user similarity based on data from external sources, on the dataset of e-learning system students' (same users as in [8]) behavior on the wide Web [19]. Similarly we plan to evaluate this idea on dataset from system BrUmo describing user behavior on the several Web sites [20].

Next we plan to train the proposed approach also for other short-term behavioral prediction tasks. We currently work on analysis of news portal data. Within this dataset we plan to use proposed approach for the task of the customer conversion rate (estimation if user will buy the premium content after hitting the paywall in the browsing session, which lock all the content on site for him if he did not paid for premium content yet, e.g., on the news portal). This data indicate similar characteristics (extensive streams of actions, imbalanced ratio between numbers of users who bought the premium content and who does not, changing data characteristic according to marketing campaigns etc.).

Acknowledgement. This work is partially supported by grants No. KEGA 009STU-4/2014 – Virtual Learning Software Lab for Collaborative Task Solving and grant No. VG 1/0646/15 – Adaptation of access to information and knowledge artifacts based on interaction and collaboration within web environment.

References

1. Senot, C., Kostadinov, D., Bouzid, M., Picault, J., Aghasaryan, A., Bernier, C. :Analysis of strategies for building group profiles, In: Proc. of the Int. Conf. on User Modeling, Adaptation and Personalization – UMAP'10, LNCS, vol. 6075, pp. 40–51, Springer, Heidelberg (2010)

2. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: Recommender Systems Handbook, Springer-Verlag., New York, NY, USA, (2010)

3. Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., Sun, J.: Temporal recommendation on graphs via long- and short-term preference fusion, In: Proc. of the 16th ACM SIGKDD int. Conf. on Knowledge discovery and data mining - KDD'10, pp. 723-731 (2010)

4. Miguéis, V. L., Camanho, A., e Cunha, J.F.: Customer attrition in retailing: An application of multivariate adaptive regression splines, In: Expert Systems with Applications, vol. 40, no.16, pp. 6225-6232 (2013)

5. Peña-Ayala, A.: Review: Educational data mining: A survey and a data mining-based analysis of recent works, In: Expert Syst. Appl., vol. 41, pp. 1432-1462 (2014)

6. PhridviRaj M.S.B., GuruRao, C.V.: Data mining–past, present and future–a typical survey on data streams, In: Procedia Technology, vol. 12, pp. 255-263 (2014)

7. Bifet, A., Holmes, G., Kirkby R., Pfahringer, B.: Moa: Massive online analysis, In: The J. of Machine Learn. Res., vol. 11, pp. 1601-1604 (2010)

8. Kaššák, O., Kompan, M., Bieliková, M.: Students' Behavior in a Web-Based Educational System: Exit Intent Prediction, Submitted to Engineering Applications of Artificial Intelligence Journal.

9. Li, F., Lei, J., Tian, Y., Punyapatthanakul, S., Wang, Y.J.: Model selection strategy for customer attrition risk prediction in retail banking, In: Proc. of the 9th Australasian Data Mining Conf., pp. 119-124 (2011)

10. Wojewnik, P., Kaminski, B., Zawisza M., Antosiewicz, M.: Social-Network Influence on Telecommunication Customer Attrition, In: Agent and Mul.Agent Sys. Tech, and App., vol. 6682, pp. 64-73 (2011)

11. Sherif, H., Greene, D., Mitchell, J.: Dropout prediction in MOOCs using learner activity features, In; Experiences and best practices in and around MOOCs, vol. 7 (2014)

12. Spiliopoulou, M., Mobasher, B., Berendt, B., Nakagawa, M.: A framework for the evaluation of session reconstruction heuristics in web-usage analysis, In: Informs J. on Computing, vol. 15, no. .2, pp. 171-190 (2003)

13. Yu, C.H., DiGangi, S., Jannasch-Pennell, A., Kaprolet, C.: A data mining approach for identifying predictors of student retention from sophomore to junior year, In: J. of Data Science, vol. 8, pp. 307-325 (2010)

14. Sun, Y., Wong, A.K., Kamel, M.S.: Classification of imbalanced data: A review, In: Int. J. of Patt. Rec. and Art. Intell, vol. 23, no.4, pp. 687-719 (2009)

15.Bottou, L.: Stochastic gradient descent tricks, In: Neural Networks: Tricks of the Trade, LNCS, vol. 7700, pp. 421-436 (2012)
16.Šimko, M., Barla, M., Bieliková, M.: ALEF: A framework for adaptive web-based learning 2.0, In: Proc. of IFIP Advances in Information and Communication Technology, 2010, Springer, vol. 324/2010, pp. 367-378 (2010)
17.Wang, W., Zhao, D., Luo, H., Wang, X.: Mining User Interests in Web Logs of an Online News Service Based on Memory Model, In: Proc. of the 8th Int. Conf. on Networking, Architecture and Storage, pp. 151-155 (2013)
18.Višňovský, J., Kaššák, O., Kompan, M., Bieliková, M.: The Cold-start Problem: Minimal Users' Activity Estimation, In: 1st Workshop on Recommender Systems for Television and online Video (RecSysTV) in conjunction with 8th ACM Conf. on Recommender Systems, p. 4 (2014)
19.Labaj, M., Bieliková, M.: Conducting a Web Browsing Behaviour Study – An Educational Scenario, In: Proc. of SOFSEM 2015: Theory and Practice of Computer Science, LNCS, vol. 8939, pp. 531-542 (2015)
20.Šajgalík, M., Barla, M., Bieliková, M.: Efficient Representation of the Lifelong Web Browsing User Characteristics, In: UMAP 2013 Extended Proceedings, 2nd Int. Workshop on LifeLong User Modeling, LLUM 2013. CEUR, vol. 997 (2013)
21.Yang, Y.C.: Web user behavioral profiling for user identification, In: Decision Support Systems, vol. 49, pp. 261-271 (2010)

Semi-Supervised Learning of Event Calculus Theories

Nikos Katzouris[1,2], Alexander Artikis[3,2], and Georgios Paliouras[2]

[1] Dept. of Informatics & Tel/coms, National & Kapodistrian University of Athens
[2] Institute of Informatics & Tel/coms, NCSR "Demokritos"
[3] Dept. of Maritime Studies, University of the Piraeus
{nkatz,a.artikis,paliourg}@demokritos.iit.gr

Abstract. Learning programs in the Event Calculus with Inductive Logic Programming is a challenging task that requires proper handling of negation and unobserved predicates. Learners that are able to handle such issues, typically utilize abduction to account for unobserved supervision, and learn by generalizing all examples simultaneously to ensure soundness, at the cost of an often intractable search space. In this work, we propose an alternative approach, where a semi-supervised framework is used to obtain the unobserved supervision, and then a hypothesis is constructed by a divide-and-conquer search. We evaluate of our approach on a real-life, activity recognition application.

Keywords: Event Calculus, Inductive Logic Programming, Semi-Supervised Learning

1 Introduction

The Event Calculus [10] is a temporal logic for reasoning about events and their effects. Over the past few years, it has been used as a reasoning engine in large-scale applications, motivating research related to scalable inference [1], uncertainty handling [17] and learning [9].

Learning Event Calculus programs with Inductive Logic Programming (ILP) [5] has two major difficulties: First, it is a non-Observational Predicate Learning (non-OPL) [12] task, meaning that target predicates differ from the ones used to record the examples. Second, it requires handling of Negation as Failure (NaF) during learning, which the Event Calculus uses to model persistence of properties over time. Recently, a number of ILP systems, such as XHAIL [15] and TAL-RASPAL [2] have been introduced, that are able to address these problems by combining ILP with Abductive Logic Programming (ALP) [6]. ALP allows hypothesizing with unobserved knowledge, thus solving non-OPL, and has a non-monotonic semantics, allowing reasoning under NaF. The above-mentioned systems generalize all available examples simultaneously, aiming for soundness, which, in the presence of NaF, is not ensured by set-cover approaches [15]. The price is that with a sufficient amount of data, the complexity of theory-level search results in an intractable search space.

In practice, the requirement for soundness is often relaxed to account for noise in the data. In this case, the expensive theory-level search is no longer necessary. In contrast, a set-cover search could be adopted, to scale-up learning. However, mainstream set-cover systems, like Progol [12] and Aleph[4] cannot adequately address non-OPL, especially in the presence of NaF [14]. A plausible workaround would be to use ALP to solve the non-OPL problem, by abductively acquiring supervision in terms of target predicates, and then pass this supervision to a set-cover learner. A problem with this approach is that the supervision acquired via ALP is often too few to learn something meaningful in a set-cover fashion. This is because ALP systems are typically biased towards finding the smallest/simplest explanation of the observations, resulting in a very small number of positive target predicate instances.

To address the problem, we propose a semi-supervised learning setting. A small amount of unobserved supervision is acquired via ALP. The input examples serve as unlabelled (w.r.t. the target predicates) instances, and a k-NN classifier is used to label them. We use Aleph for learning, and compare this approach to the XHAIL system on an activity recognition application. Our results indicate comparable hypotheses.

The rest of this paper is structured as follows. In Section 2 we present the basics of Event Calculus, ILP and ALP. In Section 3 we discuss in more detail the problem addressed in this work, while in Section 4 we present our approach. In Section 5 we present the experimental evaluation, while in Sections 6 and 7 we discuss related work and draw our main conclusions.

2 Background

Event Calculus. The ontology of the Event Calculus comprises *time points* represented by integers; *fluents*, i.e. properties which have certain values in time; and *events*, i.e. occurrences in time that may affect fluents. The axioms of the formalism incorporate the common sense *law of inertia*, according to which fluents persist over time, unless they are affected by an event. In this work we use a simplified version of the Event Calculus, which we henceforth denote by EC. The axioms of The EC are presented below. Following Prolog's convention, predicates and ground terms in logical formulae start with a lower case letter, while variable terms start with a capital letter. Also *not* denotes Negation as Failure.

$$holdsAt(F, T+1) \leftarrow \quad initiatedAt(F, T). \tag{1}$$

$$holdsAt(F, T+1) \leftarrow \quad holdsAt(F, T), \quad not\ terminatedAt(F, T). \tag{2}$$

Axiom (1) states that a fluent F holds at time T if it has been initiated at the previous time point, while Axiom (2) states that F continues to hold unless it is terminated.

[4] `http://www.cs.ox.ac.uk/activities/machlearn/Aleph/aleph.html`

Example Time 1:	**Example Time 2:**
$happensAt(walking(id_1), 1)$,	$happensAt(walking(id_1), 2)$,
$happensAt(walking(id_2), 1)$,	$happensAt(walking(id_2), 2)$,
$coords(id_1, 201, 454, 1)$,	$coords(id_1, 201, 454, 2)$,
$coords(id_2, 230, 440, 1)$,	$coords(id_2, 227, 440, 2)$,
$direction(id_1, 270, 1)$,	$direction(id_1, 275, 2)$
$direction(id_2, 270, 1)$,	$direction(id_2, 278, 2)$
$\boldsymbol{not\ holdsAt(moving(id_1, id_2), 1)}$	$\boldsymbol{holdsAt(moving(id_1, id_2), 2)}$

Table 1. Two examples from the domain of activity recognition.

Inductive Logic Programming. An ILP algorithm assumes a set of positive (E^+) and negative (E^-) examples and some background knowledge B. From that, it constructs a clausal theory H (inductive hypothesis) that along with B logically entails (*covers*) the examples, i.e. $B \cup H \models E^+$ and $B \cup H \not\models E^-$.

Abductive Logic Programming. An ALP algorithm assumes a set of observations E, a background theory B and a set of abducible predicates A. From that, it derives a set Δ (abductive explanation) of ground atoms, such that $B \cup \Delta \models E$ and each predicate in Δ appears in A.

The learning setting. We use an example from an activity recognition application, as defined in the CAVIAR[5] project, to illustrate our learning setting. The CAVIAR dataset consists of videos of a public space, where actors perform some activities. These videos have been manually annotated by the CAVIAR team to provide the ground truth for two types of activity. The first type corresponds to a person's activities at a certain time point (short-term activities), for instance *walking*, *running* and so on. The second type corresponds to activities that involve more than one person (long-term activities), for instance two people *moving together*, *fighting*, *meeting* and so on.

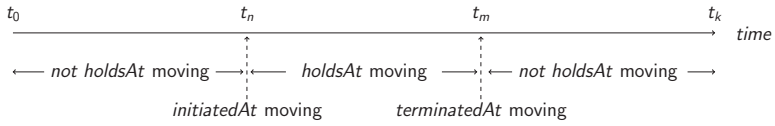
Table 1 presents two training examples for the *moving together* long-term activity, for time points 1 and 2 respectively. Each example is a "snapshot" of the domain, and consists of the *annotation* and the *narrative*. The annotation, shown in bold in Table 1, specifies whether *moving together* between two persons holds at the particular time point. The narrative consists of a person's short-term activity, in addition to other spatial knowledge, such as (x, y) coordinates and direction. Based on the annotation, an example is negative (example at time 1, Table 1), or positive (example at time 2, Table 1). Negative annotation is generated via the Closed World Assumption.

From such training instances, and using the EC as background knowledge, the goal is to derive conditions under which long-term activities are initiated or terminated. We thus wish to learn rules with *initiatedAt/2* and *terminatedAt/2* in the head, making our problem non-OPL, since the annotation is given in *holdsAt/2* predicates.

Non-OPL handling. Hybrid learners that combine ILP & ALP can naturally handle non-OPL, thanks to the ability of the underlying abductive proof procedure, to hypothesize with non-observed predicates. The XHAIL system [15]

[5] `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/`

(a)



(b)

target concept/time	$[t_0, t_n)$	t_n	(t_n, t_m)	t_m	$(t_m, t_k]$
`initiatedAt-moving`	negatives	positive	unlabelled	negative	negatives
`terminatedAt-moving`	unlabelled	negative	negatives	positive	unlabelled

Fig. 1. (a) Schematic representation of *moving* being initiated and terminated in time. **(b)** Labels of examples for the time interval presented in (a) for two target classes: `initiatedAt-moving` and `terminatedAt-moving`.

is a hybrid learner that first explains abductively the input data in terms of target predicates, and then uses the abduced atoms to generate a hypothesis. As an example, given the data in Table 1 and $A = \{initiatedAt/2\}$ as the set of abducible predicates, an abductive explanation (w.r.t. the EC) of the fact that $moving(id_1, id_2)$ holds at time 2, is that it is initiated at time 1. Hence XHAIL derives the atom $initiatedAt(moving(id_1, id_2), 1)$, which along with EC entails the examples in Table 1. Subsequently, each abduced atom in an abductive explanation of the input supervision is used to generate a *bottom clause* [12]. Collectively, these bottom clauses form the *Kernel Set* [15], as a lower bound for the hypothesis space. A hypothesis is found by searching the space of theories that θ-subsume the Kernel Set.

3 Defining the Problem

XHAIL scales poorly because it tries to cover all the examples simultaneously in order to ensure soundness [15]. However, in applications with large amounts of noisy data, the target is a less-than-perfect theory. A more practical approach to scale-up learning in such cases, would be to utilize efficient, off-the-self ILP learners, like Aleph, which constructs hypotheses one clause at a time, in a set-cover loop. However, although Aleph supports abduction, its abductive capabilities are limited and cannot be used for abduction with EC programs [14, 9]. A workaround would be to use an ALP system to acquire the missing supervision and then use this supervision with Aleph.

A problem with this idea is that the indeterminacy of abduction may result in supervision of poor quality. To illustrate the issue, assume that we are given a stream of examples, where the annotation dictates that $moving(id_1, id_2)$ *holds* in a time interval $I = [t_n, \ldots, t_m]$ (see Figure 1 (a)), and we wish to learn the conditions under which it is *initiated*. There is a multitude of alternative

initiatedAt/2 explanations of the *holdsAt/2* observations. One extreme is to assume that $moving(id_1, id_2)$ is initiated at $t-1$, for each $t \in I$. This would give a sufficient amount of *initiatedAt/2* supervision (as much as the original *holdsAt/2* one), but a large amount of it may be wrong[6]. The other extreme is to assume that $moving(id_1, id_2)$ is initiated at t_{n-1}. Note that this assumption suffices to explain all observations in the interval I, since once initiated, $moving(id_1, id_2)$ persists, by the axioms of the EC, until it is terminated. This is the simplest (and the only necessary) abductive explanation. However, this results in too few *initiatedAt/2* supervision.

We propose a semi-supervised approach to enrich the abductively acquired supervision. In the proposed setting, a minimal (simplest) abductive explanation serves as the initial positive supervision for the target concepts. For instance, in Figure 1 (b), the example at time t_n (resp. t_m) is (abductively-acquired) positive supervision for the initiation (resp. termination) of *moving*. All examples in $[t_n, t_m)$ (resp. $[t_0, t_n) \cup [t_m, t_k]$) are unlabelled examples for the initiation (resp. termination) of *moving*, that may be taken into account to improve the quality of the outcome. We next describe our approach in detail.

4 Semi-Supervised Learning

A semi-supervised learning algorithm [18] tries to use both labelled and unlabelled data to learn a predictor that predicts future instances better than a predictor learned from the labelled data alone.

A simple and widely-used approach to SSL, which we also adopt in this work, is k-Nearest Neighbor (k-NN) classification [4]. In a k-NN setting, the class of an unlabelled example is "approximated" by the class that results by voting between its k nearest, or most-similar labelled examples, according to some distance/similarity measure. Formally, if e is an unlabelled example, then using k-NN, its label is approximated by:

$$f(e) = \underset{c=1,\ldots,N}{\operatorname{argmax}} \sum_{e' \in N_k(e)} d(e, e') \cdot \delta(c, f(e')) \tag{3}$$

where f is a function that assigns labels to examples, $c = 1, \ldots, N$ is the set of available class labels, $d(x, y)$ is a distance metric, $N_k(e)$ is the set of the k *labelled* examples, nearest to e according to d and $\delta(x, y) = 1$, if $x = y$, else 0.

Distance measures used with attribute-value representations are based mostly on the euclidean distance and its generalizations. Such measures are not appropriate for logical settings, where objects are structured. In order to allow for techniques from distance-based learning to be used in a logical setting, a lot of work has been done on defining distance/similarity measures for relational data [3, 7, 16, 13, 8, 11]. In this work, we base our k-NN approach on the *Hausforff metric*, as introduced in [13].

[6] Examples of such cases are fluents that are *strongly initiated/terminated*. Consider for instance a fluent representing a ball rolling. Kicking the ball may have initiated this fluent, but the initiation condition does not hold while the fluent holds.

4.1 The Hausdorff Metric for Relational Data

A metric space $(\mathcal{X}, d)$ is called bounded if $d(x, y) \leq n$ for all $x, y \in \mathcal{X}$ and some n. The Haussdorff Metric is defined on the set of closed subsets of a bounded metric space as follows:

Definition 1 (The Haussdorff Metric). Let $(\mathcal{X}, d)$ be a bounded metric space and $\mathcal{C}(\mathcal{X})$ the set of all closed subsets of $\mathcal{X}$. Assume the following functions:

$\sigma(x, Y) = min_{y \in Y} d(x, y)$ for non-empty $Y \subseteq \mathcal{X}$ and $x \in \mathcal{X}$.
$\rho(Y, Z) = max_{x \in Y} d(x, Z)$ for non-empty $Y, Z \subseteq \mathcal{C}(\mathcal{X})$.
$h(Y, Z) = max(\rho(Y, Z), \rho(Z, Y))$ for non-empty $Y, Z \subseteq \mathcal{C}(\mathcal{X})$ with $h(\emptyset, \emptyset) = 0$ and $h(\emptyset, Z) = 1$.

$(\mathcal{C}(\mathcal{X}), h)$ is a metric space [13] called the Haussdorff Metric induced by d.

The set of expressions of a first-order language, can be equipped with the Haussdorff Metric by defining an appropriate distance function as follows:

Definition 2 (Distance function on the expressions $\mathcal{X}$ of a language). Let $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such as

1. $d(x, x) = 0$ for all $x \in \mathcal{X}$
2. If $p \neq q$ then $d(p(x_1, \ldots, x_n), q(y_1, \ldots, y_m)) = 1$
3. $d(p(x_1, \ldots, x_n), p(y_1, \ldots, y_n)) = \frac{1}{2n} \sum_{i=1}^{n} d(x_i, y_i)$.

It can be shown [13] that d is a metric and the resulting metric space is bounded by 1. Moreover, each subset of this space is both open and closed, hence the set of its closed subsets coincides with the set of all its subsets. Therefore we can define the distance between subsets of expressions by means of the Haussdorff distance. In what follows, we use the Haussdorff distance to measure the distance between two examples in the form of Herbrand Interpretations, as in Table 1.

4.2 Labelling Data

As in many ILP approaches, we assume that some language bias is provided, specifying the structure of literals and clauses. In this work we use *mode declarations* [12]. Head mode declarations represent classes in our semi-supervised setting. Ground instances of head mode atoms represent instances of the particular class. For example the atom $initiatedAt(moving(id_1, id_2), 10)$ is an instance of the corresponding class initiatedAt-moving. Such an atom is a positive example for the particular class, while negative examples are generated via the Closed World Assumption. Labelling an input *holdsAt/2* example e amounts to generating from e a ground instance of an *initiatedAt/2* or *terminatedAt/2* class (thus generating a positive example), or inferring the absence thereof, (thus implying a negative example).

A minimal abductive explanation of the input *holdsAt/2* observations serves as a small amount of initial positive supervision, as discussed in previous sections. Based on these labelled positives, and using also the labelled negatives, we

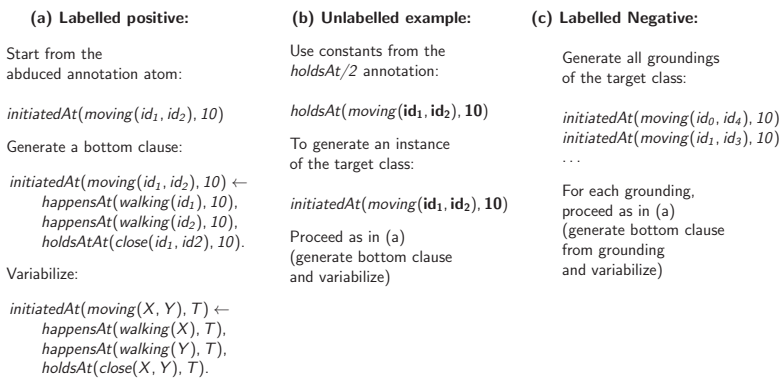


Fig. 2. Extracting structure from positive, unlabelled and negative examples in the form of bottom clauses.

produce labels for the unlabelled input examples. To do so, for each unlabelled example e_u two values are computed, $hd^-(e_u)$ and $hd^+(e_u)$, where the former is the minimum Haussdorf distance of e_u from all negatives and the latter is the minimum Haussdorf distance of e_u from all positives. If $hd^+(e_u) > hd^-(e_u)$ then e_u is labelled as a positive example and a proper instance of a target class (i.e. a ground *initiatedAt/2* or *terminatedAt/2* atom) is generated as annotation. In the opposite case, e_u is labelled as a negative example.

To calculate Haussdorf distances, clausal structure is extracted from the examples, in the form of bottom clauses [12]. Figure 2 illustrates the process for the three cases of (a) labelled positives, (b) unlabelled examples and (c) labelled negatives. In all cases, generated ground instances of the target class are used as heads for bottom clauses, thus extracting related structure (as indicated by body mode declarations) from within the Herbrand interpretation that represents the example. Haussdorf distances between examples are then computed between these bottom clauses. The Hausdorff distance between two bottom clauses is calculated on the corresponding Herbrant Interpretations that result by joining the head and the body of each clause. Since what needs to be compared for similarity is "lifted" clausal structure that may be generated from the examples, rather than specific ground instantiations, the bottom clauses are variabilized, i.e. ground terms are replaced by variables as indicated by the mode declarations (see Figure 2).

5 Experimental Evaluation

We implemented our approach using Aleph and the answer set solver Clingo[7] as the ALP component. We conducted a preliminary experimental evaluation on

[7] `http://potassco.sourceforge.net/`

	Aleph+SSL	XHAIL
Training Time (sec)	81.32 (± 19.02)	480.50 (± 94.78)
Hypothesis size	12.26 (±6.32)	7.50 (±4.34)
Precision	60.318 (±0.438)	67.145 (± 0.652)
Recall	89.545 (± 0.648)	82.254 (±0.505)
F_1-score	72.448 (± 0.367)	73.935 (±0.687)

Table 2. Comparison of Aleph with semi-supervised learning and XHAIL.

the CAVIAR dataset. As a baseline to compare with we used the XHAIL system, which, like Aleph, uses the data to guide its search in a bottom-up manner.

Experimental Setting. We used ALP to acquire the initial *initiatedAt/2* and *terminatedAt/2* supervision, from all CAVIAR videos. Note that for all videos, this supervision was too few for Aleph to learn something: Aleph returned a simple enumeration of the supervision, since given the number of positives, the gain of generalizing them, in terms of compression, was negligible. After labelling the unlabelled examples, we used Aleph to learn a theory From each video separately, for the long-term activities of *moving*, *meeting* and *fighting*. We also used XHAIL to learn a theory from each video, using only the original data (i.e. without enhancing the supervision). Each theory constructed from each system was evaluated against all other CAVIAR videos. Table 2 presents the results.

Results. Training times for XHAIL are significantly higher, as compared to those of Aleph's, due to the combinatorial complexity of XHAIL's theory level search. On the other hand, this search is responsible for the fact that XHAIL learned more compressed programs. The lower precision scores for Aleph are attributed to the fact that Aleph frequently produced over-fitted *terminatedAt/2* rules, which on unseen data, failed to capture the termination of fluents, resulting in a large number of false positives. However, some low-scale experiments on CAVIAR indicate that these scores may be improved by learning from more than one video at a time. Concerning recall, Aleph outscores XHAIL. In general, Aleph learned from a richer supervision, and produced theories that were able to better fit unseen instances, resulting in a fewer number of false negatives, as compared to XHAIL. In contrast, XHAIL learned from small (minimal) abductive explanations from each video, which resulted in missing interesting patterns, subsequently producing a larger number of false negatives on unseen data, and thus, a lower recall.

6 Related Work

A substantial amount of work on distance-based methods for relational data exists in the literature. KGB [3] uses a similarity measure, based on the comparison of structural components of logical entities, to perform generalization via clustering. RIBL [7] is a k-NN classifier that extents KGB's structural similarity measure, by taking into account the values of attributes in logical objects,

in addition to other objects that they are related to. Propagating similarity in this way, results in "indeterminacy of associations" [8], a problem with increased computational complexity. This issue is addressed by the similarity measure introduced in [8], which uses a stratification framework for level-wise comparison of objects. A drawback of all these approaches is that they assume function-free Horn logic as the representation language, and thus cannot be applied to nested representations required by EC programs. In addition, these measures are not metrics, hence they lack the desirable mathematical properties of the Hausdorff distance, used in this work.

In addition to the above-mentioned, purely syntactic similarity frameworks, semantic frameworks have also been proposed for logical distance base-learning. [16] presents a k-NN classifier that first generates a user-specified number of alternative hypotheses, and then classifies an example by voting between the hypotheses that entail the examples and those that do not. k-FOIL [3] relies on the FOIL ILP system to generate a set of rules, which are then used to define a kernel function for the example space, based on the coverage of the examples by the rules. It then tries to refine these rules, evaluating each refinement via a support vector machine trained on the current kernel. A difference of such approaches from the work presented here, is that they use the logical learning framework to construct a classifier, while we use a simple k-NN classifier to facilitate the learning framework.

7 Conclusions and Future Work

We presented a semi-supervised framework for learning Event Calculus theories. The main problem we address is to use the input, observed predicates, to acquire sufficient supervision, in terms of hidden, target predicates. To this end we use abduction to obtain a small set of target predicate instances, which serve as the initial set of positive labelled examples, and then use a k-NN classifier based on the Hausdorff distance, to obtain new instances, thus enriching the supervision. We presented a preliminary evaluation on an activity recognition application, with promising results concerning training time speed-ups and hypothesis quality, as compared to the baseline of the ILP-ALP system XHAIL. Future work involves further experimentation and assessment of more sophisticated distance measures and different semi-supervised learning settings. We also plan to combine our approach with existing techniques for learning Event Calculus programs [9].

8 Acknowledgements

This work was funded by the EU project REVEAL[8] (FP7 610928).

[8] http://revealproject.eu/

References

1. Alexander Artikis, Marek Sergot, and George Paliouras. An event calculus for event recognition. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 27(4):895–908, 2015.
2. D. Athakravi, D. Corapi, K. Broda, and A. Russo. Learning through hypothesis refinement using answer set programming. In *Proc. of the 23rd Int. Conference of Inductive Logic Programming (ILP)*, 2013.
3. Gilles Bisson. Learning in FOL with a similarity measure. In *Proceedings of the 10th National Conference on Artificial Intelligence. San Jose, CA, July 12-16, 1992.*, pages 82–87, 1992.
4. Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.
5. Luc De Raedt. *Logical and Relational Learning*. Springer, 2008.
6. Marc Denecker and Antonis Kakas. Abduction in logic programming. In *Computational Logic: Logic Programming and Beyond*, pages 402–436. 2002.
7. Werner Emde and Dietrich Wettschereck. Relational instance-based learning. In *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3-6, 1996*, pages 122–130, 1996.
8. Stefano Ferilli, Teresa Maria Altomare Basile, Marenglen Biba, Nicola Di Mauro, and Floriana Esposito. A general similarity framework for horn clause logic. *Fundamenta Informaticae*, 90(1):43–66, 2009.
9. Nikos Katzouris, Alexander Artikis, and George Paliouras. Incremental learning of event definitions with inductive logic programming. *Machine Learning*, To appear.
10. R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):6796, 1986.
11. Niels Landwehr, Andrea Passerini, Luc De Raedt, and Paolo Frasconi. kfoil: Learning simple relational kernels. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 389–394, 2006.
12. S. Muggleton. Inverse entailment and progol. *New Generation Computing*, 13(3&4):245–286, 1995.
13. Shan-Hwei Nienhuys-Cheng. Distance between herbrand interpretations: A measure for approximations to a target concept. In *Proc. 7th Int. Workshop Inductive Logic Programming*, page 213226, 1997.
14. O. Ray. Using abduction for induction of normal logic programs. In *ECAI'06 Workshop on Abduction and Induction in Articial Intelligence and Scientic Modelling*, 2006.
15. Oliver Ray. Nonmonotonic abductive inductive learning. *J. Applied Logic*, 7(3):329–340, 2009.
16. Michèle Sebag. Distance induction in first order logic. In *Inductive Logic Programming*, pages 264–272. Springer, 1997.
17. A. Skarlatidis, G. Paliouras, A. Artikis, and G. Vouros. Probabilistic Event Calculus for Event Recognition. *ACM Transactions on Computational Logic*, 16(2):11:1–11:37, 2015.
18. Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

Deep Bayesian Tensor for Recommender System

Wei Lu and Fu-lai Chung

Department of Computing, Hong Kong Polytechnic University
Kowloon, Hong Kong

Abstract. How to efficiently learn user behavior features is a key for recommendation. Tensor factorization offers a useful approach for complex multi-relational data learning and missing element completion. Targeting the volatile and sparse issue of online video recommendation, we propose a deep probabilistic tensor decomposition model for item recommendation. Extended from the Canonical PARAFAC (CP) decomposition, this method provides a fully conjugate Bayesian treatment for parameter learning. Through incorporating multi-layer factorization, a richer feature representation facilitates a better and comprehensive understanding of user behaviors and hence gives more helpful recommendations. The new algorithm, called Deep Canonical PARAFAC Factorization (DCPF), is evaluated on both synthetic and large-scale real world problems. Empirical results demonstrate the superiority of the proposed method and indicate that it can better capture latent patterns of interaction relationships.

1 Introduction

Relational data based personalized recommendation plays an essential role in today's e-commerce operations. While those emerging web sites provide services of millions of TV shows, movies, music and news clips, they are also a main source of capturing browsing or operational data for a huge amount of users. To increase user stickiness and to enhance the overall satisfaction, services working on finding the most relevant contents to users are highly desired. There are two traditional and widely applied approaches to this kind of tasks, i.e., Content-Based Filtering (CBF) and Collaborative Filtering (CF). CBF compares new information with the historical profile to predict its relevance to certain users [1]. CF recommends items based on the common preferences of a user group, without using the item attributes. Although both of them have performed superiorly well on many systems, they have drawbacks facing the challenges aroused by the increasing availability of large-scale digitized data.

Taking the online video recommender system as an example, in view of the fact that most large-scale digitized data nowadays can be regarded as multi-relational data, one may construct a three-way tensor (User $\times$ Video $\times$ Tag), which stores both user-item and item-feature (tag) interaction. Assuming that the historical behaviors of users are sound sources for preference estimation, the high-level semantic topics of video tags can be regarded as a comprehensive representation of user features. Moreover, since the tags are manually labeled, error, incompleteness and abundance exist. How to overcome these deficiencies and explore a better representation of user preference features is also an important concern of model construction. Regarding the interaction

of information, a tensor format is hence a natural and sound approach. In the proposed recommender system, we could obtain the click records of each individual active user, and add multiple categories of tags to each video in the database. So the tensor can be constructed flexibly either in a real-valued way using the number of clicks, or a binary way using the action user takes on the video.

Traditional multi-way factor models suffer from the drawback of failing to capture coupled and nonlinear interactions between entities [14]. Also, they are not robust to datasets containing noisy and missing values. Through proper generative models, non-parametric Bayesian multi-way analysis algorithms (like [5] [14] [10]) are especially appealing, since they provide efficient ways to deal with distinct data types as well as data with missing values and noises. Meanwhile, deep networks have been proved great empirical success in various domains [2]. With their capability in providing more compact nonlinear representations for feature learning, it would be interesting to adopt deep learning in one or more of the tensor modes and assess its effectiveness on tensor completion.

Motivated by the aforementioned considerations, this paper presents a fully conjugate deep probabilistic approach for tensor decomposition. Based on the Canonical PARAFAC (CP) decomposition, the proposed model is capable of clustering the three-way data along each direction simultaneously. To find a more compact representation in the latent space of each mode, a multi-layer factorization is imposed on the mode factor matrix to incorporate nonlinear mapping. As a fully conjugate Bayesian model, efficient Gibbs sampling inference is facilitated, with automatic determination of core tensor rank.

The rest of the paper is organized as follows. Related work and CP decomposition are reviewed in Section 2. A new hierarchical recommendation framework, i.e., DCPF, is introduced in Section 3 and its deep factorization is elaborated. The detailed inference mechanism using fully conjugate Gibbs sampling is also presented. Then, experiments on both synthetic and real-world scenarios are described. Performance comparison between single layer and 2-layer implementations are presented for video recommendation and tag completion. Finally, we conclude this work in Section 5.

2 Related Work

2.1 Canonical PARAFAC (CP) Decomposition

The core of our proposed model is the Canonical PARAFAC (CP) decomposition [6]. CP, as a special case of Tucker decomposition, decomposes a tensor into a sum of rank-1 components [7], as illustrated in Fig. 1. A K-mode tensor $X \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_K}$, where n_k denotes the dimension of each mode, can be expressed by:

$$X = \sum_{r=1}^{R} \lambda_r \cdot u_r^{(1)} \circ u_r^{(2)} \cdots \circ u_r^{(K)}$$

Here, we adopt the notations from [7]. The column vectors $\{u_r^k\}_{k=1}^K \in \mathbb{R}^{n_k \times 1}$ denote the latent factors for each mode, combining which forms the factor matrices $U^{(k)}$. R is

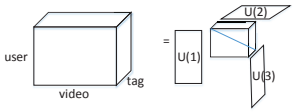


Fig. 1. CP decomposition: a three mode user-video-tag relational dataset example.

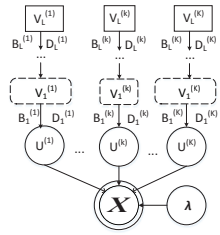


Fig. 2. Graphic model: The observed X is decomposed into a diagonal core tensor $\boldsymbol{\lambda}$ and K-mode matrices. Each factor matrix $U^{(k)}$ is further constructed through a L layer deep network .

a positive integer indicating the rank of the core tensor. λ_r is the weight associated with the r^{th} rank component and $\circ$ denotes vector outer product. In an element-wise way, the tensor element $x_{\mathbf{i}}$ with subscript $\mathbf{i} = i_1, ..., i_K$ denoting the K-dimensional index of the i^{th} entry in the observed tensor can be concisely expressed as:

$$X = \sum_{r=1}^{R} \lambda_r \prod_{k=1}^{K} u_{i_k r}^{(k)} \tag{1}$$

2.2 Multi-task Learning, Transfer Learning and Deep Learning

Learning multi-relational features can be linked with two other state-of-art machine learning methods, i.e. multi-task learning and transfer learning. Multi-task learning tries to learn multiple tasks simultaneously through uncovering the common latent features [9], while transfer learning is distinguished by removing the assumption that the training and test data are drawn from the same feature space and the same distribution. Since the essence of creative learning is to infer the unexplored feature space, both learning paradigms might be helpful for solving approach design.

Significant recent research on deep models has proved its effect on data representation. The form of the proposed multi-layer implementation is most related to [4] and [8]. The main idea is that an unsupervised deep model can be viewed as a hierarchy of factor-analysis [15], with the factor decomposition from lower layer serving as the input of deeper layer.

3 Deep CP Decomposition

3.1 Model Description

Let Y denote an incomplete K-order tensor. For different types of observation Y, we assume that there exists a latent real-valued tensor X, from which the observations are sampled from. Y and X are related element-wisely through a noise function f

[13], depending on the data type being modeled, e.g. Gaussian for real valued data, or Bernoulli-logistic for binary valued data. The goal is to infer the parameters of CP decomposition, $\boldsymbol{\lambda}$ and $\{U^{(k)}\}_{k=1}^{K}$, based on sparse observation Y. Assuming that the elements ${y_i}'s$ of the observations are i.i.d, for continuous observations with Gaussian noise, the joint likelihood distribution of Y can be written as:

$$\mathrm{p}(Y|X) = \prod_{\mathbf{i}} \mathcal{N}(y_{\mathbf{i}}|x_{\mathbf{i}}, \tau_o^{-1}) \tag{2}$$

where τ_o is the precision of the noise, and $\mathbf{i} = i_1, ...i_k, ...i_K$.

The problem is how to reduce the size of tensor rank so as to make a rich feature-based and scalable user representation during model construction. Our low-rank construction is adopted from [3] and [10] using the multiplicative gamma process (MGP). Putting the prior on the super-diagonal elements of the core tensor $\boldsymbol{\lambda}$, the number of diagonal elements will increasingly shrink to zero. When it stabilizes, the number of the remaining elements can be inferred as the appropriate rank for dimensionality reduction.

3.2 Multi-layer Sparse Factorization

To enhance the feature representation, factor matrix $U^{(k)}$ for mode k can be further constructed through an unsupervised deep model in terms of a hierarchy of factor analysis. Assuming that the hierarchical tensor factorization model is performed for L layers, the original data Y is represented in terms of $n_k \times R$ tensor factor matrix $U^{(k)}$ as in equation (1). $U^{(k)}$ can be further divided as $W_1^{(k)} D_1^{(k)} + E_1^{(k)}$. For the deeper layers, the input for each layer is the previous layer's factor loading matrix (as shown in Fig. 2). The discarding of residue between layers acts as noise filtering. Hence, each factor matrix is further represented by a lower rank factor loading component $W^{(k)} \in \mathbb{R}_{n_k \times M}$ and factor score $D^{(k)} \in \mathbb{R}_{M \times R}$, where M indicates the number of factors for this layer. The matrix E captures the idiosyncratic noise. $l = 1, 2, \ldots, L$ specifies how deep the network wishes to go to.

The inference of the factor number for each layer is realized through a Beta-Bernoulli process [11] as $W^{(k)} = B^{(k)} \odot V^{(k)}$. In practice, the number of layers is initialized large, thus the element $\{b_{nm}^{(k)}\}_{1 \le n \le n_k, 1 \le m \le M} \in \{0, 1\}$ of $B^{(k)}$ can indicate whether the m^{th} factor has been used or not. To go to the next layer, we denote $\hat{M}$ as the number of factors that has nonzero indicator $B^{(k)}$ for at least one sample, and use these factor loadings as the entry of the next layer. Model fitting at the deeper layers are similar to the first layer. With the gradual deduction of factor numbers, the computational complexity decreases with layer getting deeper.

3.3 Probabilistic Hierarchical Tensor Model

We now propose a hierarchical generative framework for a three-way tensor data whereby the aforementioned tensor construction and deep model fitting can be performed. The graphic model of DCPF model is shown in Fig. 2. The multiplicative gamma process

prior is constructed as $\tau_r = \prod_{t=1}^{r} \delta_t,\ \delta_t \sim \mathcal{G}amma(a_r, 1)\ a_r > 1$, and is placed on the precision of the Gaussian distribution for λ_r as $\lambda_r \sim \mathcal{N}(0, \tau_r^{-1})$.

Since exact inference is intractable, we implement the posterior computation using Markov Chain Monte Carlo (MCMC). In the proposed model, all conditional distributions are analytic. The choices for prior hyper-parameters are relatively standard in Bayesian analysis, hence no particular tuning is required. Updating equations for the latent parameters are provided in detail as follows.

• Update mode factor matrix $U^{(k)}$

For $1 \leq r \leq R$, $1 \leq k \leq K$, at the (r, k) tuple, all the other entities are regarded as non-variables, so x_i can be rewritten as:

$$x_{\mathbf{i}} = \underbrace{(\lambda_r \prod_{k' \neq k, k'=1}^{K} u_{i_{k'} r}^{(k')})\, u_{i_k r}^{(k)}}_{} + \underbrace{\sum_{r' \neq r} \lambda_{r'} \prod_{k=1}^{K} u_{i_k r'}^{(k)}}_{} \tag{3}$$

Let the first parentheses part equal to $p_{i_k r}^{(k)}$ and the second part be $q_{i_k r}^{(k)}$. With Gaussian noise precision τ_ϵ, the prior of $u^{(k)}$ is $\mathcal{N}\left(\mu^{(k)}, \tau_\epsilon^{-1}\right)$, where $\mu^{(k)}$ equals to $(B^{(k)} \odot V^{(k)})D^{(k)}$. Thus the conjugate posterior can be inferred as:

$$\mathbf{u}_{i_k}^{(k)} \sim \mathcal{N}\left(\hat{\mu}_{i_k}^{(k)}, \hat{\Sigma}_{i_k}\right) \tag{4}$$

with the posterior expectation and covariance as

$$\begin{aligned} \hat{\Sigma}_{i_k} &= (\tau_o \sum_{i_k} \mathbf{p}_{i_k}^{(k)2} + \tau_\epsilon^{-1}) \\ \hat{\mu}_{i_k}^{(k)} &= \hat{\Sigma}_{i_k}^{-1} \left(\tau_\epsilon \mu_{i_k}^{(k)} + \tau_o \sum_i (y_i - \mathbf{q}_{i_k}^{(k)}) \mathbf{p}_{i_k}^{(k)} \right) \end{aligned} \tag{5}$$

• Update binary indicator matrix $B_l^{(k)}$ and factor loading matrix $V_l^{(k)}$

For each entity $b_{i_k m}^{(k)}$ of $B_l^{(k)}$, we have

$$p(b_{i_k m}^{(k)} = 1 | -) = \widetilde{\pi_{i_k m}^{(k)}} \tag{6}$$

where $\frac{\widetilde{\pi_{i_k m}^{(k)}}}{1 - \widetilde{\pi_{i_k m}^{(k)}}} = \frac{\pi_{i_k m}^{(k)}}{1 - \pi_{i_k m}^{(k)}} exp[-\frac{\tau_\epsilon}{2}(v_{i_k m}^{(k)\ 2} \mathbf{d}_m^{(k)} \mathbf{d}_m^{(k)T} - 2 v_{i_k m}^{(k)} U_{-m}^{(k)} \mathbf{d}_m^{(k)T})]$

$U_{-m}^{(k)}$ here equals to $\mathbf{u}^{(k)} - \sum_m (b_{i_k m}^{(k)} \odot V^{(k)}) D^{(k)}$ and $b_{i_k m}^{(k)}$ is the most recent sample [4].

Taking the advantage of conjugate property, the posterior mean and covariance for the factor loading element $v_{i_k m}^{(k)}$ can be derived as:

$$\Sigma_v = 1 \oslash (\tau_v + \tau_\epsilon \mathbf{d}_m^{(k)} \mathbf{d}_m^{(k)T} b_{i_k m}^{(k)}) \tag{7}$$

$$\mu_v = \tau_\epsilon b_{i_k m}^{(k)} \Sigma_v \odot (U_{-m}^{(k)} \mathbf{d}^{(k)} + \mathbf{d}_m^{(k)} \mathbf{d}_m^{(k)T} v^{(k)}) \tag{8}$$

$\odot$ and $\oslash$ are the element-wise product and division operator respectively. For multi-layer implementation, after sampling $\mathbf{v}$, it is used as the input to the next layer. Thus, there is a residue filtering between each layer.

The sampling for factor score matrix $D^{(k)}$ is similar to $V^{(k)}$. Note that since for each layer of the deep network, the basic model is the same, the layer superscripts are omitted for clarity.

4 Experiments

We perform experiments on both synthetic toy data and large-scale real-word dataset to verify the performance of the DCPF model on expressing high-level semantic features of user behavior, and the effect of deep structure exploration for recommendation through multi-layer tensor construction.

4.1 Toy Example

The first example we considered is a toy problem, in which 3-D and 4-D cases are tested to verify the tensor completion performance of DCPF. The 3-D synthetic data is of size $15 \times 14 \times 13$ with 50% non-zero values. The 4-D synthetic data is of size $20 \times 20 \times 20 \times 20$ with 1000 non-zero values. Table 1 compares the results using a baseline method Bayesian CP (BCP), i.e. a fully Bayesian version of the standard probabilistic CP decomposition [12]. The inferred rank using our method is 8 and 10 for the two synthetic datasets. Since BCP has to specify the rank, it was run with ranks ranging from $3 - 10$, and the rank that generates smallest mean squared error (MSE) is chosen for comparison. We compare the reconstruction errors (MSE) in Table 1. For both cases, one layer and 2-layer DCPF provides competitive performances comparing to the state-of-art BCP.

Table 1. Synthetic data MSE comparison

	3-D data (R=8)	4-D data (R=10)
Bayesian CP	0.2431 ± 0.0247 (R=11)	0.0922 ± 0.0207 (R=10)
DCPF	0.2502 ± 0.0055	0.0459 ± 0.0014
2-layer DCPF	0.2490 ± 0.0006	0.0412 ± 0.0011

We also construct a three-way $100 \times 100 \times 100$ tensor, with sparseness control (missing percentage) of $50\% - 90\%$ (Table 2). From varying the percentage of missing values, we can infer that a multi-layer filtering of the factor matrix will prevent the degrading of the reconstruction performance especially when the data has higher sparseness percentage.

The scalability is tested with tensor size $100 \times 100 \times 100$. Specifically, with $100,000$ entries of observations and a fixed core tensor rank at 50, based on MATLAB implementation, the averaged time for 50 iterations is $3,542$ seconds.

Table 2. Reconstruction error comparison of different data sparse percentages (lower the better)

	90%	80%	70%	60%	50%
Bayesian CP	0.4137	0.4123	0.4120	0.4093	0.3993
DCPF	0.4104	0.4100	0.3989	0.3959	0.3957
2-layer DCPF	0.3951	0.3865	0.3811	0.3697	0.3542

4.2 Video Recommendation

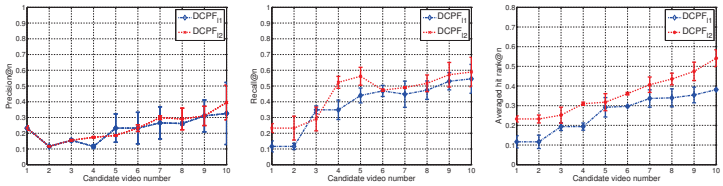
For real-world application, we use records in three consecutive weeks (August 1 - August 21, 2014) at the same time slot from Tencent QQ browser, a one-stop browser used by millions of users. The database stores historical viewing records of active users extracted from the Hadoop distributed file system, with each record referring to one video clicked. Besides the time slot, user id, video id, number of clicks by this particular user and the total number of clicks in history, there are four categories of information provided as video word tag (in Chinese): (1) type (e.g. action, mystery); (2) region (e.g. US, main land China); (3) director; and (4) actor. Based on these information, a three-way tag$\times$ user $\times$ video tensor was constructed. There are $4,071,811$ samples in total with $137,465$ unique users, and $9,393$ unique videos they clicked. We focus on warm-start test for the current application, which requires both active users and candidate videos occurred before. So the training and testing subsets are generated as follows.

We select a subset of users who have over 10 historical records and adopt a 5-fold cross-validation for evaluation. To guarantee that all the users and items have already appeared in the training sets, for records that have been viewed by each user for over five times, we distribute at least one of them into each fold. For records that have appeared less than five times ($246,642$ in total), we always put them into the training fold. The dimensionality of the testing tensor is $1421 \times 4013 \times 231$.

Since the generated factor matrices discover the latent groups of associations among tags, users and videos, we can utilize them to construct score matrices for recommendation [16]. For example, the i^{th} row of user factor matrix $U^{(2)}$ provides an additive combination of cluster components for user i. The higher weight it is, the more relevant user i is related to the topic, and similarly for the j^{th} row of video factor matrix $U^{(3)}$. Thus, groups can be recommended according to the linear add-up of their corresponding factor weights. The score matrix S for user-video pairs can be defined as:

$$S = \sum_{r=1}^{R} \lambda_r u_r^{(2)} u_r^{(3)T} \tag{9}$$

Usually for the browser interface, $5-10$ videos are listed as a guessing of user interest. The values for precision@n, recall@n and averaged hit rate are illustrated in Fig. 3. The increase of pool number will enhance the probability to recommend the favored video. Thus, the three metrics have the trend of gradually growing. Single layer and 2-layer DCPF have similar precision for prediction, but a multi-layer implementation allows an obvious higher hit rate, which indicates that it can pick the correct choice of users at an earlier stage of recommendation.



(a) Precision (b) Recall (c) Averaged hit rate

Fig. 3. Video recommendation performance comparison for single layer and 2-layer DCPF with varying candidates number n ranging from 1-10.

4.3 Performance Evaluation

To visualize how the multi-layer implementation of DCPF actually influences the feature representation, we present four sample factors of tags discovered using both single layer and 2-layer DCPF in this section. We examine $U^{(1)}$ and $U^{(3)}$, which represent the latent factors of tags and videos respectively. The rank is 78, with 64 factors out of 70 are used in the next layer. Four top weighted factors from each factor matrix are selected with eight highest score items each for concise visualization of topics.

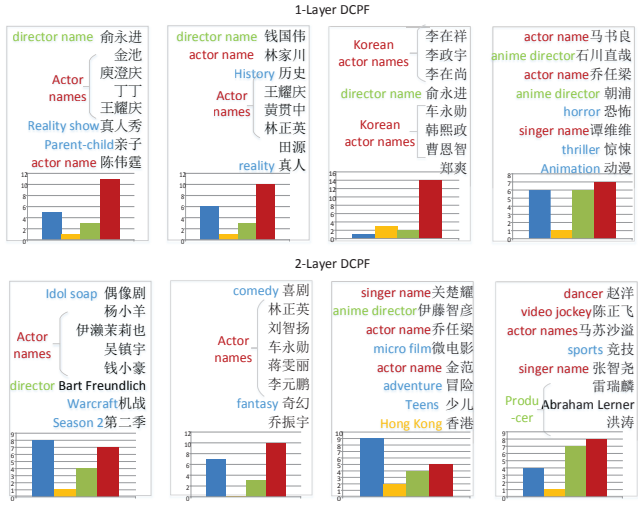


Fig. 4. Four top weighted factors learned from single layer and 2-layer DCPF. Each block shows two aggregated statistics of the factor: the eight most representative tags and a histogram of the distribution of 20 top weighted tags from 4 categories. The Columns in each of the histogram correspond to type, region, director and actor from left to right.

As shown in Fig. 4, semantic topics using single layer DCPF are primarily consisted of tags from the actor category. This phenomena illustrates that on one hand, certain types of users indeed choose their watching list according to the favor of particular actors, on the other hand, this could be also due to the high occurrences of actors and reduplicative annotation. When going deeper, a 2-layer factorization filters out noises and abundances. Thus, factor weights for the other three categories of tags are better explored. This observation also provides a possible explanation for the better hit rate performance of 2-layer DCPF shown in Fig.3. Since the user preferences are naturally and comprehensively mixed, although the group of tags for each factor seems more irregular on the surface, they actually better interpret the semantic level topics of user tastes.

5 CONCLUSION

With the exponential growth in large-scale relational data, effective feature learning techniques are greatly desired. In recommender systems, volatile user and sparse video tags present challenges to traditional collaborative filtering systems. Tensor factorization provides an effective way for joint analysis of user and video features. Through a scalable framework for tensor completion, we are able to recommend personalized items flexibly. Deep learning is also leveraged to explore richer item representation of user preferences. Our model can perform fully conjugate Bayesian inference via Gibbs sampling and can assess recommendation performance quantitatively.

Although the data we use in this paper are only real valued ones, the framework can also be extended to handle counting or binary ones. We aim to incorporate various types of information to enhance user behavior representation in the future. Also currently the novelty is evaluated based on generating new combination of existent items. Creative construction for data from previously unexplored domain based on current knowledge are also appealing for future targets.

Acknowledgment

The authors would like to thank Tencent Mobile Internet group for providing the data and suggestions.

References

1. Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. In: Proceedings of the twenty-first international conference on Machine learning. p. 9. ACM (2004)
2. Bengio, Y.: Learning deep architectures for ai. Foundations and trends® in Machine Learning 2(1), 1–127 (2009)
3. Bhattacharya, A., Dunson, D.B.: Sparse bayesian infinite factor models. Biometrika 98(2), 291–306 (2011)
4. Chen, B., Polatkan, G., Sapiro, G., Blei, D., Dunson, D., Carin, L.: Deep learning with hierarchical convolutional factor analysis. IEEE transactions on pattern analysis and machine intelligence 35(8) (2013)

5. Chu, W., Ghahramani, Z.: Probabilistic models for incomplete multi-dimensional arrays (2009)
6. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.i.: Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons (2009)
7. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM review 51(3), 455–500 (2009)
8. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 609–616. ACM (2009)
9. Pan, S.J., Yang, Q.: A survey on transfer learning. Knowledge and Data Engineering, IEEE Transactions on 22(10), 1345–1359 (2010)
10. Rai, P., Wang, Y., Guo, S., Chen, G., Dunson, D., Carin, L.: Scalable bayesian low-rank decomposition of incomplete multiway tensors. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). pp. 1800–1808 (2014)
11. Thibaux, R., Jordan, M.I.: Hierarchical beta processes and the indian buffet process. In: International conference on artificial intelligence and statistics. pp. 564–571 (2007)
12. Xiong, L., Chen, X., Huang, T.K., Schneider, J.G., Carbonell, J.G.: Temporal collaborative filtering with bayesian probabilistic tensor factorization. SIAM
13. Xu, Z., Yan, F., Qi, Y.: Bayesian nonparametric models for multiway data analysis (2013)
14. Xu, Z., Yan, F., et al.: Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis. arXiv preprint arXiv:1108.6296 (2011)
15. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 2528–2535. IEEE (2010)
16. Zheng, N., Li, Q., Liao, S., Zhang, L.: Flickr group recommendation based on tensor decomposition. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 737–738. ACM (2010)

Combining a Relaxed EM Algorithm with Occam's Razor for Bayesian Variable Selection in High-Dimensional Regression

Pierre-Alexandre Mattei[1], Pierre Latouche[2], and Charles Bouveyron[1]
Julien Chiquet[3]

[1] Laboratoire MAP5, UMR CNRS 8145, Université Paris Descartes
[2] Laboratoire SAMM, EA 4543, Université Paris 1 Panthéon-Sorbonne
[3] Laboratoire LaMME, UMR CNRS 8071/UEVE, USC INRA, Évry, France

Abstract. We address the problem of Bayesian variable selection for high-dimensional linear regression. We consider a generative model that uses a spike-and-slab-like prior distribution obtained by multiplying a deterministic binary vector, which traduces the sparsity of the problem, with a random Gaussian parameter vector. The originality of the work is to consider inference through relaxing the model and using a type-II log-likelihood maximization based on an EM algorithm. Model selection is performed afterwards relying on Occam's razor and on a path of models found by the EM algorithm. Numerical comparisons between our method, called spinyReg, and state-of-the-art high-dimensional variable selection algorithms (such as lasso, adaptive lasso, stability selection or spike-and-slab procedures) are reported. Competitive variable selection results and predictive performances are achieved on both simulated and real benchmark data sets. An R package implementing the spinyReg method is currently under development and is available at `https://r-forge.r-project.org/projects/spinyreg`.

1 Introduction

Over the past decades, parsimony has emerged as a very natural way to deal with high-dimensional data spaces [3]. In the context of linear regression, finding a parsimonious parameter vector can both prevent overfitting, make an ill-posed problem (such as a "large p, small n" situation) tractable, and allow to interpret easily the data by finding which predictors are relevant. The problem of finding such predictors is referred to as *sparse regression* or *variable selection* and has mainly been considered either by likelihood penalization of the data, or by using Bayesian models.

Penalized likelihood. The most natural sparsity-inducing penalty, the ℓ_0-pseudonorm, unfortunately leads to an NP-hard optimization problem [16] that is intractable as soon as the number of predictors exceeds a few dozens. To overcome this restriction, convex relaxation of the ℓ_0-pseudonorm, that is, ℓ_1-regularization,

have become a basic tool in modern statistics. The most spread formulation of the ℓ_1-penalized linear regression was introduced by [21] as the "least absolute shrinkage and selection operator" (lasso). Several algorithms allow fast computations of the lasso, even when the number of predictors largely exceeds the number of observations. However, the crude lasso is not model-consistent unless some cumbersome conditions on the design matrix [25]. Moreover, it can be sensitive to highly correlated predictors [27] and its distributional properties can be surprisingly complex [18].

Bayesian modelling. Bayesian models have also been widely studied in a variable selection context [17]. The most efficient techniques essentially rest on spike-and-slab procedures. Spike-and-slab models, first introduced by [15], use mixtures of two distributions as priors for the regression coefficients: a thin one, corresponding to irrelevant predictors (the *spike*, typically a Dirac law or a Gaussian distribution with small variance) and a thick one, corresponding to the relevant variables (the *slab*, typically a uniform or Gaussian distribution of large variance). Markov chain Monte Carlo (MCMC) methods have been usually chosen to select models with the highest posterior distributions. MCMC techniques have an important computational cost and may suffer from poor mixing properties in the case of spike-and-slab-like priors [17]. A few deterministic methods have also recently been proposed to tackle this issue [19, 24].

Our approach. As an alternative, our approach uses spike-and-slab-like priors induced by a binary vector which segregates the relevant from the irrelevant predictors. Such vectors, introduced by [6] have been widely used in the Bayesian literature, but have always been considered as random parameters. In most Bayesian contexts, such a binary vector would be classically endowed with a product of Bernoulli prior distributions. In our work, the originality is to consider a deterministic binary vector, and to relax it in order to rely on an EM algorithm. This relaxed procedure allows us to find a family of p models, ordered by sparsity. Model selection is performed afterwards by maximizing the marginal likelihood over this family of models.

Notation. For two matrices A and B of $\mathcal{M}_{n,p}$, we define their Hadamard product as $A \odot B = (a_{ij}b_{ij})_{i\leq n, j\leq p}$ where a_{ij} and b_{ij} respectively denote the (i,j)-th coordinate of A and B. The identity matrix of dimension n is denoted by $\mathbf{I}_n$. Given a binary vector $\mathbf{z} \in \{0,1\}^p$, we denote $\bar{\mathbf{z}}$ the binary vector of $\{0,1\}^p$ whose support is exactly the complement of $\mathrm{Supp}(\mathbf{z})$. Given a binary vector $\mathbf{z} \in \{0,1\}^p$ and a matrix $\mathbf{A} \in \mathcal{M}_{n,p}$, we denote $\mathbf{A}_\mathbf{z}$ the extracted matrix of $\mathbf{A}$ where only the columns corresponding to the nonzero indexes of $\mathbf{z}$ have been kept.

2 A sparse generative model

Let us consider the following regression model

$$\begin{cases} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ \boldsymbol{\beta} &= \mathbf{z} \odot \mathbf{w}, \end{cases} \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^n$ is the set of n observed responses, $\mathbf{X} \in \mathcal{M}_{n,p}(\mathbb{R})$ is the design matrix with p input variables. The vector $\boldsymbol{\varepsilon}$ is a noise term with $p(\boldsymbol{\varepsilon}|\gamma) = \mathcal{N}(\boldsymbol{\varepsilon}; 0, \mathbf{I}_n/\gamma)$. A prior distribution $p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}; 0, \mathbf{I}_p/\alpha)$ with an isotropic covariance matrix is further assumed. Moreover, we denote by $\mathbf{z} \in \{0,1\}^p$ a binary deterministic parameter vector, whose nonzero entries correspond to the active variables of the regression model. It is worth noticing that such modeling induces a spike-and-slab-like prior distribution for $\boldsymbol{\beta}$:

$$p(\boldsymbol{\beta}|\mathbf{z}, \alpha) = \prod_{j=1}^{p} p(\beta_j|z_j, \alpha) = \prod_{j=1}^{p} \delta_0(\beta_j)^{1-z_j} \mathcal{N}(\beta_j; 0, 1/\alpha)^{z_j}. \tag{2}$$

Contrary to standard spike-and-slab models [15] which assume a Bernoulli prior distribution over $\mathbf{z}$, we see $\mathbf{z}$ here as a deterministic parameter to be inferred from the data. As we shall see in Section 3, this allows us to work with a marginal log-likelihood which involves an Occam's razor term, allowing model selection afterwards. In the same spirit, we do not put any prior distribution on γ nor α. From now on, to simplify notations, the dependency on $\mathbf{X}$ in conditional distributions will be omitted.

Proposition 1 *The posterior distribution of* $\mathbf{w}$ *given the data is given by*

$$p(\mathbf{w}|\mathbf{Y}, \mathbf{Z}, \alpha, \gamma) = \mathcal{N}(\mathbf{w}; \mathbf{m}, \mathbf{S}), \tag{3}$$

where $\mathbf{S} = (\gamma \mathbf{Z}\mathbf{X}^T\mathbf{X}\mathbf{Z} + \alpha \mathbf{I}_p)^{-1}$ *and* $\mathbf{m} = \gamma \mathbf{S}\mathbf{Z}\mathbf{X}^T\mathbf{Y}$.

The vector $\mathbf{m}$ is the maximum a posteriori (MAP) estimate of $\boldsymbol{\beta}$. Next proposition assures that it recovers the support of the parameter vector. Moreover, its nonzero coefficients correspond to ridge estimates with regularization parameter α/γ of the model where only the q predictors corresponding to the support of $\mathbf{z}$ have been kept.

Proposition 2 *We have* $\mathrm{Supp}(\mathbf{m}) = \mathrm{Supp}(\mathbf{z})$ *almost surely and*

$$\mathbf{m}_\mathbf{z} = \left(\mathbf{X}_\mathbf{z}^T \mathbf{X}_\mathbf{z} + \frac{\alpha}{\gamma} \mathbf{I}_p \right)^{-1} \mathbf{X}_\mathbf{z}^T \mathbf{Y}. \tag{4}$$

3 Inference

This section now focuses on inferring the model proposed above. To this end, $\mathbf{w}$ is seen as a latent variable while $\mathbf{Z} = \mathrm{diag}(\mathbf{z})$, α, γ are parameters to be estimated from the data $(\mathbf{X}, \mathbf{Y})$ using an empirical Bayes framework. The estimators of $\mathbf{z}$,

α and γ will be the ones that maximize the *evidence* (or *type-II likelihood*) of the data:

$$p(\mathbf{Y}|\mathbf{X}, \mathbf{z}, \alpha, \gamma) = \int_{\mathbb{R}^p} p(\mathbf{Y}|\mathbf{X}, \mathbf{w}, \mathbf{z}, \alpha, \gamma) p(\mathbf{w}|\alpha) d\mathbf{w}. \tag{5}$$

Seing $\mathbf{w}$ as a latent variable, a natural optimization procedure is the expectation-maximization (EM) algorithm introduced by [4]. However, the maximization of (5) would be problematic for two reasons. First, because the optimization problem in $\mathbf{z}$ is combinatorial and 2^p values of $\mathbf{z}$ are possible. Then, because in this case, the parameter space is partly discrete and all theoretical convergence properties of the EM algorithm require a continuous parameter space [23, 13]. To overcome these issues, we propose to use a simple relaxation by replacing the model parameter by a vector $\mathbf{z}^{\text{relaxed}}$ in $[0, 1]^p$. This relaxation allows us to efficiently maximize the new, relaxed version of (5) using an EM approach. Interestingly, this relaxed model is somehow related to the automatic relevance determination (ARD) [11, 22]. However, our method avoids several drawbacks of this technique, for more details, see the extended working paper [12]. From now on, and until the end of this section, we will only consider the relaxed model with $\mathbf{z}^{\text{relaxed}} \in [0, 1]^p$. In order to simplify notations, we denote $\mathbf{Z} = \text{diag}(\mathbf{z}^{\text{relaxed}})$.

E-step. At the E-step of the relaxed EM algorithm, one has to compute the expectation of the complete data log-likelihood $\mathbb{E}_{\mathbf{w}}(\log p(\mathbf{Y}, \mathbf{w}, |\mathbf{Z}, \alpha, \gamma))$ with respect to the posterior distribution $p(\mathbf{w}|\mathbf{Y}, \mathbf{Z}, \alpha, \gamma)$. Consequently, the parameters $\mathbf{S}$ and $\mathbf{m}$ of the Gaussian posterior (3) have to be computed at each step.

M-step. At the M-step, the expectation of the complete data log-likelihood $\mathbb{E}_{\mathbf{w}}(\log p(\mathbf{Y}, \mathbf{w}|\mathbf{Z}, \alpha, \gamma))$ with respect to $p(\mathbf{w}|\mathbf{Y}, \mathbf{Z}, \alpha, \gamma)$, is maximized over $\mathbf{Z}, \alpha, \gamma$. This leads to the following M-step updates.

Proposition 3 *The values of* γ, α, $\mathbf{z}^{\text{relaxed}}$ *maximizing* $\mathbb{E}_{\mathbf{w}}(\log p(\mathbf{Y}, \mathbf{w}|\mathbf{Z}, \alpha, \gamma))$ *are*

$$\hat{\gamma}^{-1} = \frac{1}{n} \left\{ \mathbf{Y}^T\mathbf{Y} + \mathbf{z}^{\text{relaxed}^T}(\mathbf{X}^T\mathbf{X} \odot \boldsymbol{\Sigma})\mathbf{z}^{\text{relaxed}} - 2\mathbf{z}^{\text{relaxed}^T}(\mathbf{m} \odot (\mathbf{X}^T\mathbf{Y})) \right\} \tag{6}$$

$$\hat{\alpha} = \frac{p}{\text{Tr}(\boldsymbol{\Sigma})} \tag{7}$$

$$\hat{\mathbf{z}}^{\text{relaxed}} = \underset{\mathbf{u} \in [0,1]^p}{\text{argmax}} \left\{ -\frac{1}{2}\mathbf{u}^T(\mathbf{X}^T\mathbf{X} \odot \boldsymbol{\Sigma})\mathbf{u} + \mathbf{u}^T(\mathbf{m} \odot (\mathbf{X}^T\mathbf{Y})) \right\} \tag{8}$$

Notice that it can be shown (see the extended verson [12]) that the $\mathbf{z}^{\text{relaxed}}$ update (8) is a quadratic program (QP) which is strictly concave if, and only if $\mathbf{X}$ has no null column. Since it is always the case in practice, fast convex optimization techniques can be used to solve (8) efficiently.

Computational cost. At each iteration, the most expensive step is the inversion of the $p \times p$ matrix $\mathbf{S}$ during the E-step. It would imply a $O(p^3)$ complexity, not allowing us to deal with high-dimentional data. However, using the Woodbury identity, one can write when $p > n$,

$$\mathbf{S} = \frac{1}{\alpha}\mathbf{I}_p + \frac{1}{\alpha^2}\left(\mathbf{Z}\mathbf{X}^T\right)\left(\frac{1}{\gamma}\mathbf{I}_n + \frac{1}{\alpha}\mathbf{X}\mathbf{Z}^2\mathbf{X}^T\right)^{-1}\left(\mathbf{X}\mathbf{Z}\right).$$

Thus, the final computational cost has therefore a $O(p^2 \min(n,p))$ complexity, which is more suitable for high-dimensional problems. For more details and a comparison with the complexity of state-of-the art Bayesian and frequentist methods see the extended working paper [12].

4 Model selection

In practice, the vector $\mathbf{z}^{\text{relaxed}}$ has to be binarized in order to select the relevant input variables. A common choice would consist in relying on a threshold τ such that z_j is set to 1 if $z_j \geq \tau$, and to 0 otherwise. However, numerical experiments showed that such a procedure would lead to poor estimates of $\mathbf{z}$. In order to perform an efficient variable selection, we will use the outputs of the relaxed EM algorithm to create a path of models and, relying on Occam's razor, we will afterward maximize the type-II likelihood over this path to finally select the relevant variables.

4.1 Occam's Razor

One of the key advantages of the approach proposed is that it maximizes a marginal log-likelihood, which automatically penalizes the model complexity by adding a term to the sum of squared errors.

Proposition 4 *Up to unnecessary additive constants, the negative type-II log-likelihood can be written as*

$$\begin{aligned} -\log p(\mathbf{Y}|\mathbf{z},\alpha,\gamma) &= -\log p(\mathbf{Y}|\mathbf{m},\mathbf{z},\gamma) + \text{pen}(\mathbf{z},\alpha,\gamma) \\ &= \frac{\gamma}{2}||\mathbf{Y} - \mathbf{X}_\mathbf{z}\mathbf{m}_\mathbf{z}||_2^2 + \text{pen}(\mathbf{z},\alpha,\gamma) \end{aligned} \tag{9}$$

where

$$\text{pen}(\mathbf{z},\alpha,\gamma) = -\log p(\mathbf{m}|\alpha) - \frac{1}{2}\log\det\mathbf{S} \tag{10}$$

$$= \frac{\alpha}{2}\left\|\mathbf{m}\right\|_2^2 - \frac{\log\alpha}{2}\left\|\mathbf{m}\right\|_0 - \frac{1}{2}\log\det(\gamma\mathbf{X}_\mathbf{z}^T\mathbf{X}_\mathbf{z} + \alpha\mathbf{I}_q) \text{ a.s} \tag{11}$$

is the Occam factor.

The sparse generative model therefore automatically adds a ℓ_0-ℓ_2 penalty to the likelihood of the model at the MAP value of $\mathbf{w}$. This is somehow similar to the "elastic net" penalty of [27], combined with a penalty linked to the volume of the gaussian posterior $\mathcal{N}(\mathbf{w}; \mathbf{m}, \mathbf{S})$. Notice that, when α is small, the Occam factor will be extremely sparsity-inducing but the coefficients will have a large variance. When α is close to 1, this penalty will lead to moderately sparse but notably shrinked solution. Moreover, if we write $\lambda = (\alpha - \log \alpha)/2$ and $\kappa = \alpha/(\alpha - \log \alpha)$, we obtain almost surely the expression

$$\text{pen}(\mathbf{z}, \alpha, \gamma) = \lambda \left((1-\kappa) \left\| \mathbf{m} \right\|_0 + \kappa \left\| \mathbf{m} \right\|_2^2 \right) - \frac{1}{2} \log \det(\gamma \mathbf{X}_\mathbf{z}^T \mathbf{X}_\mathbf{z} + \alpha \mathbf{I}_q),$$

involving a convex combination of the ℓ_0 and ℓ_2 penalties in an elastic net fashion. The elastic net can therefore be seen as some kind of strictly convex approximation of Occam's automatic penalty. Interestingly, the term $\text{pen}(\mathbf{z}, \alpha, \gamma)$ exactly corresponds to Occam's razor described by [10] and detailed by [1, chap. 4]. Such a term has been widely used for model selection purposes and is linked to the Bayesian information criterion and to Bayesian hypothesis testing [9] .

4.2 Path of Models

We rely on $\hat{\mathbf{z}}^{\text{relaxed}}$ to find a path of models which are likely to have a high evidence. We build a path by assuming that the larger the coefficients of $\hat{\mathbf{z}}^{\text{relaxed}}$ are, the more likely they are to correspond to relevant variables.

We define the set of vectors $(\hat{\mathbf{z}}^{(k)})_{k \leq p}$ as the binary vectors such that, for each k, the k top coefficients of $\hat{\mathbf{z}}^{\text{relaxed}}$ are set to 1 and the others to 0. For example, $\hat{\mathbf{z}}^{(1)}$ contains only zeros and a single 1 at the position of the highest coefficient of $\hat{\mathbf{z}}^{\text{relaxed}}$. The set of vectors $(\hat{\mathbf{z}}^{(k)})_{k \leq p}$ defines a path of models to look at for model selection. Note that this path allows us to deal with a family of p models (ordered by sparsity) instead of 2^p, allowing our approach to deal with a large number of input variables. Thus, the evidence is evaluated for all $\hat{\mathbf{z}}^{(k)}$ and the number $\hat{q}$ of relevant variables is chosen such that the evidence is maximized:

$$\hat{q} = \text{argmax}_{1 \leq k \leq p} p(\mathbf{Y} | \hat{\mathbf{z}}^{(k)}, \hat{\alpha}, \hat{\gamma}) \quad \text{and} \quad \hat{\mathbf{z}} = \hat{\mathbf{z}}^{(\hat{q})}. \tag{12}$$

We called our algorithm, which successively runs the relaxed and performs model selection over the path of models using (12), spinyReg. Several details about its implementation can be found in the working paper [12]

5 Numerical comparisons

Simulation setup. In order to consider a wide range of scenarios, we use three different simulation scenarios: "uniform", "Toeplitz" and "blockwise". The simulation of the parameter $\mathbf{w}$ and of the noise $\boldsymbol{\varepsilon}$ is common for the three schemes: $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I}_p/\alpha)$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}_n/\gamma)$. The design matrix $\mathbf{X}$ is simulated according to a Gaussian distribution with zero mean and a covariance matrix R

depending on the chosen scheme. The correlation structure of $R = (r_{ij})_{i,j=1,...,p}$ is as follows:

- "uniform": $r_{ii} = 1$ for all $i = 1, ...p$ and $r_{ij} = \rho$ for $i, j = 1, \ldots, p$ and $i \neq j$,
- "Toeplitz": $r_{ii} = 1$ for all $i = 1, ...p$ and $r_{ij} = \rho^{|i-j|}$ for $i, j = 1, \ldots, p$ and $i \neq j$,
- "blockwise": $R = \text{diag}(R_1, ..., R_4)$ is a 4-blocks diagonal matrix where R_ℓ is such that $r_{\ell ii} = 1$ and $r_{\ell ij} = \rho$ for $i, j = 1, \ldots, p/4$ and $i \neq j$.

Then, $\mathbf{Z}$ is simulated by randomly picking q active variables among p. The predictive vector Y is finally computed according to Equation (1).

5.1 Benchmark study on simulated data

We now compare the performance of spinyReg with three of the most recent and popular variable selection methods based on ℓ_1 regularization: the lasso [21], the adaptive lasso [26] and the stability selection [14]. We also added two recent spike-and-slab approaches: the multi-slab framework of CLERE [24] and the EP procedure of [7]. To this end, we simulated 100 data sets for each of the three simulations schemes (uniform, Toeplitz and blockwise), for three data set sizes ($n = p/2$, $n = p$, $n = 2p$) and two values for the correlation parameter ($\rho = 0.25$ and $\rho = 0.75$). The other simulation parameters were $p = 100$, $q = 40$, $\alpha = 1$ and $\gamma = 1$. The measures used to evaluate the method performances are the prediction mean square error on test data (MSE, hereafter), the F-score (the harmonic mean of precision and recall, which provides a good summary of variable selection performances) and the estimated value of q (number of relevant predictors). Details about the implementation of all the algorithms we compared are provided in the extended working paper[12]. We present on Fig. 1 the results for one simulation setup: the blockwise case with $\rho = 0.75$. All the other results are in the extended working paper[12]. Note that similar conclusions can be drawn on these other scenarios.

We can see that spinyReg and SSEP outperform other methods and have close variable selection performances. SpinyReg appears to be at his best in the "$n = p/2$" case on these runs. Most of the methods perform well in MSE except stability selection and CLERE when $n \leq p$. In particular, spinyReg has the best prediction performance for $n = p/2$ with the highly correlated blockwise case. The lasso has a clear tendency to overestimate the number of active variables, particularly when n becomes large. Conversely, stability selection has the opposite behavior and underestimates q. Its very conservative behavior has the advantage that it avoids false-positives. It turns out that spinyReg provides consistently a good estimate of the actual value of q.

5.2 Study on classical regression data sets

We now consider four real-world data sets: the classical `prostate` data set used for example by [21], the `eyedata` data set of [20], the `OzoneI` data set included

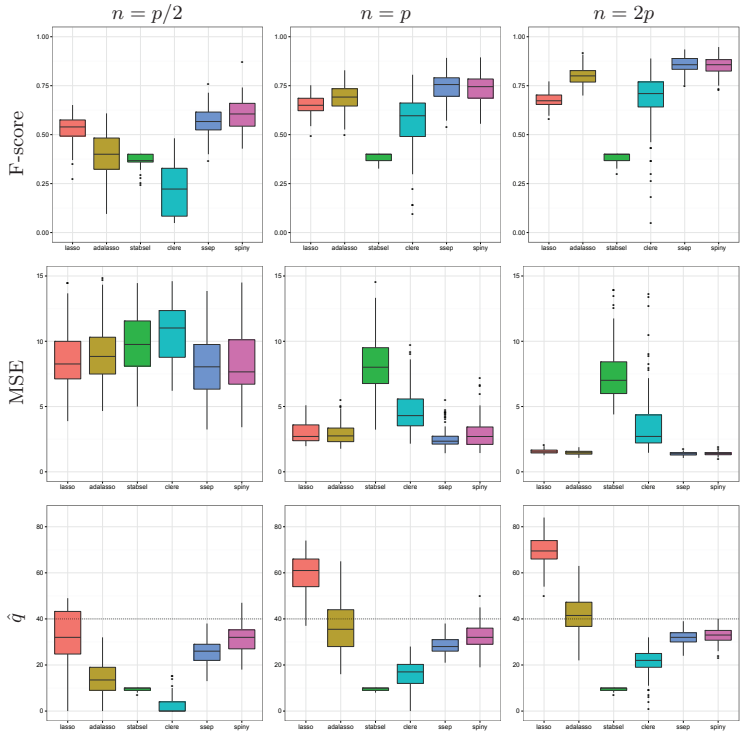


Fig. 1. Scenario "blockwise" with $\rho = 0.75$.

in the `spikeslab` package [8] and which uses the ozone data set of [2] with some additional interactions and the `DiabetesI` data set which is also available in the `spikeslab` package and uses the diabetes data set of [5] with some additional interactions. Applying the same methods as before, we trained our data randomly using 80% of the observations and computed the test error on the remaining data. Repeating this procedure 100 times, we computed the mean and the standard deviation of the test error and of the number of variables selected. Results are reported in Table 1. We did not compute the test error for methods which did not succeed in selecting variables. We can see that spinyReg obtains competitive predictive results on all data sets. Moreover, we can note that it is less conservative than most other algorithms. On the challenging `eyedata` data set for example, while the two other Bayesian methods fail to select at least one variable, spinyReg selects three quarters of the predictors and has the lowest MSE. The three ℓ_1 based methods select only a few variables and have higher

	Prostate ($n = 77$, $p = 8$)		Eyedata ($n = 96$, $p = 200$)	
	MSE×100	Selected variables	MSE×100	Selected variables
Lasso	63.6 ± 21.8	3.33 ± 0.877	1.26 ± 0.964	16.7 ± 5.56
Adalasso	58.4 ± 15.9	4.42 ± 1.57	1.50 ± 1.248	2.4 ± 0.700
Stability Selection	61.6 ± 14.4	1.94 ± 0.239	1.58 ± 0.850	1.7 ± 0.823
Clere	59.8 ± 19.7	2.87 ± 0.825	-	-
SSEP	56.6 ± 15.0	2.76 ± 0.474	-	-
SpinyReg	58.3 ± 15.4	3.34 ± 0.607	1.25 ± 0.920	143 ± 9
	OzoneI ($n = 162$, $p = 134$)		DiabetesI ($n = 353$, $p = 64$)	
	MSE	Selected variables	MSE/1000	Selected variables
Lasso	18.9 ± 4.96	10.3 ± 2.27	3.22 ± 0.407	7.43 ± 2.41
Adalasso	16.84 ± 4.48	8.32 ± 3.16	3.02 ± 0.395	9.31 ± 2.25
Stability Selection	17.9 ± 5.25	9.68 ± 1.10	2.97 ± 0.387	7.77 ± 0.423
Clere	19.6 ± 5.48	5.43 ± 2.55	3.15 ± 0.384	2.33 ± 0.587
SSEP	29.6 ± 10.2	74.8 ± 5.45	3.70 ± 0.647	62.0 ± 1.36
SpinyReg	18.9 ± 5.46	10.79 ± 2.69	3.13 ± 0.376	8.5 ± 1.45

Table 1. Results on real-world data sets

MSE. Let us finally highlight that the medium prediction rank of spinyReg is the second best, behind the adaptive lasso.

6 Conclusion

We considered the problem of Bayesian variable selection for high-dimensional linear regression through a sparse generative model. The sparsity is induced by a deterministic binary vector which multiplies with the Gaussian regressor vector. The originality of the work was to consider its inference through relaxing the model and using a type-II log-likelihood maximization based on an EM algorithm. Model selection can be performed relying on Occam's razor and on a path of models found by the EM algorithm. Numerical experiments on simulated data have shown that spinyReg performs well compared to the most recent competitors both in terms of prediction and of selection, especially in moderately sparse cases and with highly correlated predictors. An extended working paper [12] also contains an application to a new high-dimensional regression data set ($n = 316$, $p = 1158$) involving the prediction of the number of visitors of the Orsay museum in Paris using bike-sharing system data.

References

1. C.M. Bishop. *Pattern recognition and machine learning*. Springer-Verlag, 2006.
2. L. Breiman and J. H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 80(391):580–598, 1985.
3. E. Candès. Mathematics of sparsity (and a few other things). In *Proceedings of the International Congress of Mathematicians, Seoul, South Korea*, 2014.
4. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood for incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B39:1–38, 1977.
5. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

6. E. I. George and R. E. McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.
7. D. Hernández-Lobato, J. M. Hernández-Lobato, and P. Dupont. Generalized spike-and-slab priors for bayesian group feature selection using expectation propagation. *The Journal of Machine Learning Research*, 14(1):1891–1945, 2013.
8. H. Ishwaran, U. B Kogalur, and J. S. Rao. spikeslab: Prediction and variable selection using spike and slab regression. *R Journal*, 2(2), 2010.
9. Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.
10. D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.
11. D. J. C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural computation*, 11(5):1035–1068, 1999.
12. P.-A. Mattei, P. Latouche, C. Bouveyron, and J. Chiquet. Combining a relaxed EM algorithm with occam's razor for bayesian variable selection in high-dimensional regressionn. *HAL preprint*, submitted.
13. G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions. Second Edition.* John Wiley & Sons, New York, 2008.
14. N. Meinshausen and P. Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B*, 27, 2010.
15. T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression (with discussion). *Journal of the American Statistical Association*, 83:1023–1036, 1988.
16. B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.
17. R. B. O'Hara and M. J. Sillanpää. A review of bayesian variable selection methods: what, how and which. *Bayesian analysis*, 4(1):85–117, 2009.
18. B. M. Pötscher and H. Leeb. On the distribution of penalized maximum likelihood estimators: The lasso, scad, and thresholding. *Journal of Multivariate Analysis*, 100(9):2065–2082, 2009.
19. V. Ročková and E. I. George. Emvs: The em approach to bayesian variable selection. *Journal of the American Statistical Association*, just-accepted, 2013.
20. T. E. Scheetz, K.-Y. A. Kim, R. E. Swiderski, A. R. Philp, T. A Braun, K. L. Knudtson, A. M. Dorrance, G. F. DiBona, J. Huang, and T. L. Casavant. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434, 2006.
21. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. (Statistical Methodology)*, 58(1):267–288, 1996.
22. M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.
23. C. F. J. Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.
24. L. Yengo, J. Jacques, and C. Biernacki. Variable clustering in high dimensional linear regression models. *Journal de la Société Française de Statistique*, 155(2):38–56, 2014.
25. P. Zhao and B. Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.
26. H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.
27. H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B. (Statistical Methodology)*, 67(2):301–320, 2005.

Polylingual Multimodal Learning

Aditya Mogadala

Institute AIFB, Karlsruhe Institute of Technology, Germany
{aditya.mogadala}@kit.edu

Abstract. The growth of multimedia content on the web raise diverse challenges. Over the last decades various approaches are designed to support search, recommendations, analytics and advertising majorly based on the textual content. Now, due to the overwhelming availability of media (images and videos) require advancements in the existing technologies to leverage multimedia information. Recent progress made in machine learning to foster continuous representations of the text and effectual object detection in videos and images provide new opportunities. In this aspect, my research work aims to leverage data generated from multimedia to support various applications by finding cross-modal semantic similarity. In particular, it aims to compare semantically similar content generated across different channels by jointly modeling two different modalities. Modeling one or more modalities together can be helpful to generate missing modalities and retrieve cross-modal content. My research also extends textual information in multiple languages to support the growth of polylingual content on the web.

1 Introduction & Motivation

The web contains different modalities. A modality represents information from multimedia items like image, video and text. Most of the time, one or more modalities are represented together to provide a multi-view experience.

Solving challenges posed by multimedia content provide different applications. Most of the earlier research considered multimedia items separately and designed approaches for the tasks like image and video retrieval [1, 2], image annotation [3], image segmentation [4], face detection [5], person identification and tracking [6] etc. In the recent years, multimedia and computer vision communities published considerable research in bridging the gap between modalities to facilitate cross-modal applications [7]. Their research aims to address the problems of automatic image tagging with class labels [3], usage of image queries for text retrieval [8], automatic generation of image and video descriptions [9–11]. Some of these approaches leverage more than one modality by jointly modeling them together. I divide these multimodal learning approaches into three different categories. The first set of approaches generate annotations or tags for images or videos. The second set of approaches provide descriptions (captions) to images and videos with larger phrases or sentences. While, the third set of approaches identifies images and text belonging to same semantic category with cross-modal

retrieval [12, 13]. But, most of the work pertaining to text along with images or videos is limited to English.

Natural language processing (NLP) and information retrieval (IR) communities have been working on different multilingual [14] and cross-lingual applications [15] over the past decades. While they only concentrate on text and diminish the importance of other modalities present in multimodal documents.

Given these limitations of earlier research conducted in natural language processing and computer vision communities supporting different applications. My research work focus on leveraging multilingual and cross-lingual approaches by using more than one modality. Also, it aims to extend multimodal learning to multiple languages. In particular, it aims to find similarities between text present in multiple languages and images or videos by jointly modeling them. Some of the challenges and issues in this research are itemized below.

- How to jointly model different multimedia items like images and videos along with text.
- How to handle variations of text present in different forms like keywords, phrases, noisy data (e.g. tweets, comments) and paragraphs.
- How to design language independent multimodal learning approaches.
- How to extract features which are scalable to multiple data-sets and are not domain dependent.

The remainder of this proposal is organized into the following sections. Background and related work are mentioned in section 2. The section 3 presents limitations of state of the art and provide novelty of my research. Approach along with datasets and evaluation metrics are mentioned in section 4. The section 5 details the work done and the work in progress. Conclusion is discussed in section 6.

2 Background & Related Work

My research work identifies its background from other learning approaches like multi-view and multi-task learning, structured prediction etc., below I list four closely related categories based on tasks.

2.1 Cross-lingual Semantic Similarity (Text only)

Matching semantically similar documents or words belonging to two different languages had been important task to support applications like cross-language information retrieval, machine translation and cross-language classification. In general, cross-language similar documents are retrieved using query translation [16], CL-LSI [17] or CL-KCCA [18]. Other approaches found relatedness with lexicons and semantic knowledge bases [19]. Fuzzy and rough set approaches [20] are also proposed to find cross-language semantic similarity. But lately, more interest is developed to learn latent spaces of two different languages in-order to predict word translations. Latent topics considered as concepts was used for

semantic word similarity tasks in monolingual [21] and multilingual [22, 23] settings. Cross-language latent topics concentrate on extracting concepts based on global co-occurrence of words in a training corpus with and without considering contextual information.

2.2 Cross-modal Semantic Category Similarity

Bridging different modalities is sometimes seen as a task of matching modalities to same semantic class or categories. There are several approaches proposed using joint dimensionality reduction approaches [12, 13] or formulating an optimization problem [24] where correlation between modalities is found by separating the classes in their respective feature spaces. Other approaches aim in learning heterogeneous features implicitly without any external representation. Joint representation of multiple media used by Zhai et al., [25] focus on learning which incorporates sparse and graph regularization.

2.3 Cross-modal Description with Text

Lately, considerable interest has been shown to automatically generate descriptions or captions for images and videos. These descriptions can be either belong to annotations or variable length phrases. Srivastava et al. [26] had learned a joint model of images and their textual annotations with deep boltzmann machines to generate one from other. Vincente et al., [27] automatically created a dataset containing 1 million images with associated visually relevant descriptions from Flickr[1] by performing queries and filtering. Other approaches [9, 10] generated image descriptions with a constraint on text neural language model and images or used fixed templates. Some approaches extended the idea from still images to videos with deep recurrent neural networks [11] and unified frameworks [28]. Other approaches [29] mapped complex textual queries to retrieve videos by understanding visual concepts and semantic graph generated from sentential descriptions. Anna et al. [30] pursued a different use-case by creating a descriptive video service to help visually impaired users to follow a movie.

2.4 Cross-modal Description with Knowledge Bases

Multimedia search understands and organize images and videos in a better manner. Combining visual and semantic resources to analyze the image annotations can provide some world knowledge to visual information. Imagenet [31] is a hierarchal image database structured on lexical database Wordnet. Other approaches combined image tags from Flickr with a commonsense reasoning engine ConceptNet [32] to understand the general tagging behavior. Qi et al., [33] propagates semantic knowledge from text corpus to provide annotations to web images. Visual knowledge bases like NEIL [34] help to continuously build common sense relationships and labels instances of the given visual categories from Internet.

[1] https://www.flickr.com/

Language and context information leveraged from structured knowledge bases was used by Alexander et al., [35] to suggest basic-level concept names depending on the context in which visual object occurs.

3 Limitations with State of the art

The main motivation of my research is to find semantic similarity between content generated across modalities. Though some of the approaches mentioned in the section 2 achieve this in various ways, there are still some limitations which need to be addressed. In this aspect, my research extends or improves multimodal learning for existing and newly created tasks. Below, I divide these tasks originating from two different perspectives.

3.1 Vision for Language

Most of the research conducted earlier is used to compare content across languages for various tasks like cross-language retrieval, cross-language semantic similarity or cross-language classification and mostly focused on only textual information. But due to growth of multimodal content, different language documents are frequently accompanied by images or videos. In my research, I aim to bridge languages with visual clues present in these multimodal documents. This approach creates less dependency on language specific tools and can be scalable to languages which lack resources or tools. Siberalla et al. [36] made a similar attempt to identify semantically similar words with the help of visual information. Though it was only limited to English vocabulary.

3.2 Language for Vision

As mentioned in section 2, there are many ways to link text with images or videos. Text can be generated as annotations, descriptions or labels of semantic categories. Approaches that annotate objects in an image or videos are either limited by domain or leverage information from visual knowledge bases. Most of the keywords which are used to annotate objects are present in English and are depending on word translations to extend them to other languages. Similarly, approaches that are developed for automatic image and video captioning with variable length descriptions are also limited to English. Possible explanation for this limitation is due to approaches that use predefined templates or depend on generation of grammar. Similar issue is been observed with approaches which considered cross-modal retrieval based on same semantic category labels of images and text. Most of the knowledge bases (KB) like DBpedia etc are cross-lingual, though approaches which leverage KB still work with annotations in English.

Observing the possibilities and to support the multilingual web, my research aim to extend multimodal learning beyond English language. This can trigger various applications that can improve multimedia search or cross-modal recommendations.

4 Approach

Variation in discrete and continuous representations of information like text and images or videos respectively require composite approaches to find semantic similarity. In this aspect, my research aims to learn correlations between media and textual content by learning joint space representation. As discussed earlier in the section 1, learning correlations between two different modalities can be divided based on tasks that support cross-modal retrieval and generation. Below, I formulate the problem for each of these tasks and explore possible approaches.

4.1 Polylingual Cross-Modal Semantic Category Retrieval

Multimodal documents are found on web in the form of pair-wise modalities. Sometimes, there can be multiple instances of modalities present in the documents. To reduce the complexity, I assume a multimodal document $D_i = (Text, Media)$ to contain a single media item (image or video) embedded with a textual description. A collection $C_j = \{D_1, D_2...D_i...D_n\}$ of these documents in different languages $L = \{L_{C_1}, L_{C_2}...L_{C_j}...L_{C_m}\}$ are spread across web. Formally, my research question is to find a cross-modal semantically similar document across language collections L_{C_o} using unsupervised similarity measures on low-dimension correlation space representation. To achieve it, I propose following approach which learns correlated space between modalities in different languages for cross-modal retrieval.

Correlated Centroid Space Unsupervised Retrieval (C^2SUR) [37] In this approach, I find correlated low-dimension space of each text and media (Image) with kernel canonical correlation analysis (kCCA) modified with k-means clustering.

Let $m_T = \{m_{T_1}...m_{T_k}\}$ and $m_I = \{m_{I_1}...m_{I_k}\}$ denote the initial k centroids for the correlated text and image space respectively obtained with kCCA. Iterating over the samples of the training data, I perform assignment and update steps to obtain the final k centroids. The assignment step assigns each observed sample to its closest mean, while the update step calculates the new means that will be a centroid.

Correlated low-dimension space of text and image samples of the training data is given by CS_{Tr_T} and CS_{Tr_I} respectively. Choice of k is dependent on number of classes in the training data, while p represents the total training samples. $S_{T_i}^{(t)}$ and $S_{I_i}^{(t)}$ denote new samples of text and image modalities assigned to its closest mean. Algorithm 1 lists the procedure. Now the modified feature space is used for cross-modal retrieval with distance metrics like cosine etc.

Experimental Data and Evaluation To evaluate the approach in a polylingual scenario, I use the wiki dataset[2] containing 2866 English texts and images

[2] http://www.svcl.ucsd.edu/projects/crossmodal/

Algorithm 1 Correlated Centroid Space

Require: $CS_{Tr_T} = x_{T_1}...x_{T_p}$, $CS_{Tr_I} = x_{I_1}...x_{I_p}$
Ensure: $p > 0$ {**Output**: Final K-Centroids}
Assignment Step:
$S_{T_i}^{(t)} = x_{T_j} : ||x_{T_j} - m_{T_i}|| \leq ||x_{T_j} - m_{T_{i^*}}||\forall i^* = 1...k$
$S_{I_i}^{(t)} = x_{I_j} : ||x_{I_j} - m_{I_i}|| \leq ||x_{I_j} - m_{I_{i^*}}||\forall i^* = 1...k$
Update Step:
$m_{T_i}^{(t+1)} = \frac{\sum_{x_{T_j} \in S_{T_i}^{(t)}} x_{T_j}}{|S_{T_i}^{(t)}|}$, $m_{I_i}^{(t+1)} = \frac{\sum_{x_{I_j} \in S_{I_i}^{(t)}} x_{I_j}}{|S_{I_i}^{(t)}|}$

created using Wikipedia's featured articles is expanded to German and Spanish [3] while keeping the original images for every language. Thus, the expanded dataset consists of text and image pairs in three different languages. Evaluation for cross-modal retrieval will be done with mean average precision (MAP) [12, 13] and mean reciprocal rank (MRR) scores. Experiments are 10 fold cross-validated to reduce selection bias.

4.2 Polylingual Cross-Modal Description

Description of a given video or an image depends on the generation of text. To achieve it several approaches are designed with dependency on predefined textual templates and image annotations. Objects identified in an image is used to fill predefined templates to generate descriptions. Though these kind of approaches imposes limits, may still have the advantage that results are more likely to be syntactically correct. Also, it limits its generalization to many languages.

Few other approaches overcame these limitations by generating grammar, though they pose similar issues as earlier in a polylingual scenario. In this aspect, I find my research question of language independent multimodal learning approaches. Recently, for image descriptions two different approaches proposed using multimodal log-bilinear model(MLBL) [9] and recurrent neural network(mRNN) [10] which does not use language specific information and can show impressive results if applied to different languages. mRNN is feed with image and textual features generated with region convolution neural networks (RCCN) and continuous word representations respectively. Now to generate descriptions, an idea similar to Long Short-Term Memory (LSTM) [38] is used for a sequence model. LSTM is helpful to decode the vector into a natural language string. Similar approaches were extended to videos [11].

In this aspect, my research aims to learn multilingual space for textual features along with image or videos to support polylingual multimodal learning. Multilingual space will help to produce descriptions for the languages that are represented in the space. Considerable research has been done in learning multilingual space in NLP community to support cross-language applications. Re-

[3] http://people.aifb.kit.edu/amo/data/Text-Ger-Spa.zip

cently, multilingual models are developed for compositional distributional semantics [39].

My research aims to use multilingual space of continuous word representations combined with CNN as an input to modified mRNN to support polylingual cross-modal description. Generated descriptions are further verified for its correctness and readability with Knowledge bases(KB) concepts, entities and common sense facts.

Experimental Data and Evaluation There are several data sets available for English text and images or videos. Flickr8K, Flickr30K and COCO[4] datasets contain images and descriptions, while ImageCLEF [5] in subtask-2 provide images with annotations to generate descriptions. Though there are few datasets for textual descriptions in multiple languages, IAPR TC-12 benchmark [6] provide each image with an associated text caption in three different languages (English, German and Spanish). For evaluation, The BLEU scores [40] are used to evaluate generated sentence by measuring the fraction of n-grams that appear in the ground truth.

5 Results and Work in Progress(WIP)

Below, I present the initial results obtained for polylingual cross-modal semantic category retrieval and discuss the work in progress (WIP) for polylingual cross-modal description.

5.1 Polylingual Cross-modal Semantic Category Retrieval (Results)

Table 1 shows the initial results obtained on text and image queries for English, German and Spanish on the Wiki dataset. I used polylingual topic models(PTM) [22] to extract textual features as a distribution of topics in multiple languages, while each image is represented as 128-dimension SIFT descriptor histogram. MAP scores for C^2SUR for German and Spanish with different topic variations. For example, C^2SUR-10 represents 10-topics. Please note, that the related work can only be applied to English text.

5.2 Polylingual Cross-Modal Description (WIP)

For polylingual cross-modal descriptions, significant contribution comes from building multilingual space of languages. Currently, I am working on building multilingual space of word embeddings for one or more languages using class aligned document corpora and sentence aligned parallel corpora. This is achieved using noise contrastive large-margin updates which ensure non-aligned parallel sentences and non-aligned classes documents observe a certain margin from each other.

[4] http://mscoco.cloudapp.net/
[5] http://www.imageclef.org/2015/annotation
[6] http://www.imageclef.org/photodata

(Language)	System	**Image Query**	**Text Query**	**Average (MAP)**
English	SM [12]	0.225	0.223	0.224
	Mean-CCA [13]	0.246 ± 0.005	0.194 ± 0.005	0.220 ± 0.005
	SCDL [41]	0.252	0.198	0.225
	SliM^2 [42]	0.255	0.202	0.229
	GMLDA [24]	0.272	0.232	0.252
	C^{2}SUR-10	**0.273 ± 0.002**	**0.262 ± 0.003**	**0.268 ± 0.003**
German	C^{2}SUR-10	**0.284 ± 0.002**	**0.263 ± 0.003**	**0.276 ± 0.003**
	C^{2}SUR-100	0.236 ± 0.004	0.250 ± 0.008	0.243 ± 0.006
	C^{2}SUR-200	0.278 ± 0.002	0.253 ± 0.002	0.266 ± 0.002
Spanish	C^{2}SUR-10	0.250 ± 0.001	**0.268 ± 0.002**	**0.259 ± 0.002**
	C^{2}SUR-100	0.258 ± 0.008	0.243 ± 0.004	0.251 ± 0.006
	C^{2}SUR-200	**0.267 ± 0.003**	0.244 ± 0.002	0.256 ± 0.003

Table 1. Text and Image Query Comparison **(Wiki)**

6 Conclusion

In this proposal, I presented my research on jointly learning heterogeneous features generated from two different modalities mainly polylingual text and image or videos. I aim to do this by segregating the approach to two different tasks. In the first task, textual information and media (image or video) is mapped to the same category with cross-modal retrieval. While in the second task, more sophisticated approaches are used to generate one modality from another. Inherently, these tasks provide better support to search, recommendations, analytics and advertising based multimedia applications.

7 Acknowledgements

I would like to thank Achim Rettinger (rettinger@kit.edu) and Rudi Studer (rudi.studer@kit.edu) for their guidance. This research has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 611346.

References

1. Zheng, L., Wang, S., Liu, Z., and Tian, Q. Packing and padding: Coupled multi-index for accurate image retrieval. In Computer Vision and Pattern Recognition (CVPR). (2014) 1947–1954
2. Lew, M. S. Special issue on video retrieval. International Journal of Multimedia Information Retrieval. (2015) 1–2
3. Moran, S., and Lavrenko, V. Sparse kernel learning for image annotation. In Proceedings of International Conference on Multimedia Retrieval. (2014)
4. Papandreou, G., Chen, L. C., Murphy, K., and Yuille, A. L. Weakly-and semi-supervised learning of a DCNN for semantic image segmentation. arXiv preprint arXiv:1502.02734. (2015)

5. Zhu, X., and Ramanan, D. Face detection, pose estimation, and landmark localization in the wild. In Computer Vision and Pattern Recognition (CVPR). (2012) 2879–2886
6. Tapaswi, M., Bauml, M., and Stiefelhagen, R. Improved Weak Labels using Contextual Cues for Person Identification in Videos. In IEEE Intl. Conf. on Automatic Face and Gesture Recognition (FG) (Vol. 4). (2015)
7. Rafailidis, D., Manolopoulou S., and Daras P.: A unified framework for multimodal retrieval. Pattern Recognition. **46.12** (2013) 3358–3370
8. Mishra, Anand, Karteek Alahari, and C. V. Jawahar.: Image Retrieval using Textual Cues. IEEE International Conference on Computer Vision(ICCV). (2013)
9. Kiros, R., Salakhutdinov, R., and Zemel, R. Multimodal neural language models. In Proceedings of the 31st International Conference on Machine Learning. (2014) 595–603
10. Karpathy, A., and Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. arXiv preprint arXiv:1412.2306. (2014)
11. Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., and Saenko, K. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. arXiv preprint arXiv:1412.4729. (2014)
12. Rasiwasia, Nikhil, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos.: A new approach to cross-modal multimedia retrieval. In Proceedings of the international conference on Multimedia. (2010) 251–260
13. Rasiwasia, Nikhil, Dhruv Mahajan, Vijay Mahadevan, and Gaurav Aggarwal.: Cluster Canonical Correlation Analysis. In Proceedings of the Seventeenth AISTATS. (2014) 823–831.
14. Klementiev, A., Titov, I., and Bhattarai, B. Inducing crosslingual distributed representations of words. In COLING. (2012)
15. Peters, C., Braschler, M., and Clough, P.: Cross-Language Information Retrieval. Multilingual Information Retrieval. (2012) 57–84
16. Zhou, D., Truran, M., Brailsford, T., Wade, V., and Ashman, H. Translation techniques in cross-language information retrieval. ACM Computing Survey 45, 1, Article 1. (2012)
17. Dumais, Susan T., Todd A. Letsche, Michael L. Littman, and Thomas K. Landauer. Automatic cross-language retrieval using latent semantic indexing. In AAAI spring symposium on cross-language text and speech retrieval, vol. 15, p. 21. (1997)
18. Vinokourov, A., Shawe-Taylor, J., and Cristianini, N. Inferring a semantic representation of text via cross-language correlation analysis. Advances in neural information processing systems, (2003) 1497–1504
19. Navigli, Roberto, and Simone Paolo Ponzetto. BabelRelate! A Joint Multilingual Approach to Computing Semantic Relatedness. In AAAI. (2012)
20. Huang, Hsun-Hui, and Yau-Hwang Kuo. Cross-lingual document representation and semantic similarity measure: a fuzzy set and rough set based approach. Fuzzy Systems, IEEE Transactions on 18.6: (2010) 1098–1111
21. Blei, D. M. Probabilistic topic models. Communications of the ACM, 55(4). (2012) 77–84
22. Mimno, David, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. Polylingual topic models. In Proceedings of Empirical Methods in Natural Language Processing(EMNLP). (2009) 880–889
23. Vuli, I., and Moens, M. F. Probabilistic models of cross-lingual semantic similarity in context based on latent cross-lingual concepts induced from comparable data. In EMNLP. (2014)

24. A. Sharma, A. Kumar, H. Daume, and Jacobs D.: Generalized multiview analysis: A discriminative latent space. Computer Vision and Pattern Recognition (CVPR). (2012)
25. Zhai, Xiaohua, Yuxin Peng, and Jianguo Xiao.: Learning Cross-Media Joint Representation with Sparse and Semi-Supervised Regularization. IEEE Journal. (2013)
26. Srivastava, N., and Salakhutdinov, R. R. Multimodal learning with deep boltzmann machines. In Advances in neural information processing systems. (2012) 2222–2230
27. Ordonez, V., Kulkarni, G., and Berg, T. L. Im2text: Describing images using 1 million captioned photographs. In Advances in Neural Information Processing Systems. (2011) 1143–1151
28. Xu, R., Xiong, C., Chen, W., and Corso, J. J. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In Proceedings of AAAI Conference on Artificial Intelligence. (2015)
29. Lin, D., Fidler, S., Kong, C., and Urtasun, R. Visual semantic search: Retrieving videos via complex textual queries. In Computer Vision and Pattern Recognition (CVPR). (2014) 2657–2664
30. Rohrbach, A., Rohrbach, M., Tandon, N., and Schiele, B. A Dataset for Movie Description. arXiv preprint arXiv:1501.02530. (2015)
31. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition(CVPR). (2009) 248–255
32. Havasi, C., Speer, R., and Alonso, J. ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In Recent advances in natural language processing. (2007) 27–29
33. Qi, G. J., Aggarwal, C., and Huang, T. Towards semantic knowledge propagation from text corpus to web images. In Proceedings of the 20th international conference on WWW. (2011) 297–306
34. Chen, X., Shrivastava, A., and Gupta, A. Neil: Extracting visual knowledge from web data. In Computer Vision (ICCV). (2013) 1409–1416
35. Mathews, A., Xie, L., and Xuming He. Choosing Basic-Level Concept Names using Visual and Language Context. IEEE Winter Conference on Applications of Computer Vision (WACV). (2015)
36. Silberer, C., and Lapata, M. Learning grounded meaning representations with autoencoders. In ACL. (2014) 721–732
37. Mogadala, A., and Rettinger, A. Multi-modal Correlated Centroid Space for Multilingual Cross-Modal Retrieval. In Advances in Information Retrieval. (2015) 68–79
38. Hochreiter, S., and Schmidhuber, J. Long short-term memory. Neural computation, 9(8). (1997) 1735–1780.
39. Hermann, K. M., and Blunsom, P. Multilingual Models for Compositional Distributed Semantics. In ACL. (2014)
40. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311318. Association for Computational Linguistics. (2002)
41. Wang, S., Zhang, L., Liang, Y., and Pan, Q.: Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In Computer Vision and Pattern Recognition (CVPR). (2012) 2216–2223
42. Zhuang, Y., Wang, Y., Wu, F., Zhang, Y., and Lu, W.: Supervised coupled dictionary learning with group structures for multi-modal retrieval. In Proceedings of 25th AAAI. (2013)

Combining Social and Official Media in News Recommender Systems

Nuno Moniz and Luís Torgo

LIAAD - INESC Tec
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
nmmoniz@inescporto.pt,ltorgo@dcc.fc.up.pt

Abstract. The exponential growth of available information to end-users poses many questions and challenges to researchers. This research originates from the issues raised by the high demand both in quality and quickness by the end-user concerning news recommender systems. The objective of this research is to combine real-time official and social recommendations of news articles incorporating an approach capable of addressing the issues related to news recency, through the prediction of news importance, but also, to handle the discrepancies between the official media and social recommendations. This research is threefold. On the first hand, the issue of obtaining the information necessary for this endeavour. On the second hand, the prediction of future importance for a given news article and its real-time adjustment taking into consideration the evolution of the respective attributed importance. And finally, the interflow of official media and social rankings.

Keywords: predictive models, resampling strategies, recommender systems, social and official media

1 Introduction

The possibility of each user publicly making available any information, may it be an individual person or organization, provoked an explosion of available information and a growing demand concerning the computation capability to analyse, interpret and act upon this referred information. This is most visible concerning the users demand for information.

This demand is clear when the evolution in both information retrieval (*e.g.* search engines) and filtering (*e.g.* recommender systems) are considered. These systems, factoring or not the influence of a given social network or other mediums, provide a ranked recommendation of resources containing what has been classified by itself as most relevant to a given query or profile. Therefore, these suggestions are based on data that ranges from a given point in the past to the present. This opens the issue dealt with this research.

Time consists of past, present and future. The suggestions provided by search engines and recommender systems are in most cases based on the analysis, computation and production of rankings given past data although some of them

provide a recommendation of resources that are very active in the present, such as Digg, by incorporating information provided by its own users in real-time. This process may also be done using social networks such as Twitter which provides a dynamic, multi-topic, real-time medium to determine the popularity or importance of the news articles through their respective number of publications. These publications are limited to 140 characters and are called tweets.

News recommendation systems are a good example to explain the advantages of social networks data. When a given news article is published there is no available social networks data concerning that publication. Therefore, we have a latency problem related to its recency and the most recent articles will take a certain period of time until they are considered. In the case of Digg, real-time social data is factored to shorten the latency in helping the determination of its importance or interest. It should be noted that the concrete process of rank production of these systems is not public although there are some clues available.

A second point relating to the use of data from social networks is also important: the difference between the recommendation made by these systems and the importance attributed by the public. This is approached in DeChoudhury et al. [4] where relevance to the end user is addressed concluding that the most popular algorithms (HITS and PageRank) would probably not point out the most relevant information comparing with that recovered from Twitter.

Therefore, although it is possible to provide a recommendation with fair acceptance given the past and present data, as the referred systems do, the possibilities when considered the prediction of future data are scarce. The importance of this prediction is to minimize the referred latency and enable the treatment of recent news without a waiting period.

Additionally, it is known that the distribution of news articles in Twitter and their number of tweets are described by a power-law distribution. This shows that only a small portion of cases are in fact highly shared, and therefore important to the public. Our interest is to predict these cases in order to favour them concerning the social recommendation of news articles.

This research intends to address two issues referred in this section. First, the problem of latency regarding news articles recency. Second, the discrepancy between the recommendations made by the official media and the public. Therefore, the objective of this research is to combine real-time official and social recommendations of news articles incorporating an approach capable of addressing the issues regarding latency, through the prediction of news importance. This research is threefold. On the first hand, the issue of obtaining the information necessary for this endeavour. On the second hand, the prediction of future importance for a given news article and its real-time adjustment taking into consideration the evolution of the respective attributed importance. And finally, combining official media and social rankings.

2 Background

This section provides an overview of two main areas in which this research is based. The first area is Information Filtering, with a specific emphasis on Recommender Systems. The second is Prediction Models, specifically, on the subject of predicting the future importance of news. To be concrete, when referring to importance, this is translated as the importance given by the public to a given news article, measured by the respective number of tweets.

2.1 Recommender Systems

Concerning recommender systems, they provide the ability to assist the users by suggesting what the best choices should be in a given scope. The following paragraphs depict examples of those possibilities related to news recommender systems, the focus of our research.

Phelan et al. [10] describe an approach which includes harnessing real-time information from Twitter in order to promote news stories. For this endeavor, the authors achieve a basis for story recommendation by mining Twitter data, identifying emerging topics of interest and matching them with recent news coverage from RSS feeds. This work is extended by Phelan et al. [9] through the increase in comprehension and robustness of the recommendation framework, using different sources of recommendation knowledge and strategies. Hannon et al. [6] applies and evaluates several profiling and recommendation strategies in order to recommend followers and/or followees, using Twitter data as a basis for both processes. Abrol and Khan [1] proposes TWinner, a tool capable of combining social media to improve quality of web search and predicting whether the user is looking for news or not using a query expansion approach.

The application "Hotstream" is proposed by Phuvipadawat and Murata [11]. This application uses a method designed to collect, group, rank and track breaking news in Twitter. The ranking process uses popularity and reliability factors. Reliability reports to the numbers of followers from posting users. The authors established that popularity is determined from the numbers of retweets.

The referred approaches are similar to the one proposed in this research. The main difference is that in every case, the issue of latency is not addressed any further. Nevertheless, the work referred presents various options for discussion regarding the combination of official media and social recommendations. The main difference that this work provides is related to the following section.

2.2 Importance Prediction

Determining the importance of a given news is a very interesting variable when referring to news-based recommender systems. The ability to describe documents as being more or less important to the public is a crucial variable to consider. This has been pursued by combining documents from legacy media sources (*e.g.* newspapers), and the produced content in social media by their users.

A considerable portion of research concerning prediction models using Twitter data (*e.g.* [13], [19]) has been focused on the retweet function of the platform (*i.e.* the ability to re-publish a given tweet). The dynamics of the retweeting function are thoroughly discussed in the work of Rudat et al. [12].

In the work of Bandari et. al [2] classification and regression algorithms are examined in order to predict popularity of articles in Twitter. The distinguishing factor of this work from others that attempt to predict popularity of events (*e.g.* [14], [15]), is that it attempts to do this prior to the publication of the item. To this purpose, the authors used four features: source of the article, category, subjectivity in the language and named entities mentioned.

Dilrukshi et al. [5] uses machine learning techniques, namely SVM (Support Vector Machine), to classify tweets according to 12 distinct groups. The purpose of the authors research is to enable the identification of the most popular news group for a given country, within a given time frame.

The referenced work provides an overview of the state of art regarding our scope in this research. Concerning the prediction of importance, as it was referred previously, we are not attempting to predict only the number of tweets a given news will obtain, but also the prediction of whether it is a rare case or not. And although the referenced work obtains enthusiastic results they are not focused on these rare cases. As such, given that we are dealing with a power-law distribution, where the rare cases of highly tweeted news represent a very small part, our interest is to identify them correctly and not the great majority of the cases that present a very low number of publications.

3 Contribution

The goal behind our proposal is to prove that by combining official media and social recommendations it is possible to lessen the discrepancies between them, but also, to tackle the latency problem related to news articles recency. To this endeavour, it is necessary to collect information in order to obtain the news articles on one hand, and on the other, the real-time importance, attributed by the public. Upon detection, information concerning the news articles is used to predict their importance after a certain period of time. Then, in combination with the evolution of the importance attributed by the public, the predictions are continuously updated in order to factor the public reaction. The final output is a recommendation on which news articles concerning a given topic are or will be, after a certain period of time, considered to be highly important. The operationalization of our proposal is done through a three-step pipeline: (i) information retrieval, (ii) recency rank adaptor and (iii) rank aggregator.

3.1 Information Retrieval

Concerning the first step, and recalling the objectives set out by this research, it requires information from two types of sources: official media and social recommendations. In order to collect that information it is necessary to find sources which enable a stable process of retrieval.

At this point, we collect information from two sources: Google News and Twitter. Google News is a news recommender system (official media recommendations) and Twitter is a micro-blogging platform (social media recommendations). Both sources enable the extraction of information through organized parsing, in the case of the former, and a public API, in the case of the latter.

The first provides a real-time collection of published news, with information such as the title, subtitle, media source, publish date, direct link and the on-time rank according to the Google News. This will be referred as the Media Rank. One of our objectives is to include official media recommendations from multiple sources (*e.g.* Yahoo News).

The second enables users to post information onto the web in real-time and interact with others. In our case, Twitter is used to judge the public attributed importance to each of the news in the documents set provided by Google News. By using the Twitter API, it is possible to continuously retrieve the number of tweets associated to each news item in a given set of topics. We decided to establish a two day limit concerning the timespan of retrieval of information from Twitter for each given news, based on the work of Yang and Leskovec [18] which suggests that after a few days the news stop being tweeted. Despite the results of the referred authors research which indicates that this period could achieve four days, some initial tests on our data sets have shown that after a period of two days the number of tweets is residual, and therefore we chose this time interval.

3.2 Recency Rank Adaptor

The second step encompasses two processes concerning the prediction of the future importance a given news will obtain. These processes are separated due to the approach embedded in each of them. The first is an *a priori* approach, and the second, an *a posteriori* approach.

This step deals with two connected issues reporting to the recency issues described formerly. The first is the fact that there is no available information concerning the public opinion on a given news item upon its publication. In this case we need prediction models which are capable of determining the number of tweets of news items with special focus on the rare cases, having no related available data (*a priori*). The second issue is that even after its publication (*a posteriori*), we have different levels of information portraying the news items. As such, we need a prediction approach which is capable of tackling this unbalance and accurately predict rare cases of news items with a high number of tweets having scarce data (*i.e.* in the moments after the publication of a given news item the amount of related information is small).

Concerning the *a priori* models and the skewed distribution of the number of tweets, previous work [16, 17] has shown that standard regression tools fail on tasks where the goal is accuracy at the rare extreme values of the target variable. Several methodologies have been proposed for addressing this type of tasks and resampling methods are among the simplest and most effective. We experimented with two of the most successful resampling strategies: SMOTE [3]

and under-sampling [7]. On our models we use the extension for regression tasks of SMOTE (SMOTEr) and under-sampling proposed by Torgo and Ribeiro [17].

As for the *a posteriori* models, we are developing four algorithms which represent the combination of two selection and two prediction approaches. In this situation, we have information concerning the real-time number of tweets a given news obtains. Therefore, concerning the selection approaches, we tested the use of interquartile range (IQR) on the target variable, with and without prior probability distribution. As for prediction, we tested the use of a weighted and a scalar approach. The weighted approach uses distance within the IQR range as a weight variable. Therefore, for each case in the train set that for a given timeslice (periods of 20 minutes since the publication time) the target value is within the IQR range, the distance to the value of the test case is normalized in a $[0, 1]$ scale and multiplied by its respective final number of tweets. Finally, the sum of all these cases is divided by the overall sum of weights. As for the scalar approach, it is based on the calculation of the average slope of train cases within the IQR range of a given test case, concerning the present timeslice and the final timeslice. This is multiplied by the number of timeslices remaining (considering the referred two days limit for obtaining information regarding a given news item) and added to the present number of tweets the test case has obtained.

The resulting predictions from these models are then combined and produce the Public Opinion Ranking. This procedure takes into account the alive-time of each given news item. The logic is simple: as time evolves and more information regarding each given news item is available, the predictions based on real-time data are more reliable than those of the *a priori* prediction. Therefore, the combination of both the prediction tasks, *a priori* and *a posteriori*, is executed with the decay and increase of their respective weight in the final predicted value. One of the key elements of this combination is the discovery of the tipping point, or the moment when this shift in terms of predictive reliability occurs.

3.3 Rank Aggregator

So far, two distinct ranks have been mentioned. The rank provided by Google News which is referred as Media Rank, and the Public Opinion Rank described in the former section. This step holds the objective of producing a final suggestion based on both of them which is continuously updated. Therefore, this step aggregates these two ranks taking into consideration their respective weights.

The Media Rank provides an insight to the past and present, and the Public Media Rank on the predicted future importance and therefore, rank. We have not developed any research concerning the combination of both ranks yet, but given the results from the previous sections, it is possible that the evolution in terms of weights should be depicted in the form of a power-law distribution (convex and concave), and this will be the research focus concerning this component.

This component provides the final output of the proposal, a ranked suggestion of news articles that combines the present and the prediction of future importance for each of the articles.

4 Experiments and Results

This section reports to the experiments and results which were obtained so far. These refer to the prediction models described in Section 3.2.

The experiments are based on news concerning four specific topics: economy, microsoft, obama and palestine. These topics were chosen due to two factors: its actual use and because they report to different types of entities.

Concerning the *a priori* approach, for each of the topics we constructed a dataset with news mentioned in Google News between 2014-May-01 and 2014-Sep-03, with queries of the top 100 news every 15 minutes. For each item the following information was collected: title, headline, publication date and its position in the ranking. These datasets were built using the Twitter API to check the number of times the news were tweeted in the two days following its publication, which represents our target variable value. As stated, the distribution of the values of the target variable is highly skewed. Therefore, we applied the previously described re-sampling strategies SMOTEr and under-sampling. The evaluation of the models is based on the utility-based regression framework proposed in the work of Torgo and Ribeiro [16]. The metrics proposed assume that the user is able to specify the most relevant sub-range of target variable values. This is done by specifying a relevance function that maps the target variable into a $[0, 1]$ scale of relevance. Using this mapping and a user-provided relevance threshold the authors defined a series of metrics that focus the evaluation of models on the cases that matter for the user. These experiments are described in detail in Moniz and Torgo [8]. The evaluation of the prediction models is presented in Table 1, using three metrics: precision, recall and the F1 measure. From the perspective of our application, we focus on the F1 measure, because it penalises false positives (i.e. predicting a very high number of tweets for a news that is not highly tweeted). The evaluation of the rankings produced using the results of the former are presented in Table 2, using three metrics: mean reciprocal rank and the normalized discounted cumulative gain with $k = 10$ and $k = 50$. For each regression algorithm the best estimated scores are denoted in italics, whilst the best overall score is in bold.

Table 1. Precision, Recall and F1-Score estimated scores for all topics, for the *a priori* approach.

	economy			microsoft			obama			palestine		
	prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
lm	0.23	0.05	0.08	0.09	0.03	0.04	0.15	0.00	0.00	0.18	0.09	0.12
lm_SMOTE	*0.64*	*0.26*	*0.37*	0.49	*0.23*	*0.31*	0.53	*0.39*	*0.45*	0.54	*0.14*	*0.22*
lm_UNDER	*0.64*	0.23	0.34	*0.50*	0.20	0.29	*0.55*	0.38	*0.45*	*0.55*	0.09	0.15
svm	0.46	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.01
svm_SMOTE	0.67	0.52	0.58	*0.66*	**0.59**	**0.62**	**0.68**	**0.71**	**0.69**	**0.83**	**0.54**	**0.65**
svm_UNDER	*0.70*	**0.55**	**0.62**	0.64	**0.59**	**0.62**	0.65	0.70	0.68	0.80	**0.54**	**0.65**
mars	0.18	0.02	0.04	0.05	0.01	0.02	0.31	0.01	0.01	0.16	0.07	0.10
mars_SMOTE	0.67	0.39	0.49	0.51	0.34	0.41	0.54	0.50	0.52	0.53	0.23	0.32
mars_UNDER	**0.76**	*0.52*	*0.61*	**0.67**	*0.47*	*0.55*	*0.62*	*0.61*	*0.62*	*0.75*	*0.41*	*0.52*
rf	0.28	0.03	0.06	0.13	0.02	0.03	0.31	0.01	0.02	0.09	0.03	0.04
rf_SMOTE	0.67	*0.51*	*0.58*	0.50	0.48	0.49	0.53	0.61	0.57	0.62	*0.43*	0.51
rf_UNDER	*0.73*	0.46	0.57	*0.64*	*0.51*	*0.56*	*0.63*	*0.65*	*0.64*	*0.76*	*0.43*	*0.54*

Table 2. MRR, NDCG@50 and NDCG@10 scores for all topics, for the *a priori* approach.

	economy			microsoft		
	MRR	NDCG50	NDCG10	MRR	NDCG50	NDCG10
svmSMOTE	0.61	**0.71**	0.75	**0.59**	**0.73**	**0.74**
svmUNDER	**0.63**	**0.71**	**0.76**	**0.59**	**0.73**	**0.74**
marsSMOTE	0.23	0.49	0.49	0.27	0.52	0.53
marsUNDER	0.26	0.52	0.56	0.28	0.55	0.59
rfSMOTE	0.23	0.49	0.50	0.26	0.52	0.52
rfUNDER	0.22	0.49	0.50	0.25	0.52	0.53
	obama			palestine		
svmSMOTE	0.29	**0.53**	**0.52**	**0.46**	**0.69**	**0.69**
svmUNDER	**0.30**	**0.53**	**0.52**	**0.46**	**0.69**	**0.69**
marsSMOTE	0.18	0.43	0.42	0.38	0.64	0.65
marsUNDER	0.18	0.43	0.42	0.38	0.64	0.66
rfSMOTE	0.22	0.45	0.45	0.31	0.62	0.63
rfUNDER	0.19	0.43	0.43	0.32	0.62	0.63

Concerning the *a posteriori* approach, the dataset was constructed with data on news items that appeared in Google News over a timespan of one month. The items collected for this dataset were obtained through queries of the previously referred topics where each top 100 news were retrieved. The queries were made every 20 minutes during the referred timespan. The information collected for each of the items is the same as the previously stated. An auxiliary dataset was built in order to enable the analysis of the evolution in number of tweets of all news items. To obtain this data, the Twitter API was also used, in 20 minute intervals, since the first moment the news was recommended by Google News until two days past from its original publication date. The evaluation of this research is also based on the utility-based regression previously referenced and is presented in Table 3. The evaluation of the rankings produced using the results of the former and the comparison to the Google News evaluation are presented in Table 4 using four metrics: mean average precision, mean r-precision, mean reciprocal rank and normalized discounted cumulative gain with $k = 10$. The best scores in each metric are denoted in bold.

Table 3. Precision, Recall and F1-Score estimated scores for all topics, for the *a posteriori* approach.

	economy			microsoft			obama			palestine		
Approach	prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
Weighted	0.657	**0.801**	**0.718**	0.518	**0.908**	0.654	0.607	**0.871**	**0.712**	**0.537**	**0.943**	**0.683**
Weighted+Prior	0.656	0.790	0.712	0.518	0.902	0.652	0.605	0.864	0.708	0.536	0.937	0.680
Scalar	**0.658**	0.796	0.715	**0.520**	**0.908**	**0.656**	**0.608**	0.867	0.711	0.536	0.940	0.681
Scalar+Prior	0.657	0.785	0.711	**0.520**	0.902	0.654	0.607	0.859	0.708	0.536	0.936	0.680

5 Future Work

So far our research has been focused on the study of modeling techniques that are able to accurately forecast the rare cases of highly tweeted news items, with the objective of enabling their prompt recommendation. These news are rare

Table 4. MAP, MRP, MRR and NDCG estimated scores for all topics, for the *a posteriori* approach.

	economy				microsoft			
Approach	MAP	MRP	MRR	NDCG	MAP	MRP	MRR	NDCG
Google	0.161	0.170	0.333	0.611	0.231	0.204	0.409	0.626
Weighted	**0.715**	**0.795**	**0.799**	**0.871**	**0.798**	**0.845**	**0.905**	**0.912**
Weighted+Prior	0.659	0.757	0.762	0.842	0.780	0.831	0.891	0.902
Scalar	0.713	0.794	**0.799**	0.870	**0.798**	**0.845**	**0.905**	**0.912**
Scalar+Prior	0.657	0.757	0.761	0.841	0.779	0.832	0.891	0.902
	obama				palestine			
Approach	MAP	MRP	MRR	NDCG	MAP	MRP	MRR	NDCG
Google	0.175	0.156	0.398	0.596	0.148	0.175	0.315	0.568
Weighted	**0.893**	0.891	**0.899**	**0.954**	**0.929**	**0.933**	**0.952**	**0.970**
Weighted+Prior	0.858	0.875	0.878	0.937	0.921	0.924	0.950	0.967
Scalar	**0.893**	**0.898**	**0.899**	0.953	0.928	0.932	**0.952**	**0.970**
Scalar+Prior	0.859	0.878	0.878	0.937	0.920	0.922	0.950	0.967

and this poses difficult challenges to existing prediction models. We evaluated proposed methods for addressing these problems in our particular task and confirmed the hypothesis that resampling methods are an effective and simple way of addressing the task of predicting when a news item will be highly tweeted upon its publication. Also, we approach this problem in a *a posteriori* context, with a stream of real-time data on the popularity of news items with different alive-time and different levels of available information for each of the news items. The evaluation of the prediction models and the rankings produced based on the four algorithms proposed show that they are capable of achieving good results, with a small overall advantage to the combination of the non-prior probability distribution selection algorithm and the weighted prediction algorithm.

Concerning future work, the recency rank adaptor requires further research. Although the evaluation of the *a priori* approach has been consolidated with an extended dataset, that is not true for the *a posteriori* approach. As such, this is the next step in our research. Thereon, it is necessary to study and develop an approach capable of dynamically combining both predictions, through the incorporation of the tipping point, previously described.

The output of the recency rank adaptor, the Public Opinion Rank, is to be combined with Media Rank. This step is still an open research question which will be addressed. Nonetheless, the focus of the research should be the dynamics of the weights of both ranks (Public Opinion and Media) along the time dimension.

Finally, the objective is to build a prototype which encompasses the research developed in order to enable the real-time evaluation of the general proposal in comparison to the well-known news recommender systems such as Google News.

6 Acknowledgments

This work is financed by the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project UID/EEA/50014/2013. The work of N. Moniz is supported by a PhD scholarship of FCT (SFRH/BD/90180/2012). The authors would like to thank the reviewers for their comments and suggestions.

References

[1] Abrol, S., Khan, L.: Twinner: understanding news queries with geo-content using twitter. In: Proc. of the 6th Workshop on Geographic Information Retrieval. pp. 10:1–10:8. ACM (2010)

[2] Bandari, R., Asur, S., Huberman, B.A.: The pulse of news in social media: Forecasting popularity. CoRR (2012)

[3] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. JAIR 16, 321–357 (2002)

[4] De Choudhury, M., Counts, S., Czerwinski, M.: Identifying relevant social media content: leveraging information diversity and user cognition. In: Proc. of the ACM HT '11. pp. 161–170. ACM (2011)

[5] Dilrukshi, I., de Zoysa, K., Caldera, A.: Twitter news classification using svm. In: 8th Edition of ICCSE. pp. 287–291 (April 2013)

[6] Hannon, J., Bennett, M., Smyth, B.: Recommending twitter users to follow using content and collaborative filtering approaches. In: Proc. of the 4th ACM RecSys. pp. 199–206. ACM (2010)

[7] Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: Proc. of the 14th Int. Conf. on Mach. Learn. (1997)

[8] Moniz, N., Torgo, L.: Socially driven news recommendation (arXiv:1506.01743) (Jun 2015)

[9] Phelan, O., McCarthy, K., Bennett, M., Smyth, B.: Terms of a feather: Content-based news recommendation and discovery using twitter. In: ECIR'11. pp. 448–459 (2011)

[10] Phelan, O., McCarthy, K., Smyth, B.: Using twitter to recommend real-time topical news. In: Proc. of RecSys '09. pp. 385–388. ACM (2009)

[11] Phuvipadawat, S., Murata, T.: Breaking news detection and tracking in twitter. In: Proc. of WI-IAT '10. IEEE Computer Society (2010)

[12] Rudat, Anja, B.J.H.F.W.: Audience design in twitter: Retweeting behavior between informational value and followers' interests. Computers in Human Behavior 35(Complete), 132–139 (2014)

[13] Suh, B., Hong, L., Pirolli, P., Chi, E.H.: Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In: Proc. of SOCIALCOM '10. pp. 177–184. IEEE Computer Society (2010)

[14] Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Commun. ACM 53(8), 80–88 (Aug 2010)

[15] Tatar, A., Leguay, J., Antoniadis, P., Limbourg, A., de Amorim, M.D., Fdida, S.: Predicting the popularity of online articles based on user comments. In: Proc. of WIMS '11. ACM (2011)

[16] Torgo, L., Ribeiro, R.: Utility-based regression. In: Proc. of PKDD'07. pp. 597–604. Springer (2007)

[17] Torgo, L., Ribeiro, R.P., Pfahringer, B., Branco, P.: Smote for regression. In: EPIA. vol. 8154, pp. 378–389. Springer (2013)

[18] Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proc. of WSDM'11. ACM (2011)

[19] Zaman, T., Fox, E.B., Bradlow, E.T.: A Bayesian Approach for Predicting the Popularity of Tweets. Tech. Rep. arXiv:1304.6777 (Apr 2013)

Long term goal oriented recommender systems

Amir Hossein Nabizadeh[1], Alípio Mário Jorge[2], and José Paulo Leal[3]

[1,2]LIAAD, INESC TEC, Porto, Portugal
[3]CRACS, INESC TEC, Porto, Portugal
[1,2,3]Departamento de Ciências de Computadores, Universidade do Porto, Portugal
amirhossein@dcc.fc.up.pt[1], amjorge@fc.up.pt[2], zp@dcc.fc.up.pt[3]

Abstract. Recommenders assist users to find items of their interest in large datasets. Effective recommenders enhance users satisfaction and improve customers loyalty. Current recommenders concentrate on the immediate recommendations' value and are appraised as such but it is not adequate for long term goals. In this study, we propose long term goal recommenders that satisfy current needs of users while conducting them toward a predefined long term goal either defined by platform manager or by users. A goal is long term if it is going to be obtained after a sequence of steps. This is of interest to recommend learning objects in order to learn a target concept, and also when a company intends to lead customers to purchase a particular product or guide them to a different customer segment. Therefore, we believe it is beneficial and useful to develop a recommender algorithm that promotes goals either defined by users or platform managers. In addition, we also envisage methodologies to evaluate the recommender and demonstrate the long term goal recommender in different domains.

Keywords: Recommender System, Course generation, Course sequence, Persuasive Recommender System, Learning Design, Pattern recognition, Long Term Recommender System.

1 Introduction

Current recommenders focus on the immediate needs of users. This is insufficient to obtain long term goals. Therefore, we propose Long Term Recommender Systems (LTRS) that besides satisfying immediate needs of users, conduct them toward a predefined long term goal by generating a set of relevant recommendations step by step [12]. A goal is long term if there are intermediate goals or if there is a sequence of recommendations to attain the long term goal. This goal is domain dependent and can be defined by the owner of the system or by users. Goal can be purchasing an item, learning a course, following a specific genre or singer, etc.

LTRS can be applied in different domains. For instance, in E-learning domain, LTRS aid users (e.g. teachers and learners) to have more productive activities (teaching and learning) meanwhile consuming less time. In this case, a long term goal can be defined by a teacher as doing a relevant assignment or passing an

exam after getting the long sequence of recommendations. Another example is in music domain. For example, a music company has a contract with a singer and due to some reasons the company expects to lose that singer and so it will lose a part of its music market. As a result, the company looks for solutions to retain its market and keep the same level of selling after losing that singer. One of the solutions can be diversifying the customers' taste (following other singers or other music genre) which can be done by generating a set of recommendations that influence users taste through time. Also, in the case of music, a company may use LTRS to guide the users from a preferred music genre to a target genre in order to enhance its profit on selected products. In this case, LTRS gradually influence users' interests through time.

The main research question of this study is: how can we produce recommendation sequences that successfully conduct the user to a target area in the item space, while satisfying immediate user needs? A goal can be defined as a predetermined area (in case of music, area can be a specific genre of music) in the item space of interest to both the user and the platform manager. To attain a long term goal, a recommendation algorithm must act strategically and not simply tactically. Subsequently, our main objective is to design a recommendation strategy that is able to attain strategic goals of users and platform managers.

In this study, we plan to adopt Learning Design (LD) principles and methods (such as course sequence, course generation, pattern sequence recognition) in order to build our recommender. LD is an activity to build an effective learning path by finding suitable learning objects [4]. The main advantage of LD recommenders is recommending a learning path not only based on the similarity among learning objects or among learners. It makes the generated recommendations more accurate. In addition, persuasive systems are also useful to generate our proposal. These systems were proposed by Fogg [5] in order to influence users' thoughts and behaviors, and are focused on psychological aspects of recommendations. The persuasiveness principles describes how the recommendations can be generated and represented in order to have more influence on the users [19]. Due to the fact that LTRS recommendations must be convincing for the users otherwise they do not follow the recommendations and the goal can not be obtained, therefore we believe persuasiveness principles can enhance the effectiveness of our recommendations.

The quality of a LTRS should be measured on how it can influence users' decisions and conduct the users towards a predefined target area. Although there are some techniques in order to assess the accuracy of RS such as *Precision*, *Recall* or MSE, these are not sufficient to evaluate the strategic capabilities of a LTRS. We then argue that complementary means of evaluation will be needed for LTRS.

In this paper, we propose the idea of Long Term Recommender Systems that guide users toward a predefined goal by generating relevant recommendations. LTRS will be supported by LDRS and persuasiveness principles. In addition, we plan to design a general evaluation framework in order to assess the results of LTRS and demonstrate our system in different domains.

The remainder of this paper is structured as follows. Section 2 surveys the related work methods and algorithms that are usable for a LTRS. The research methodology is detailed in Section 3 and then we conclude the paper with conclusion part.

2 Related work

2.1 Learning Design

In the area of e-learning, Learning Design is an activity to generate an effective learning path by an appropriate sequence of learning objects [4]. Learning object is any reusable digital resource which supports the learning process [4, 18]. Researchers have utilized LD principles in recommenders area in order to recommend a learning path (a set of connected learning objects) to users. According to our survey, all LD recommenders studies can be classified into three main categories: course generation, course sequence and pattern sequence recognition method.

2.1.1 Course generation This method is the most frequently used by researchers and it generates a well-ordered sequence of Learning Objects (LO) that is customized for a learner. In this approach, a user is evaluated before receiving a recommendation (diagnostic evaluation). The learning path is generated based on the diagnostic evaluation result and user profile information, including personal information along with extra information such as preferred language and media, etc. In course generation, the entire learning path is generated and recommended to a user in a single recommendation [16]. If a user was not able to follow the path to attain the final goal, the system recommends another path.

Several researchers have applied this method along with other techniques and algorithms. For example, Vassileva and Deters [17] applied decision rules in a tool that generates individual courses. This tool exploits on previous knowledge of a user and user's goals. This tool can be updated dynamically with respect to user progress. Markov decision [3], and fuzzy petri nets [8] are also other techniques that are used in the course generation approach in order to generate and recommend a learning path.

Although this method is fast due to generating and storing all the possible learning path for each user, it ignores a learner changes and performance during following a recommended path by a learner.

2.1.2 Course sequence In comparison with course generation that recommends the whole path in a single recommendation, course sequence recommends LOs one by one based on the user's progress [1]. Initially, as in course generation, this method recommends the first LO based on user profile and diagnostic evaluation result. Unlike in course generation, course sequence recommends LOs one by one and a user evaluation happens after recommending each LO.

Some studies such as [9, 10] utilized course sequence along with different algorithms and techniques to propose their methods. Karampiperis and Sampson [10] proposed their idea by utilizing Adaptive Educational Hypermedia Systems (AEHS). In their method, first all possible learning paths that obtain the goal are generated and then, the desired one (the shortest path) is selected adaptively according to a decision model. Also Idris et al. applied Artificial Neural Network in order to present an adaptive course sequencing method [9].

Although course sequence considers user changes and progress, which was one of the main issues in course generation, it still has several problems such as lacking of an effective automated method to update the user profile and also to determine what information in the user profile needs to be updated after each evaluation.

2.1.3 Pattern sequence recognition It is similar to the course generation method since both methods recommend a sequence of well-ordered learning objects to a learner. This method extracts a sequence of LOs (path) from the available data that was successful to guide a user toward a goal and recommends it to a user with a similar goal [11, 6].

One of the studies that used this method is conducted by Klasnja-Milicevic et al. [11]. In their system, they first cluster the learners w.r.t their learning style. Then they used *AprioriAll* algorithm [14] in order to mine the behavioral patterns of any learner. Finally, a recommendation list is generated based on the rates that is provided for frequent sequences.

Appriori is one of algorithms which is applied by researchers such as [11] in order to find patterns. Researchers who utilize pattern recognition method usually face two issues. Firstly, current pattern recognition methods are slow and secondly, they find frequent patterns and rare cases will be ignored.

In general, All LDRS methods have some problems such as (1) lack of a general framework to evaluate the result and compare different approaches, (2) researchers usually could not address the scalability (handle and work with big set of users and LOs), (3) lack of efficient user profile adaption method and (4) **Time** which is also a significant factor that is ignored by many researchers. A few studies addressed **time** in course modeling phase which is not efficient since user ability and background is ignored [2]. Course modeling is a process of finding essential knowledge units of a course and find their relations in order to build a course model.

2.2 Persuasive Recommendation System

The recommendations generated by LTRS should be convincing and persuade users to follow them otherwise the main goal of LTRS which is guiding the users toward a final goal could not be attained. Therefore, we need a technology to assist us to generate more convincing recommendations for users. Persuasive technology is initiated by Fogg in 2002 [5], applies computers to influence users' thoughts and actions. After Fogg several researchers utilized this technology in recommenders domain.

Persuasive RS are based on two theories: Media equation theory [15] and Communication persuasion paradigm [13]. According to the communication persuasion paradigm, a person can be affected by others in four different scopes (1) form and content, (2) source, (3) the receiver characteristics, (4) contextual factor [13]. In our case, if we see the recommender as a person that we communicate with (media equation theory), the system can be seen as a source, the user as a receiver and recommendations as messages. The whole process of recommending is set in a specific context. Recommendations persuade receivers whether to continue using the system or not [19]. In RS field, this technology focuses on psychological aspect of recommendations and clarifies how recommendations can be represented to have more effect on users.

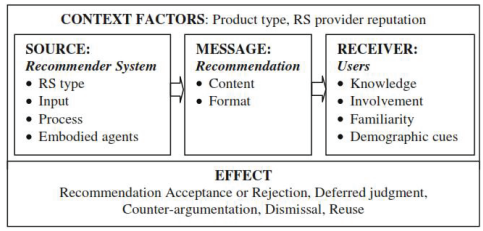


Fig. 1. Conceptual framework of persuasive RS [19].

3 Research methodology

In our proposal, we argue for the usefulness of LTRS to guide the users to a predefined goal in item space. The users are conducted toward a goal by generating a sequence of relevant recommendations in successive moments. We intend to design and develop a strategy that generates recommendations that guide the users toward a goal and also a framework in order to evaluate the success of our strategy. The proposed strategy is applicable in different domains such as E-learning, music, etc.

Figure 2 shows a conceptual view of our proposal. It shows an item space (a set of objects with different characteristics) that contains the type of objects in which the user is interested (gray highlighted area). Our strategy conducts the user towards the goal (green highlighted area) step by step, while dynamically calculates how far the target user is from the target area (i.e. assess the distance between the current position of the target user and target area after each recommendation). The purpose of each recommendation is to enlarge the interesting area (in case of E-learning it can be knowledge area) of the user's target until he reaches the items in the target area.

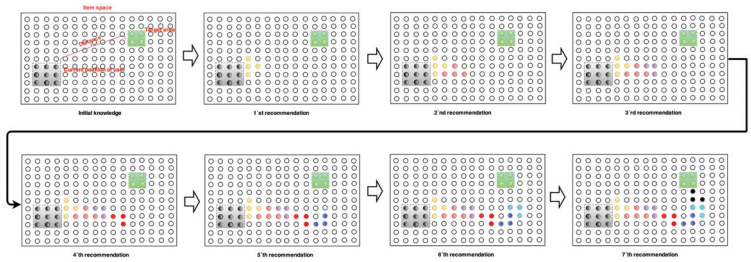


Fig. 2. Conceptual view of LTRS.

3.1 Task 1: Literature Survey and concepts definition

We already started by broadening our knowledge of the area of sequential recommender systems (e.g. learning design recommenders). Work on recommendation of structured objects in general are also of interest (sequences are a particular type of structures). User behavior studies related to recommenders such as persuasive recommender systems are also significant. In addition, in this phase, we are also interested in distance based approaches which are relevant to characterize the user trajectories in the item space and user transitions from region to region. Finally we will review existing evaluation methodologies and measures for such structured recommendation problems and specially evaluation methodologies for live environments with real users.

Furthermore, in order to design a framework for the LTRS, a few concepts should be defined: item space, target region in the item space, user location, distance between current location of user and an item and also distance between user current location and target region.

3.2 Task 2: Data feed set-up

To learn about long term interaction between users and recommenders, we are currently analyzing the log data of a music recommender that we have previously developed in the context of **Palco3.0 QREN** project. The recommender service is running for the Palco Principal website. We intend to understand how recommendations influence the evolution of users, how users react to the recommendations, their current activities and interests, etc. We collect activity data from the recommender service such as the generated recommendations and the followed recommendations.

We also look for a second application set up in the area of E-learning. We are in contact with a publishing company working in the area and also have access to programming languages tutoring environments that can be adapted to use recommendation algorithms.

3.3 Task 3: Long term user behavior and trajectory characterization

The data feed defined in Section 3.2 will be utilized in a continuous streaming fashion to identify user behavior through time and to examine the predictability of the trajectory of a user in the item space. The obtained knowledge from this phase will be significant in order to develop our strategic recommender algorithm. It will also be of interest to other researchers who are interested and work on user behavior and characterization.

3.4 Task 4: Defining a long term recommendation strategy

This phase is the main step of the study. In this phase, we plan to define a strategy that learns from user activities and generates a series of recommendations taking into account well defined long term goals and user satisfaction. Learning design recommender principles will be applied in order to generate more effective recommendations to conduct users. Furthermore, we intend to utilize distance based reasoning to make sense of the space of items and represent user's trajectories and goal in that space. Other data will also apply in order to enhance recommendations such as item features and user-item interaction ratings (preference rating or test results in the case of e-learning).

3.5 Task 5: Design an evaluation framework

Researchers evaluate their recommenders using Information Retrieval approaches (*Precision*, *Recall*, etc), Machine Learning approaches ($RSME$, MAE, etc) and Decision Support System (DSS) approaches (such as customer satisfaction and user loyalty). Although many recommenders are evaluated by IR and ML measures [19], we need to continually measure users interaction with system and DSS evaluation approaches provide more appropriate evaluation for LTRS.

Moreover, in this step, we also plan to design appropriate evaluation measures and methodologies to evaluate the success of the proposal. Due to the fact that evaluation must be performed with live recommendations on real cases (since we need to monitor how users respond to the recommendations), we see this task as a challenging one. We will determine goals to test users and evaluate the success of the methodology in guiding users toward the goals. The evaluation of results will be compared with a control group of users. In particular, we need to:

- Specify the evaluation criteria
- Define evaluation methodology
- Specify online evaluation protocols
- Perform experiments
- Statistically validate results

Furthermore, offline and user study are other methods which are applicable in order to evaluate the result of LTRS. In addition to systematic empirical evaluation of the proposed method, we also intend to demonstrate our idea on one or two real cases. Our plan is to have one e-learning case and one music recommendation case.

4 Conclusion

In this paper we propose long term goal recommender systems (LTRS) that besides satisfying immediate needs of users, conduct users towards a predefined goal. In such a scenario, user guidance would be achieved by generating a sequence of relevant recommendations through time. This strategy is applicable in different domains such as E-learning, movie, music, etc. Generating a strategy for long term goals is of interest in recommending learning resources to learn a concept, and also when a company attempts to convince users to buy certain products.

Several methods and technologies will be utilized to build LTRS. The principles of learning design activity can be useful in order to have more effective recommendations. Another technology which is useful for this purpose is persuasive technology. Persuasive technology concentrates on the psychological aspect of recommendations and explains how recommendations can be represented in order to have more effect on users.

To evaluate LTRS we will require appropriate methods to assess the success of strategic recommendations, since current measures such as *Precision*, and *Recall* are not sufficient. In any case, offline and online evaluation should be complemented with user studies.

5 acknowledgment

This study is financed by the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project UID/EEA/50014/2013.

References

1. Brusilovsky, P. and Vassileva, J. (2003). Course sequencing techniques for large-scale web-based education. *International Journal of Continuing Engineering Education and Life Long Learning*, 13(1-2):75–94.
2. Carchiolo, V., Longheu, A., and Malgeri, M. (2010). Reliable peers and useful resources: Searching for the best personalised learning path in a trust-and recommendation-aware environment. *Information Sciences*, 180(10):1893–1907.
3. Durand, G., Laplante, F., and Kop, R. (2011). A learning design recommendation system based on markov decision processes. In *KDD-2011: 17th ACM SIGKDD conference on knowledge discovery and data mining*.
4. Durand, G., Belacel, N., and LaPlante, F. (2013). Graph theory based model for learning path recommendation. *Information Sciences*, 251:10–21.
5. Fogg, B. J. (2002). Persuasive technology: using computers to change what we think and do. *Ubiquity*, 2002(December):5.
6. Fournier-Viger, P., Faghihi, U., Nkambou, R., and Nguifo, E. M. (2010). Exploiting sequential patterns found in users' solutions and virtual tutor behavior to improve assistance in its. *Journal of Educational Technology & Society*, 13(1):13–24.

7. Garrido, A., Morales, L., and Serina, I. (2012). Using AI planning to enhance e-learning processes. In *ICAPS*.
8. Huang, Y.-M., Chen, J.-N., Huang, T.-C., Jeng, Y.-L., and Kuo, Y.-H. (2008). Standardized course generation process using dynamic fuzzy petri nets. *Expert Systems with Applications*, 34(1):72–86.
9. Idris, N., Yusof, N., and Saad, P. (2009). Adaptive course sequencing for personalization of learning path using neural network. *International Journal of Advances in Soft Computing and Its Applications*, 1(1):49–61.
10. Karampiperis, P. and Sampson, D. (2005). Adaptive learning resources sequencing in educational hypermedia systems. *Educational Technology & Society*, 8(4):128–147.
11. Klašnja-Milićević, A., Vesin, B., Ivanović, M., and Budimac, Z. (2011). E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56(3):885–899.
12. Nabizadeh, A. H., Jorge, A. M., and Leal, J. P. (2015). Long term goal oriented recommender systems. *11th International Conference on Web Information Systems and Technologies (WEBIST)*, pages 552–557.
13. O'Keefe, D. J. (2002). *Persuasion: Theory and research*, volume 2. Sage.
14. Pi-lian, W. T. H. (2005). Web log mining by an improved aprioriall algorithm. *Engineering and Technology, þ*, 4(2005):97–100.
15. Reeves, B. and Nass, C. (1997). The media equation: How people treat computers, television, new media like real people places. *Computers & Mathematics with Applications*, 33(5):128–128.
16. Ullrich, C. and Melis, E. (2009). Pedagogically founded courseware generation based on htn-planning. *Expert Systems with Applications*, 36(5):9319–9332.
17. Vassileva, J. and Deters, R. (1998). Dynamic courseware generation on the www. *British Journal of Educational Technology*, 29(1):5–14.
18. Wiley, D. A. (2003). *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*.
19. Yoo, K.-H., Gretzel, U., and Zanker, M. (2012). *Persuasive Recommender Systems*. Springer.

Metalearning For Pruning and Dynamic Integration In Bagging Ensembles

Fábio Pinto, Carlos Soares and João Mendes-Moreira

INESC TEC/Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, s/n
Porto, Portugal 4200-465
fhpinto@inescporto.pt csoares@fe.up.pt jmoreira@fe.up.pt

Abstract. Ensemble methods have been receiving an increasing amount of attention, especially because of their successful application to problems with high visibility (e.g., the NetFlix prize, Kaggle competitions, etc). An important challenge in ensemble learning (EL) is the management of a set of models in order to ensure accuracy and computational efficiency, particularly with a large number of models in highly dynamic environments. We plan to use metalearning (MtL) to improve the performance of one of the most important EL algorithms: bagging. MtL uses data from past experiments to build models that relate the characteristics of learning problems with the behaviour of algorithms. Our approach consists in using MtL techniques that act at the level of 1) ensemble pruning and 2) ensemble integration to improve the performance of the original bagging algorithm. On the one hand, we present results of a technique that allows to prune bagging ensembles before actually generating the individual models with a performance equal to the original algorithm. On the other hand, we expose how we plan to extend the work done in 1) to include a dynamic approach. Our final goal is to achieve a MtL method that is able to select the most suitable subset of models according to the characteristics of the instance that is being predicted.

Keywords: Metalearning, Ensemble Learning, Bagging, Pruning

1 Introduction

We present an overview of the ongoing research regarding the application of metalearning (MtL) techniques in order to improve the performance of the bagging algorithm, particularly at the level of ensemble pruning and integration.

Bagging is an ensemble learning technique that allows to generate multiple predictive models and aggregate their output to provide a final prediction. Typically, the aggregation function is the mean (if the outcome is a quantitative variable) or the mode (if the outcome is a qualitative variable). The models are built by applying a learning algorithm to bootstrap replicates of the learning set [1].

MtL is the study of principled methods that exploit metaknowledge to obtain efficient models and solutions by adapting machine learning and data mining processes [2]. We plan to use MtL to prune and dynamically integrate bagging ensembles. Our goal is to develop such a method that is able to prune bagging ensembles according to the characteristics of the bootstrap samples and then dynamically integrate the final models according to the characteristics of those models and the test instances.

In this PhD spotlight paper, we present some promising preliminary results regarding the ensemble pruning component of our method and we expose the approach that we plan to follow in our research.

This paper is organized as follows. In Section 2, we present the methodology that we plan to follow in our research line. Section 3 presents preliminary results regarding a MtL pruning technique applied to bagging ensembles of decision trees on 53 classification datasets. Finally, Section 4 concludes the paper.

2 Methodology

The approach is summarized in Figure 1. Our method initializes by extracting data characteristics (or metafeatures) from each bootstrap sample $b_1, ..., b_n$ of the training data. These metafeatures are stored in a meta-dataset together with the relative importance of each bootstrap on a sample of all possible model combinations, $2^n - 1$. A study of the effectiveness of this sample procedure together with an exploratory analysis of the meta-data was published in [3].

Regarding the pruning component of our method, our initial approach was to characterize each bootstrap sample individually (*1st phase*). We published a paper reporting those experiments [4] (which are also discussed in Section 3 of this paper). However, we acknowledge that this type of characterization can dismiss one of the most important concepts of EL: diversity. It is well known in the EL literature that complementary classifiers can improve the accuracy over individual models. One can say that two classifiers are complementary if they make errors in different regions of the input space and therefore their predictions are diverse [5].

We plan now to enrich the characterization of the bootstrap samples by developing metafeatures that compare each bootstrap to the rest of them and, therefore, include the concept of diversity in the design of the metafeatures (*2nd phase*). Hopefully, this characterization would allow to measure the complementarity between the respective models of the bootstrap samples. If we succeed, this should improve the efficiency of the pruning method that we propose.

Next, in the *3rd phase*, we plan to extend our pruning method in such a way that its decision on *to prune or not to prune* a model is not made on a individual level. We want it to take into account the decisions made before (the models that were not pruned) in order to seek the model complementarity that we mentioned before.

Finally, in the *4th phase* of this thesis project, we plan to extend the pruning method to consider a dynamic approach. That is, the the selection and combination of the best subset of model(s) for each test instance.

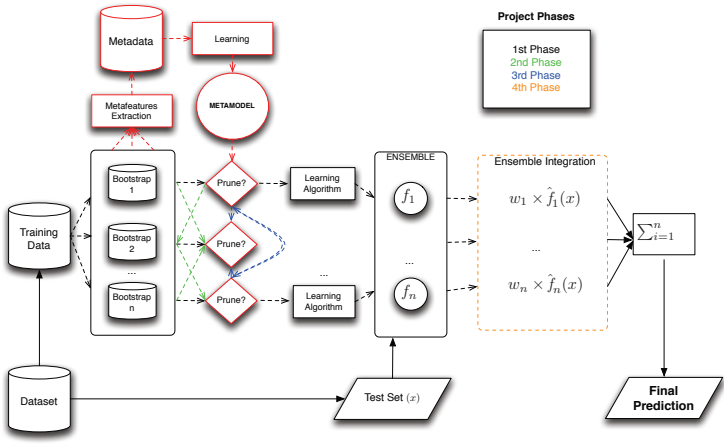


Fig. 1. Project phases. *Phase 1*: pruning and characterization of bootstrap samples is done on individual level. *Phase 2*: pruning is individual but characterization of bootstrap samples is done by comparing several bootstrap samples. *Phase 3*: pruning and characterization of bootstrap samples is done by comparing several bootstrap samples. *Phase 4*: dynamic integration of subset of models.

3 Preliminary Results

In this section we present preliminary results on the pruning method developed using the approach described in Figure 1. We followed an approach as described in the *1st phase* of this thesis project. More details on this work can be found in [4].

We use three different learning algorithms in our MtL framework: Random Forests (*Meta.RF*), M5' (*Meta.M5'*) and Support Vector Machine (*Meta.SVM*). We compare our method with 4 benchmarks: 1) *Metatarget* - in this approach we use the groundtruth of our metatarget to execute the pruning at the base-level. This allows to benchmark how good our method could be if we were able to generate a perfect meta-model; 2) *Bagging* - the same algorithm proposed by Breiman [1], without any sort of pruning; 3) *Margin Distance Minimization* (*MDSQ*) [6] - this algorithm belongs to the family of pruning methods based on

modifying the order in which classifiers are aggregated in a bagging ensemble. The main feature of these kind of methods is to exploit the complementariness of the individual classifiers and find a subset with good performance; 4) *Random pruning* - baseline approach in which the selection of models to be pruned is random. This is repeated 30 times for robust results.

Results are presented in Figure 2. We can see that *Meta.RF* has a performance very similar to bagging and this is achieved with a pruning rate of 75 % before actually generating the final models. We consider these results very promising. However, the CD diagram also shows that the method is not statistically different from MDSQ, the Metatarget and Random. We plan to improve these results following the research plan that we present in Section 2.

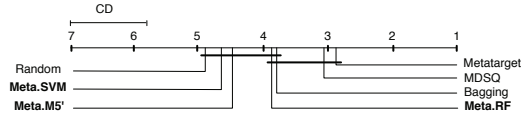


Fig. 2. Critical Difference diagrams ($\alpha = 0.05$) of the performance of the metamodels in comparison with the benchmark pruning methods.

4 Final Remarks and Future Work

This paper describes the ongoing research of a MtL method to prune and dynamically integrate bagging ensembles. A brief overview of the research plan was presented and some preliminary results were analyzed. We plan to follow the research plan in order to achieve the goals that were previously set.

References

1. Breiman, L.: Bagging predictors. Machine learning **24**(2) (1996) 123–140
2. Brazdil, P., Carrier, C.G., Soares, C., Vilalta, R.: Metalearning: applications to data mining. Springer (2008)
3. Pinto, F., Soares, C., Mendes-Moreira, J.: An empirical methodology to analyze the behavior of bagging. In: Advanced Data Mining and Applications. Springer (2014) 199–212
4. Pinto, F., Soares, C., Mendes-Moreira, J.: Pruning bagging ensembles with metalearning. In: Multiple Classifier Systems. Springer (2015) 64–75
5. Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: a survey and categorisation. Information Fusion **6**(1) (2005) 5–20
6. Martinez-Muñoz, G., Hernández-Lobato, D., Suárez, A.: An analysis of ensemble pruning techniques based on ordered aggregation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(2) (2009) 245–259

Multi-sensor data fusion techniques for the identification of activities of daily living using mobile devices

Ivan Miguel Pires[1,2], Nuno M. Garcia[1,3] and Francisco Florez-Revuelta[4]

[1]Instituto de Telecomunicações, University of Beira Interior, Covilhã, Portugal
[2]Altranportugal, Lisbon, Portugal
[3]ECATI, Universidade Lusófona de Humanidades e Tecnologias, Lisbon, Portugal
[4]Faculty of Science, Engineering and Computing, Kingston University, Kingston upon Thames, UK

[1]impires@it.ubi.pt, [2]ngarcia@di.ubi.pt, [3]F.Florez@kingston.ac.uk

Abstract. This paper presents a PhD project related to the use of multi-sensor data fusion techniques, applied to the sensors embedded in mobile devices, as a mean to identify user's daily activities. It introduces some basic concepts, such as the definition of activities of daily living, mobile platforms/sensors, multi-sensor technologies, data fusion, and data imputation. These techniques have already been applied to fuse the data acquired with different sensors, but due to memory constraints, battery life and processing power of these devices, not all the techniques are suited to be used in these environments. This paper explains an overview about the state of the research in this topic, explaining the methodology to create a best effort method to recognize a large number of activities of daily living using a mobile device.

Keywords. Sensors; data fusion; multi-sensor; mobile platforms; activities of daily living

1 Introduction

The identification of Activities of Daily Living (ADL) focuses on the recognition of a well-known set of everyday tasks that people usually learn in early childhood. These activities include feeding, bathing, dressing, grooming, moving without danger, and other simple tasks related to personal care and hygiene. On the context of Ambient Assisted Living (AAL), some individuals need particular assistance, either because the user has some sort of disability, or because the user is elder, or simply because the user needs/wants to monitor and train his/her lifestyle.

The aim of this PhD research consists in the definition of a set of ADLs that may be reliably identified with mobile devices. This includes the activities related to acquire data and recognize a set of tasks and identify which tasks are accurately recognized.

The joint selection of the set of valid sensors and the identifiable set of tasks will then allow the development of a tool that, considering multi-sensor data fusion technologies and context awareness, in coordination with other information available from the user context, such as their agenda and the time of the day, will allow to establish a profile of the tasks that the user performs in a regular activity day.

The accuracy of the identification of ADLs using a mobile device depends on the environment where the data is acquired, the methods used in data processing/imputation/fusion, and the mobile devices used. Several pattern recognition and machine learning techniques have already been used for the identification of ADLs. Besides, data collected can have noise, and statistical methods should be applied to minimize it. Hence, the algorithms for the detection of ADLs can be improved to increase the set of activities that can be accurately detected using mobile devices.

As a result of this PhD a new method to recognize a large set of ADLs with mobile devices will be developed and implemented.

This paper is organized as follows. Section 2 presents a review of the state of the art, focusing in the main concepts of this topic. Section 3 introduces the proposed solution to be developed during this PhD work. Section 4 presents the discussion and conclusion.

2 Related Work

This research topic involves many different areas of research: activities of daily living, multi-sensor, data fusion and data imputation. This section reviews previous works in these areas but constrained to the use of mobile devices.

2.1 Identification of activities of daily living

Activities of daily living (ADL) are activities that require more than just the necessary cognitive and physical abilities but a sense of personal identity and awareness response of others. These activities involve a desire to achieve a degree of physical comfort, self-care, and autonomy, which promotes feelings of independence and personal control [1]. Common activities of daily life are related to personal appearance and hygiene, domestic skills, household management, family and child care, family planning and sexual matters, budgeting and personal administration, conversational and social skills, mobility transfers, and leisure, education, training and work activities [1]. The detection of health problems, using the analysis of ADLs, is carried out by the analysis of the accuracy of the patient when performing these activities. In [2] is shown that detection of ADLs may assess how life's quality of people with dementia is affected. The evaluations of ADLs involve some psychological or medical determinations to understand people's ability to care for themselves on a day-to-day basis [3].

A variety of sensors have been used to recognize ADLs. Accelerometer, door, item, temperature, light, wearable, gravity, ECG, vital sign and RFID sensors, GPS

receivers, microphones, cameras, and other sensors, are used to detect when a person is having/preparing a meal, washing up, bathing, waking up, sleeping, standing, sitting, watching TV, using the phone, doing the chores, cycling, jogging or perform other activities [4-20].

2.2 Mobile Platforms

Mobile devices are used in the vast majority of people's daily activities [21]. These devices are embedded with a large variety of sensors [22], such as GPS receiver, accelerometer sensor, gyroscope sensor, proximity sensor, light sensor, communication sensors, acoustic sensors, digital camera and other over-the-air sensors.

The mobile platforms available in the market in 2014 [23] are Android, iOS, Windows Phone, BlackBerry, Samsung Bada, Samsung Tizen, Symbian, MeeGo, Asha, Firefox OS, and Ubuntu Touch. The two platforms responsible for the largest market share are Android and iOS operating systems [24].

2.3 Multi-Sensor

The use of multiple sensors may increase the reliability of the system. The most important stage in multi-sensor systems is signal classification with pattern recognition or machine learning methods [25].

Multiple sensors can be used in the detection of ADLs or monitor rehabilitation activities. In [26] a system, which combines different sensors, was created for data processing and logging. In [27] a human-aided multi-sensor fusion system was created. It involves the integration of the Probabilistic Argumentation System and the Structural Evidential Argumentation System, which both are variants of the Dempster-Shafer belief function theory. Detection of ADLs are carried out in [28] by using a platform composed of a base station and a number of sensor nodes, recognizing human activity with the minimum body sensor usage through the use of dynamic sensor collaboration. In [29] a wearable multi-sensor ensemble classifier for physical activity pattern recognition was developed, which combines multiple classifiers based on different sensor feature sets to improve the accuracy of physical activity type identification and recognizing 6 different physical activities. In [30] wearable inertial sensors and fiber sensors attached to different human body parts are used to capture kinetic data. Recognition is achieved by combining it neural networks and hidden Markov models.

In [31] a wireless wearable multi-sensor system was created for locomotion mode recognition, with three inertial measurement units (IMUs) and eight force sensors, measuring both kinematic and dynamic signals of human gait, using a linear discriminant analysis (LDA) classifier.

2.4 Data Fusion

Data fusion consists in the integration of data and knowledge from several sources [32]. According to [33, 34], data fusion methods belong to three categories. These are:

- Probabilistic methods (Bayesian analysis of sensor values, Evidence Theory, Robust Statistics, and Recursive Operators);
- Probabilistic approaches (Least square-based estimation methods such as Kalman Filtering, Optimal Theory, Regularization, and Uncertainty Ellipsoids and Bayesian approach with Bayesian network and state-space models, maximum likelihood methods, possibility theory, evidential reasoning and more specifically evidence theory);
- Artificial Intelligence (Intelligent aggregation methods such as Neural Networks, Genetics Algorithms, and Fuzzy Logic).

Multiple techniques related to sensor fusion are presented in [32, 34-36], using several sensors and techniques, such as Kalman filter and their variants, neural networks and other statistical methods.

2.5 Data Imputation

During acquisition time data collection can fail in some instants. These failures may be due to various reasons. Missing data failures can be classified as [37, 38]:

- Missing completely at random (MCAR) happens when missing values are randomly distributed across all observations;
- Missing at random (MAR) is the condition that exists when missing values are randomly distributed within one or more subsamples instead of the whole data set like MCAR;
- Missing not at random (MNAR) is the type of missingness that arises when missing values are not randomly distributed across observations.

However, various methods exist for the estimation of missing values in what is called Data Imputation.

Several methods related to data imputation are presented in [39]. The main methods are K-nearest neighbors and other statistical methods [39-41]. For the recognition of ADLs, some methods can be applied in pattern recognition and health state detection [39, 42, 43]. During this PhD other statistical algorithms will also be studied.

3 Proposed Solution

The proposed solution to solve the problem presented in section 2 consists in the design and development of different methods/algorithms for the automatic identification of a suitable set of ADLs using sensors embedded in off-the-shelf mobile devices. Identification will be supported with other contextual data, e.g. the agenda of the user.

The solution proposed in this PhD work for the identification of ADLs is composed of different modules/stages (figure 1):

- Sensors data acquisition;
- Sensors data processing;
- Sensors data fusion;
- Sensors data imputation;

- Data Mining/Pattern Recognition/Machine Learning techniques.

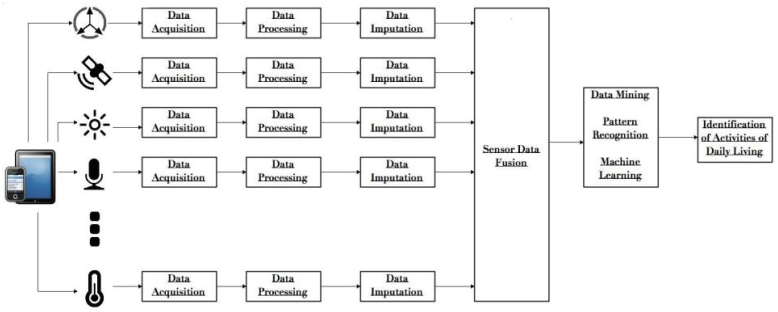


Fig. 1. Process for the identification of Activities of Daily Living using mobile sensors

The first stage 1 includes the research to determine the sensors that the system should use in order to identify accurately a large set of ADLs. Sensors that are available in mobile devices differ depending on the mobile platform used and hardware, but they are commonly those mentioned in Section 2.1.

Data acquisition process, which it is the second stage of the proposed solution, should be adapted to the environment and positioning, related to the user's body, of the mobile device. Data collected is used in the Data Processing stage. This stage must use methods to minimize the effects of environmental noise in the data collected by all the available sensors and convert these data to homogeneous units. In the PhD thesis, a new method to commute the algorithm with the number of sensors available should be created.

Due to the different number of sensors available in mobile devices, a process is needed to analyze which are the sensors available in the used mobile device and determine the maximum number of sensors that should be used during data acquisition and data processing in order to increase the accuracy of the identification.

After data acquisition and data processing stages, data fusion deals with merging appropriately the data coming from all those sensors. Although this can be done with different methods, Kalman filter and their variants are the most commonly used with low processing techniques or with server side processing. The capacities of the mobile devices are the most important criteria for choosing one method for data fusion techniques. During this PhD project, a new method to carry out efficiently sensor data fusion with a mobile device will be developed. The main objective is that this data fusion stage occurs in real-time without large local processing, because the mobile devices have low processing capacities and memory.

Sometimes data acquisition can fail due to the unavailability of sensors, unknown errors occurred in real-time collection. This can affect the global performance of the system to recognize ADLs. Hence, the existence of a module for data imputation is very important. Data imputation techniques can be applied before or after sensor data fusion using different statistical methods. Missing data can be generated using several

algorithms (*e.g.* K-nearest neighbor (KNN) schemes, likelihood-based schemes, Bayesian-based schemes and multiple imputation (MI) schemes) or other methods, such as MLE in multivariate normal data, GMM estimation, Predictive mean matching (PMM), Multiple imputation via MCMC and Multivariate imputation by Chained Equations. In this PhD project, the reliability of the existent methods will be verified and a new method for data imputation will be developed (if needed).

Next, pattern recognition or machine learning methods will be created for the identification of activities of daily living. They must be validated with a gold standard (*i.e.* inquiring a user about the activities performed or watching the user).

Finally, all these algorithms/methods will be implemented as a mobile application. The mobile application should be developed for a major mobile operating system in order to automatically detect the activities of daily living of a subject, with a comfortable degree of accuracy in different environments.

4 Discussion and Conclusion

Until now this PhD project has reviewed the state of the art of the different topics related to the identification of ADLs using a mobile. These are:

- Activities of daily living;
- Multi-sensor techniques;
- Sensors data fusion technologies;
- Sensors data imputation techniques;
- Mobile platforms;
- Context aware applications.

Currently, mobile devices, such as smartphones, tablets, among others are widely used. Mobile devices incorporate various sensors, depending on the platform used, which allow capturing a variety of data.

These sensors are able to detect different parameters about people's health, activities of daily living and other purposes. Sensors available in mobile devices are quite diverse, such as accelerometry sensors (*e.g.* gyroscope, accelerometer and magnetometer), acoustic sensors (*e.g.* microphone), location sensors (*e.g.* GPS receiver), digital camera and other over-the-air sensors (*e.g.* heart rate monitors).

This PhD project will use those sensors to identify activities of daily living. The study about the identification of activities of daily living is very complex and it is divided in some subtopics, such as multi-sensor, data fusion, mobile platforms, identification of activities of daily living and data imputation.

At this stage, a state of the art has been finalized in order to obtain the global knowledge to design and develop new methods for each one of those stages. Finally, all these methods will be embedded in a mobile application that will allow the validation with users under real conditions.

5 Acknowledgements

This work was supported by FCT project **UID/EEA/50008/2013** (*Este trabalho foi suportado pelo projecto FCT UID/EEA/50008/2013*).

The authors would also like to acknowledge the contribution of the COST Action IC1303 – AAPELE – Architectures, Algorithms and Protocols for Enhanced Living Environments.

6 References

[1] S. Morgan, "Activities of daily living," in *Community Mental Health*, ed: Springer Berlin Heidelberg, 1993, pp. 141-158.

[2] C. K. Andersen, K. U. Wittrup-Jensen, A. Lolk, K. Andersen, and P. Kragh-Sorensen, "Ability to perform activities of daily living is the main factor affecting quality of life in patients with dementia," *Health Qual Life Outcomes,* vol. 2, p. 52, 2004. doi: 10.1186/1477-7525-2-52

[3] T. R. Howe, J. S. Trotter, A. S. Davis, J. W. Schofield, L. Allen, M. Millians, *et al.*, "Activities of Daily Living," in *Encyclopedia of Child Behavior and Development*, ed: Springer US, 2011, pp. 28-29.

[4] B. Chikhaoui, S. Wang, and H. Pigot, "A Frequent Pattern Mining Approach for ADLs Recognition in Smart Environments," in *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, Biopolis, 2011, pp. 248-255.

[5] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, "Activity Recognition Using Wearable Sensors for Elder Care," in *Future Generation Communication and Networking, 2008. FGCN '08. Second International Conference on*, Hainan Island, 2008, pp. 302-305.

[6] S. Szewcyzk, K. Dwan, B. Minor, B. Swedlove, and D. Cook, "Annotating smart environment sensor data for activity learning," *Technol Health Care,* vol. 17, pp. 161-9, 2009. doi: 10.3233/THC-2009-0546

[7] S. Chernbumroong, A. S. Atkins, and H. Yu, "Activity classification using a single wrist-worn accelerometer," in *Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on*, 2011, pp. 1-6.

[8] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Trans Inf Technol Biomed,* vol. 12, pp. 20-6, Jan 2008. doi: 10.1109/TITB.2007.899496

[9] T. Maekawa, Y. Kishino, Y. Sakurai, and T. Suyama, "Activity recognition with hand-worn magnetic sensors," *Personal and Ubiquitous Computing,* vol. 17, pp. 1085-1094, 2012. doi: 10.1007/s00779-012-0556-8

[10] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity Recognition and Monitoring Using Multiple Sensors on Different Body

Positions," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, Cambridge, 2006, pp. 113-116.

[11] K. Kuspa and T. Pratkanis, "Classification of Mobile Device Accelerometer Data for Unique Activity Identification," 2013. Available: http://cs229.stanford.edu/proj2013/PratkanisKuspa-ClassificationOfMobileDeviceAccelerometerDataforUniqueActivityIdentification.pdf

[12] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter,* vol. 12, p. 74, 2011. doi: 10.1145/1964897.1964918

[13] Z. Fitz-Walter and D. Tjondronegoro, "Simple classification of walking activities using commodity smart phones," in *OZCHI '09 Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, New York, NY, USA, 2009, p. 409.

[14] P. Siirtola and J. Röning, "Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data," *International Journal of Interactive Multimedia and Artificial Intelligence,* vol. 1, p. 38, 2012. doi: 10.9781/ijimai.2012.155

[15] I. Kouris and D. Koutsouris, "A comparative study of pattern recognition classifiers to predict physical activities using smartphones and wearable body sensors," *Technol Health Care,* vol. 20, pp. 263-75, 2012. doi: 10.3233/THC-2012-0674

[16] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing,* vol. 14, pp. 645-662, 2010. doi: 10.1007/s00779-010-0293-9

[17] S. Das, L. Green, B. Perez, and M. M. Murphy, "Detecting User Activities using the Accelerometer on Android Smartphones," 2010. Available: https://www.truststc.org/education/reu/10/Papers/DasGreenPerezMurphy_Paper.pdf

[18] D. T. G. Huynh, "Human Activity Recognition with Wearable Sensors," Doktor-Ingenieur (Dr.-Ing.), Fachbereich Informatik, Technische Universitat Darmstadt, Darmstadt, 2008.

[19] S. Zhang, M. H. Ang, Jr., W. Xiao, and C. K. Tham, "Detection of activities by wireless sensors for daily life surveillance: eating and drinking," *Sensors (Basel),* vol. 9, pp. 1499-517, 2009. doi: 10.3390/s90301499

[20] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook, "Simple and Complex Activity Recognition through Smart Phones," in *Intelligent Environments (IE), 2012 8th International Conference on*, Guanajuato, Mexico, 2012, pp. 214-221.

[21] J. Heggestuen. (2013, January 12th). *One In Every 5 People In The World Own A Smartphone, One In Every 17 Own A Tablet [CHART]*. Available: http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10

[22] L. H. A. Salazar, T. Lacerda, J. V. Nunes, and C. Gresse von Wangenheim, "A Systematic Literature Review on Usability Heuristics for Mobile Phones," *International Journal of Mobile Human Computer Interaction,* vol. 5, pp. 50-61, 2013. doi: 10.4018/jmhci.2013040103

[23] R. Chang. (2014, May 23rd). *Mobile Operating Systems in 2014 - Tuts+ Code Article*. Available: http://code.tutsplus.com/articles/mobile-operating-systems-in-2014--cms-19845

[24] Z. Whittaker. (2014, May 23rd). *Android, iOS score 96 percent of smartphone share in Q4 rankings*. Available: http://www.zdnet.com/android-ios-score-96-percent-of-smartphone-share-in-q4-rankings-7000026257/

[25] R. Xu and L. He, "GACEM: Genetic Algorithm Based Classifier Ensemble in a Multi-sensor System," *Sensors,* vol. 8, pp. 6203-6224, 2008. doi: 10.3390/s8106203

[26] R. Bin Ambar, P. Hazwaj Bin Mhd, A. Abdul Malik Bin Mohd, M. S. Bin Ahmad, and M. M. Bin Abdul Jamil, "Multi-sensor arm rehabilitation monitoring device," in *Biomedical Engineering (ICoBE), 2012 International Conference on*, Penang, 2012, pp. 424-429.

[27] M. Chan, E. H. Ruspini, J. Lowrance, J. Yang, J. Murdock, and E. Yeh, "Human-aided multi-sensor fusion," presented at the Information Fusion, 2005 8th International Conference on 2005.

[28] L. Gao, A. K. Bourke, and J. Nelson, "An efficient sensing approach using dynamic multi-sensor collaboration for activity recognition," presented at the Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on, Barcelona, 2011.

[29] L. Mo, S. Liu, R. X. Gao, and P. S. Freedson, "Multi-Sensor Ensemble Classifier for Activity Recognition," *Journal of Software Engineering and Applications,* vol. 05, pp. 113-116, 2012. doi: 10.4236/jsea.2012.512B022

[30] Z. Chun and S. Weihua, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," presented at the Robotics and Automation, 2009. ICRA '09. IEEE International Conference on, Kobe, 2009.

[31] E. Zheng, B. Chen, X. Wang, Y. Huang, and Q. Wang, "On the Design of a Wearable Multi-sensor System for Recognizing Motion Modes and Sit-to-stand Transition," *International Journal of Advanced Robotic Systems,* p. 1, 2014. doi: 10.5772/57788

[32] F. Castanedo, "A review of data fusion techniques," *ScientificWorldJournal,* vol. 2013, p. 704504, 2013. doi: 10.1155/2013/704504

[33] M. A. A. Akhoundi and E. Valavi, "Multi-Sensor Fuzzy Data Fusion Using Sensors with Different Characteristics," *arXiv preprint arXiv:1010.6096,* 2010

[34] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion,* vol. 14, pp. 28-44, 2013. doi: 10.1016/j.inffus.2011.08.001

[35] J. Esteban, A. Starr, R. Willetts, P. Hannah, and P. Bryanston-Cross, "A Review of data fusion models and architectures: towards engineering

guidelines," *Neural Computing and Applications,* vol. 14, pp. 273-281, 2005. doi: 10.1007/s00521-004-0463-7

[36] MSB, "Sensor data fusion: state of the art survey." Available: https://www.msb.se/Upload/Om MSB/Forskning/Kunskapsoversikt/Sensor_data_fusion_survey.pdf

[37] P. Vateekul and K. Sarinnapakorn, "Tree-Based Approach to Missing Data Imputation," presented at the Data Mining Workshops, 2009. ICDMW '09. IEEE International Conference on Miami, FL, 2009.

[38] A. D'Ambrosio, M. Aria, and R. Siciliano, "Accurate Tree-based Missing Data Imputation and Data Fusion within the Statistical Learning Paradigm," *Journal of Classification,* vol. 29, pp. 227-258, 2012. doi: 10.1007/s00357-012-9108-1

[39] P. A. Patrician, "Multiple imputation for missing data," *Res Nurs Health,* vol. 25, pp. 76-84, Feb 2002. doi: 10.1002/nur.10015

[40] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, and M. Verleysen, "K nearest neighbours with mutual information for simultaneous classification and missing data imputation," *Neurocomputing,* vol. 72, pp. 1483-1493, 2009. doi: 10.1016/j.neucom.2008.11.026

[41] C. Liu, "Missing Data Imputation Using the Multivariate t Distribution," *Journal of Multivariate Analysis,* vol. 53, pp. 139-158, 1995. doi: 10.1006/jmva.1995.1029

[42] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Computing and Applications,* vol. 19, pp. 263-282, 2009. doi: 10.1007/s00521-009-0295-6

[43] L. Ting Hsiang, "Missing Data Imputation in Quality-of-Life Assessment: Imputation for WHOQOL-BREF," *PharmacoEconomics,* vol. 24, p. 917, 2006

Characterization of Learning Instances for Evolutionary Meta-Learning

William Raynaut[1], Chantal Soule-Dupuy[1], Nathalie Valles-Parlangeau[1], Cedric Dray[2], and Philippe Valet[2]

[1] IRIT UMR 5505, UT1, UT3
Universite de Toulouse, France
firstname.lastname@irit.fr
[2] Institut National de la Sante et de la Recherche Medicale (INSERM), U1048
Universite de Toulouse, France
firstname.lastname@inserm.fr

Abstract. Machine learning has proven to be a powerful tool in diverse fields, and is getting more and more widely used by non-experts. One of the foremost difficulties they encounter lies in the choice and calibration of the machine learning algorithm to use. Our objective is thus to provide assistance in the matter, using a meta-learning approach based on an evolutionary heuristic. We expand here previous work presenting the intended workflow of a modeling assistant by describing the characterization of learning instances we intend to use.

Keywords: Meta-Learning, Modeling, Prediction, Evolutionary Heuristics, Algorithm selection

1 Motivation

Over the last decades was produced an important variety of techniques and algorithms labeled as machine learning. But the performance of such techniques can vary a lot from a dataset to another, and the "no free lunch" theorems [21] showed that no algorithm could outperform all others on every possible problem. This led to many studies of algorithm's inner bias adequateness to diverse learning problem, such as [1] and [4] who used rule-generation machine learning techniques on the problem, describing the conditions under which the significant performance difference between algorithms holds. These applications of machine learning to the study of itself bore great significance over how this Meta-Learning problem would be addressed. Despite promising applications of such approaches over a limited range of learning tasks, like pairwise algorithm comparison [6], or recursion of adaptive learners [20], the Meta-Learning problem still carries many open perspectives. Another approach would be to address directly the question : " *Which learning algorithm will perform best on a given learning problem ?*", without having to comply to the limitation of the classic machine learning techniques employed at the meta-level.

Our own perspective view on the matter is that the meta-knowledge can be viewed as a population of meta-level learning instances (or meta-instances), each describing the evaluated application of a learning task to a given dataset, and that a good solution to a given meta-learning task can be obtained via the evolutionary exploration of this population. Such approach is giving interesting results among other classes of problems such as Boolean satisfiability (SAT) [22] or Instance selection [12], but, to our knowledge, has not yet been explored regarding Meta-Learning.

Our objective is to provide modeling assistance through an evolutionary algorithm-selection approach, which intended workflow is illustrated by figure 1 and was presented more thoroughly in [15].

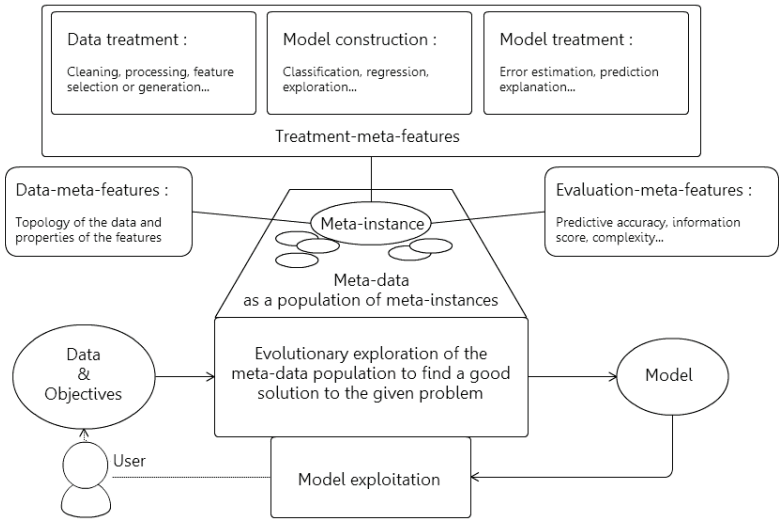


Fig. 1. Modeling assistant

One of the foremost issues we must address in order to complete our framework, and the main topic of this paper, will be the characterisation of the meta-instances. This problem can be viewed as an extended form of the dataset characterization problem faced by most meta-learning approaches, which consists in the definition of a subset of dataset properties (meta-level features of the dataset) that should allow a fine grain characterisation of datasets, while still complying to the requirements of the meta-level learner employed. It is typically solved through some kind of meta-level features selection [7], but to fit most learners requirements, dataset properties have to be aggregated into fixed-length fea-

ture vectors, which results into a important loss in information as stated in [6]. We intend to overcome this issue through the use of an evolutionary heuristic, whose fitness would rely on dissimilarity between meta-instances. Such approach would indeed allow the use of all available information to characterize the meta-instances. Relating in that way to the "anti-essentialist" representations such as discussed in [3], we believe that limitations in the representations of datasets are among the main obstacles to well performing algorithm selection, and are focusing our efforts toward the definition of such representation.

2 Characterising Learning Instances

We will here address the definition of the features that will describe our meta-instances, hence called meta-feature. Those sets of meta-features should be large enough to characterize well any modeling task, but a balance must be found to avoid the abundance of indecisive features and limit computational complexity. Furthermore, in order to discriminate between meta-features or meta-instances according to the user's need, the comparison of meta-features of a particular meta-instance - or of a given meta-feature over several meta-instances - should be possible and make sense.

As a meta-instance describes the evaluated application of a learning task to a given dataset, we can intuitively split those meta-features along three dimensions. First, meta-features describing the data (Fig.1 Data-meta-features), then, meta-features describing the applied treatments (Fig.1 Treatment-meta-features), and finally, meta-features evaluating the resulting model (Fig.1 Evaluation-meta-features).

2.1 Data Meta-features

The dataset characterization problem has been addressed along two main directions :

- In the first one, the dataset is described through a set of statistical or information theoretic measures. This approach, notably appearing in the STATLOG project [10], and in most studies afterwards [8, 20, 12], allows the use of many expressive measures, but its performance depends heavily on the adequateness of bias between the meta-level learner and the chosen measures. Experiments have been done with meta-level features selection [19] in order to understand the importance of different measures, but the elicited optimal sets of meta-feature to perform algorithm selection over two different pools of algorithms can be very different, revealing no significant tendencies among the measures themselves. This led [20] to the intuition that adapting the meta-learning process to specific tasks is in fact a meta-meta-learning problem, and so on, requiring an infinite recursion of adaptive learners to be properly solved.

- The second direction of approach to dataset characterization focuses, not on computed properties of the dataset, but on the performance of simple learners over the dataset. It was introduced as landmarking in [14], where the accuracies of a set of very simple learners are used as meta-features to feed a more complex meta-level learner. There again, the performance of the method relies heavily on the adequate choice of both the base and meta-level learner, with no absolute best combination. Further development introduced more complex measures than predictive accuracy over the models generated by the simple learners. For instance, [11] claims that using as meta-features different structural properties of a decision tree induced over the dataset by simple decision-tree learners can also result in well performing algorithm selection. [13] experiments with those approaches to algorithm selection, showing that all can result in good performance, but that no overall dominance between those methods or over the approaches relying on statistical measures can be found.

The dataset characterization problem has thus already received quite some attention in previous meta-learning studies, but, as stated before, the aggregation of meta-features into fixed-length vectors processable through the meta-level learner were source of an important information loss, even though it was partially limited in [8] with the use of histograms describing the distribution of meta-feature values. However, the paradigm shift between literal meta-learning and our approach will shift the issue to another : we are free to use varying-length meta-feature vectors, but have to design a sound way to compare them. This mostly comes as an issue when comparing meta-features computed over individual features of the dataset, as illustrated in the following example.

Example *We consider two datasets, A and B depicted in Fig.2. A describes 12 features of 100 individuals, and B, 10 features of 200 individuals. Let us say we want to compare the results of a set of 5 statistical or information theoretic measures over each individual feature, like mean, variance, standard deviation, entropy, and kurtosis (as illustrated over the second feature of A in Fig.2). The complete information we want to compare is then a 60-value vector for A, and a 50-values vector for B.*

Our stance on the matter is to compare those features by most similar pairs, while comparing A's two extra features with empty features (features with no value at all). The assumption taken here is that a feature with absolutely no value is equivalent to no feature at all. To get back to our example, we end up comparing the 5 measures taken on the two closest (according to these very measures) features in A and B, then of the second closest, and so on, to finish on comparing the measures taken over the two extra features of A with measures taken over an artificial empty feature. These different comparisons sum up to an accurate description of how different A and B are, according to our set of measures. These pairwise comparisons would allow to ignore the presentation order of the features (which holds no meaningful information), focusing on the actual topology of the dataset.

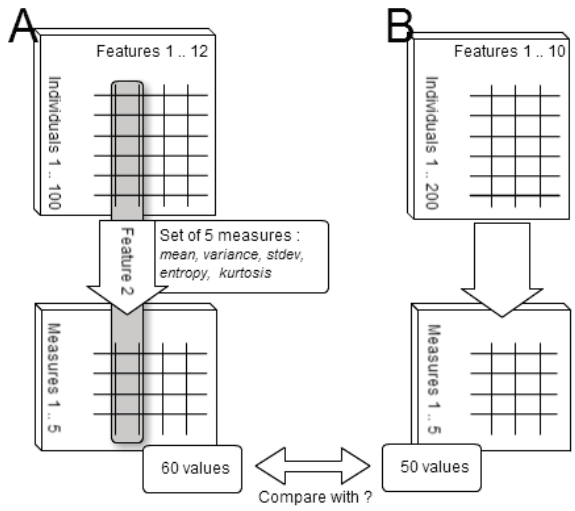


Fig. 2. Measures over individual features

Assuming that a very expressive comparison will result in a better performing fitness, this only emphasizes the need for an extensive set of meta-features. We intend to use most of the classic statistical and information theoretic measures, from the number of instances to features entropy, considering also measures of feature correlation. As the various landmarking approaches showed interesting results, we additionally consider using such measures as meta-features, but further studies might be required to limit overlapping information between the two kinds of measures.

2.2 Evaluation and Treatment Meta-features

The meta-features describing the evaluation of the resulting model should consider a wide range of criteria and allow some flexibility in its comparison to the user's need. Among many usual criteria, we are giving a particular attention to meaningful information-based criteria such as described in [9]. We also wish to investigate the definition of some explainability criteria following [17] prediction explanations, as the ability of the model to explain its predictions has been shown to be a very important factor in allowing non-experts to understand and accept them [18].

The meta-features describing the modeling treatments should consider all potential treatments producing a model of the given dataset. The characteri-

zation of treatments has notably been addressed by the algorithm profiles presented in [5], where sets of algorithm properties are learned from modeling tasks on arbitrary chosen datasets. We intend to describe a modeling algorithm not from a-priori learned properties, but from aggregated properties of the meta-instances of our population presenting the use of this particular algorithm. For instance, the current *predictive accuracy* property of a given algorithm could be defined as the mean of the *predictive accuracy* evaluation-meta-feature among the meta-instances in our current base featuring that particular algorithm. We also consider relative aggregations, such as rank over known algorithms, as no absolute value is required for comparison.

3 Conclusion and perspectives

The set of all meta-features presented above should allow fine grain description of evaluated modeling experiments, and will thus define the structure of the meta-instances over which the evolutionary heuristic will be applied. In other terms, those meta-features will be the genome of the meta-instances, along which evolution will take place, to find a modeling treatment answering the user's need.

However, in order to complete and thus evaluate this framework, several important tasks are yet to be addressed. First, a representation of the user's modeling need that would allow its automatic or semi-automatic elicitation will be required. Indeed, as the target user is a non-expert, he should be walked through the definition of his modeling need, that will define the goal of the evolution. Also, such representation could allow to lessen the computational complexity of the heuristic, by considering only instances that could answer the user's need.

Then, meta-instances comparison metrics shall be formalized in order to define the evolutionary fitness as a similarity with the evolution goal that was elicited from the user's need.

Finally two of the important challenges to address will be the definition and calibration of the evolutionary heuristic employed, and the creation of predatory mechanisms limiting the population of meta-instances. We intend to use the framework of genetic algorithms [2] and memetic algorithms [16], which present desirable properties such as unconstrained individuals and native parallelism, the later being required to deal with the important computational complexity of the intended workflow.

References

1. Aha D.W., Generalising from case studies: a case study, Proceedings Ninth International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA (1992), pp. 110, 1992
2. Jaume Bacardit and Xavier Llor. Large scale data mining using genetics-based machine learning. In Proceedings of the 15th annual conference companion on Genetic and evolutionary computation (GECCO '13 Companion), 2013
3. RPW Duin, The Dissimilarity Representation for finding Universals from Particulars by an anti-essentialist Approach, Pattern Recognition Letters (2015).
4. Gama J and Brazdil P, Proceedings of the seventh Portuguese Conference on Artificial Intelligence, Characterization of Classification Algorithms, 189-200, 1995
5. M. Hilario and A. Kalousis. Building algorithm profiles for prior model selection in knowledge discovery systems. In Proceedings of the IEEE SMC'99, International Conference on Systems, Man and Cybernetics. IEEE press, October 1999.
6. Kalousis A., Hilario M. Model Selection via Meta-learning : a Comparative Study. In Proceedings of the 12th IEEE International Conference on Tools with AI, Vancouver, November 2000
7. Alexandros Kalousis, Melanie Hilario, Feature Selection for Meta-learning, Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science Volume 2035, 2001, pp 222-233, 2001
8. Kalousis, A. Algorithm Selection via Meta-Learning. PHD Thesis, Thesis Number : 3337. University of Geneve, Department of Computer Science, 2002
9. Kononenko I, Bratko I, Information-Based Evaluation Criterion for Classifier's Performance, Machine Learning, January 1991, Volume 6, Issue 1, pp 67-80, 1991
10. D. Michie, D.J. Spiegelhalter, C.C. Taylor. Machine Learning, Neural and Statistical Classification. Ellis Horwood Series in Artificial Intelligence, 1994
11. Yonghong Peng , Peter A Flach , Pavel Brazdil , Carlos Soares, Decision Tree-Based Data Characterization for Meta- Learning, 2002
12. Raul Perez, Antonio Gonzalez, Enrique Leyva, "A Set of Complexity Measures Designed for Applying Meta-Learning to Instance Selection," IEEE Transactions on Knowledge and Data Engineering, 2015
13. Johannes Furnkranz, Johann Petrak, Extended data characteristics, A MetaLearning Assistant for Providing User Support in Machine Learning and Data Mining, METAL, 2002
14. Bernhard Pfahringer, Hilan Bensusan, Christophe Giraud-Carrier, Meta-learning by landmarking various learning algorithms. Proceedings of the Seventeenth International Conference on Machine Learning, ICML'2000, pp. 743750. June 2000
15. William Raynaut, Chantal Soule-Dupuy, Nathalie Valles-Parlangeau, Cedric Dray and Philippe Valet. Addressing the Meta-Learning problem with Metaheuristics (Extended Abstract), Metaheuristics International Conference, 2015

16. Smith J, Coevolving memetic algorithms: A review and progress report, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 6–17, 2007
17. Erik Strumbelj, Igor Kononenko, Towards a Model Independent Method for Explaining Classification for Individual Instances, Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science Volume 5182, pp 273-282, 2008
18. Erik Strumbelj, Zoran Bosni, Igor Kononenko, Branko Zakotnik, Cvetka Grai Kuhar, Explanation and reliability of prediction models: the case of breast cancer recurrence, Knowledge and Information Systems, 2010, Volume 24, Number 2, Page 305
19. L. Todorovski, P. Brazdil, and C. Soares. Report on the experiments with feature selection in meta-level learning. In P. Brazdil and A. Jorge, editors, Proceedings of the Data Mining, Decision Support, Meta-Learning and ILP Workshop at PKDD2000, pages 27-39, 2000
20. Ricardo Vilalta, Youssef Drissi, A Perspective View and Survey of Meta-Learning, Artificial Intelligence Review 2002, Volume 18, Issue 2, pp 77-95, 2002
21. Wolpert David, The Lack of A Priori Distinctions between Learning Algorithms, Neural Computation, pp. 1341-1390, 1996
22. J. Shen H. Hoos-K. Leyton-Brown L. Xu, F. Hutter. Satzilla2012: Improved algorithm selection based on cost-sensitive classification models. SAT Challenge 2012.

An Incremental Algorithm for Repairing Training Sets with Missing Values

Bas van Stein and Wojtek Kowalczyk

Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1, Leiden, The Netherlands
{b.van.stein, w.j.kowalczyk}@liacs.leidenuniv.nl

Abstract. Real-life datasets that occur in domains such as industrial process control, medical diagnosis, marketing, risk management, often contain missing values. This poses a challenge for many classification and regression algorithms which require complete training sets. In this paper we present a new approach for "repairing" such incomplete datasets by constructing a sequence of regression models that iteratively replace all missing values. Additionally, our approach uses the target attribute to estimate the values of missing data. The accuracy of our method, Incremental Attribute Regression Imputation, IARI, is compared with the accuracy of several popular and state of the art imputation methods, by applying them to five publicly available benchmark datasets. The results demonstrate the superiority of our approach.

Keywords: Missing data, Imputation, Regression, Classification, Random Forest

1 Introduction

In industrial processes and many other real-world applications, data is collected to gain insight into the process and to make important decisions. Understanding and making predictions for these processes are vital for their optimization. Missing values in the collected data cause additional problems in building predictive models and applying them to fresh data. Unfortunately, missing values are very common and occur in many processes, for example, sensors that collect data from a production lines may fail; a physician that examines a patient might skip some tests; questionnaires used in market surveys often contain unanswered questions, etc. This problem leads to the following questions:

1. How to build high quality models for classification and regression, when some values in the training set are missing?
2. How to apply trained models to records with missing values?

In this paper we address only the first question, leaving the answers to the second one for further research.

There are several methods developed for tackling this problem, see e.g., [4, 5, 11, 14, 16]. The most common method, *imputation*, reconstructs the missing values with help of various estimates such as means, medians, or simple regression models which predict the missing values. In this paper we present a more sophisticated approach, Incremental Attribute Regression Imputation, IARI, which prioritizes all attributes with missing values and then iteratively "repairs" each of them, one by one, using values of all attributes that have no missing values or are already repaired, as predictors. Additionally, the target variable is also used as a predictor in the repair process. Repairing an attribute is achieved by constructing a regression model and applying it for estimation of missing values. We use here the Random Forest algorithm, [3], [6], due to its accuracy, robustness, and versatility: it can be used to model both numerical and categorical variables. Obviously, after repairing all attributes with missing values a final model for the original target variable is trained on the repaired training set.

We tested our algorithm on five datasets: *Digits*,*Page Blocks*,*Concrete*, and *CoverType* from the UCI Machine Learning Repository, [2], and *Housing 16H* from mldata.org [1], first removing some values at random, then reconstructing them with help of IARI and several common imputation algorithms, and finally comparing the accuracy of regression or classification models trained on reconstructed datasets. The results demonstrate that in most cases, no matter how many attributes were spoiled and by how much, the IARI outperformed other imputation methods both in terms of the accuracy of the final models and the accuracy of imputation. On the other hand, the IARI algorithm is computationally very demanding–it builds as many Random Forests as the number of attributes that should be repaired. Fortunately, due to the parallel nature of the Random Forest algorithm, the runtime of the IARI algorithm can be dramatically reduced by running it on a system with multiple cores or CPUs.

The paper is organized as follows. After introducing various types of missing data and providing an overview of the relevant research on imputation methods we will present the IARI algorithm. Next, we describe in more detail an experimental framework and results of our experiments. Finally, we draw some conclusions and make recommendations for further research.

1.1 Missing Data Types

There are three categories of missing data [13, 11, 10, 6, 8]: *Missing Completely at Random* (MCAR), *Missing at Random* (MAR), and *Missing Not at Random* (MNAR). In many cases, data is MNAR, meaning that the probability that a value of a variable is missing somehow depends on the actual (observed or not) values of this or other variables. A value of a variable is MAR if the probability of being missing does not depend on the (unobserved) value of this variable. And a value of a variable is MCAR if the probability of being missing does not depend on (observed or unobserved) values of this or other variables. In real world scenarios one often cannot determine if the missing data is MCAR, MAR or MNAR because the mechanism behind missingness is not known. In such

situations domain expertise is of vital importance and it can guide the choice of a strategy for handling missing values.

2 Relevant Research

There are many ways of dealing with missing data when building a regression or classification model. Some of the most popular methods are:

***Complete Case Analysis* (CCA):** This method simply ignores all records that have missing values and selects only records with no missing values [7, 5]. When the percentage of complete records is relatively high and the data is missing at random or completely at random, this method does not affect model accuracy. However, if the amount of missing data is large the prediction accuracy will be low (not enough complete cases) and when the data is missing not at random then this method generates bias.

***Missing Indicator Variable* (MIV):** This method uses a dummy variable as an indicator for missing values [7]. For every variable that might be missing, a dummy variable is introduced, where the value of this dummy variable is 1 when the input variable is missing and 0 when the input variable is not missing. While this method is more efficient than the Complete Case Analysis, it can also create bias in the final model.

***Predictive Value Imputation* (PVI):** PVI replaces missing values by some estimates of their values [9]. In many cases the unconditional mean is used (the mean value of all non-missing values of the attribute) or a conditional mean (the mean of a specific group of records where the record with a missing value belongs to). The problem with this method is that the predictive values are always derived from the complete cases and that might introduce some bias. However, some additional mechanisms can be added to PVI which lower this bias. For example, PVI might use the conditional mean over the K nearest neighbors of a record with a missing value, and then the bias can be limited by first imputing the dataset with unconditional mean and then using the K nearest neighbors on the completed dataset to predict the values of the originally missing data. By counting the number of missing data in the neighbors, one can create a weighted average that incorporates the uncertainty of the measurements. There are several other methods to do single-value predictive imputation like *hot-deck imputation*, *cold-deck imputation* and *last observation carried forward*, where the dataset is sorted on specific variables and when a missing value is encountered, the value is replaced by the value of its predecessor.

***Regression Imputation* (RI):** Regression Imputation [9] is a PVI variant where we use regression models (Support Vector Machines, Random Forests, etc.) to estimate the imputed value. One way is to build the models to estimate the missing values using the complete cases. However, it is usually better to also incorporate the non-complete cases by first imputing the missing values with a more simple imputation method (like the unconditional

mean). In the first case (using only complete cases), there might be too few complete cases to generate good models, in the latter case there is a danger of bias by training the model with imputed (wrong) data.

***Multiple Imputation* (MI):** This is a general imputation framework by Rubin et al. [4, 13–15]. The idea is to generate multiple versions of imputed (completed) datasets, which result in multiple models. Each model is then combined into a final predictor. The framework uses a single value imputation algorithm of choice and a random component that represents the uncertainty of the imputation. By creating multiple imputed datasets, the distribution of the imputed values will reflect the distribution of the already known values and therefore reduce bias. This method allows any non-deterministic imputation algorithm to be used. After imputing the dataset several times, creating several copies, a model is being built for each complete dataset. The results of each model are combined using Rubin's Rules [4]. The combined result leads to less biased and more accurate predictions. One of the major advantages of MI is that it can be used with almost any imputation algorithm. Because of this, we do not add MI in our comparison because each of the imputation algorithms can be wrapped with Multiple Imputation.

Most of the above methods can also be used for handling missing data at prediction time. The CCA method is here an obvious exception, but imputation or using dummy variables are valid ways to deal with missing values at prediction time. It should also be mentioned that in addition to the classical "off-line" scenario, where the training set is fixed and is not changing over time, some researchers were considering an "on-line" scenario, where the model continuously updated while processing a stream of data, [18].

In this paper we propose a novel strategy that uses regression models in an attribute wise algorithm to impute missing values in the training stage using the target attribute as one of the predictors. We compare our model strategy with commonly used imputation methods and an imputation method that also uses regression models: *Regression Imputation*.

3 IARI: Incremental Attribute Regression Imputation

There are two ideas behind our method for incremental repair of training sets. First, attributes with missing values are repaired one by one, according to the priority of the attribute. The attribute with the highest priority is repaired first, the attribute with the lowest priority is repaired last. Second, the data used for repairing an attribute include all attributes that are already repaired and additionally the target attribute of the original dataset. The choice of the repair algorithm is arbitrary, in principle any regression algorithm can be used here. In our experiments we used Random Forest [3], due to its superior accuracy, speed and robustness. Random Forest requires little to no tuning, which is very important when numerous models have to be developed without human assistance. Additionally, the Random Forest algorithm provides a heuristic for ranking at-

tributes according to their importance. The IARI algorithm uses this heuristic for ordering the attributes.

It might seem counter-intuitive to include the target attribute in the set of predictors to impute an input attribute–it resembles a circular process. However, our goal is to repair a training set with help of any data we have. When the training set is fixed, a final model is trained and it can be applied to fresh data that were not used in the training process, so there is no circularity here. Moreover, results of our experiments demonstrate that including the target variable in the imputation process substantially increases the accuracy of the final model which is validated on data that were not used in the imputation process.

The IARI algorithm consists of two steps: initialization and main loop. During the initialization all attributes are split into two groups: those that contain no missing values (REPAIRED), and all others (TO_BE_REPAIRED). We assume here that the target attribute, y, contains no missing values so it falls into the REPAIRED group. Additionally, the set of attributes with missing values is ordered according to their importance. This is achieved in three steps. First, the training set is repaired with help of a simple imputation method which replaces missing values of continuous attributes by their mean values and missing values of discrete variables are replaced by their most frequent values. Second, a Random Forest model is built on the repaired training set to predict values of y. Finally, the model is applied to randomized out-of-bag samples to measure the importance of all attributes, as described in [6].

When the initialization step is finished, the algorithm enters the main loop which repairs attributes with missing values, one by one, in the order of their importance (from most to least important). To repair an attribute x, IARI creates a temporary training set which contains all attributes that are already repaired (including y) as predictors and x as the target. All records where the value of x is missing are removed from this training set and, depending on the type of x, a classification or regression variant of the Random Forest algorithm is used to model x. Finally, the model is used to impute all missing values of x and x is moved from the TO_BE_REPAIRED to the REPAIRED set.

The pseudo-code of a generic version of the *IARI* algorithm is provided below.

4 Experimental Setup

To compare the existing algorithms with our approach we used five, very different, datasets from various Machine Learning Repositories: *Digits*, *Cover Type*, *House 16H*, *Page Blocks*, and *Concrete Compressive Strength*. For a complete overview of these datasets, see the public IARI repository, [17].

In our experiments we used a popular implementation of the Random Forest algorithm that comes with the *Scikit-learn* Python package, [12]. The key learning parameter, the number of estimators, was set to 100, and the remaining parameters had default values.

For each dataset we run several experiments with 75% of the attributes containing missing values and 25% of the attributes (randomly chosen) containing

Algorithm 1 Incremental Attribute Regression Imputation

Given: A training set X with input attributes $x_1, \ldots, x_n$, a target attribute y, and a classification or regression algorithm ALG

Initialization:
for all *attributes* $x_i \in X$ **do**
  $Nmissing[i] = Count\_missing(x_i)$
  $Importance[i] = ImportanceMeasure(X, x_i, y)$
end for
$REPAIRED = y \cup \{\text{All attributes } x_i \text{ where } Nmissing[i] = 0\}$
$TO\_BE\_REPAIRED = \{\text{All attributes } x_i \text{ where } Nmissing[i] > 0\}$
while $TO\_BE\_REPAIRED \; ! = \emptyset$ **do**
  $Repair\_Attribute = SELECT\_X_i(TO\_BE\_REPAIRED, Importance)$
  $Repair\_Target = Delete\_Missing\_Values(Repair\_Attribute)$
  $Model = ALG.train(REPAIRED, Repair\_Target)$
  for all *records* $A_j \in Repair\_Attribute$ **do**
    if $is\_missing(A_j)$ **then**
      $A_j = ALG.predict(REPAIRED[j])$
    end if
  end for
  $REPAIRED = REPAIRED \cup Repair\_Attribute$
  $TO\_BE\_REPAIRED = TO\_BE\_REPAIRED \setminus Repair\_Attribute$
end while
return *REPAIRED*

no missing values. The amount of missing values in the attributes with missing data, was set to $10, 20, 30, 40, 50, 60$ percent and for each setup we run 20 experiments using different random seeds. In each experiment, the complete dataset was split in a training (80%) and a test set (20%). The deletion of values, repairing the training set and final modeling was performed on the training set. The test set was used to estimate the accuracy of the final model. When removing values from the training set we used two strategies: "missing at random", *MAR*, where values were removed uniformly at random, and "missing not at random", *MNAR*, where only values bigger than the median value of the attribute, were removed uniformly at random.

4.1 Performance Indicators

We measured two aspects of the quality of the imputation. First, we estimated, with help of cross-validation, the accuracy of the final model that was trained on the repaired dataset. The accuracy was measured either by the ratio of correctly classified cases (in case of classification) or by the *coefficient of determination*, R^2, (in case of regression):

$$R^2 = 1 - \frac{\sum_i (p_i - y_i)^2}{\sum_i (y_i - \bar{y})^2}$$

where y_i denotes the target value and p_i the predicted value.

This score indicates how well the model fits the test data. The maximal value of R^2 is 1, meaning the perfect fit; values smaller than 1 reflect the error. Furthermore, the R^2 and accuracy scores of each dataset are measured on final models that were developed with three algorithms: Random Forests, Support Vector Machines and Gradient Boosted Decision Trees. This is to demonstrate that the value of R^2 (or the accuracy score) depends on the regressor or classifier that is being used in the final modeling, and that it not always reflects the quality of the imputation itself.

Second, we measured the quality of the approximation of the imputed values. As all the imputed variables were numeric, we used the *Root Mean Squared Error*, *RMSE*, to measure the difference between the observed and imputed values:

$$\mathrm{RMSE} = \sqrt{\frac{\sum \left(v_{\mathrm{observed}} - v_{\mathrm{imputed}}\right)^2}{n}}$$

To make the comparison of results over various datasets meaningful, we standardized attributes of all training sets by centering them around 0 and dividing by their standard deviations. As the last indicator of algorithm's performance we measured the execution time. For bigger datasets the cpu time might be an issue to consider.

5 Results

For each dataset, we performed 12 experiments: one for each of the percentage levels of missing values (from 10 to 60) combined with the type of missingness (MAR or MNAR). Each experiment was repeated 20 times (with different random seeds) and the results were averaged. Additionally, for each reconstructed training set, we run three algorithms, Random Forests, Support Vector Machines and Gradient Boosted Decision Trees, to build the final models.

The results of our experiments, the accuracy of the final model (R^2 or the ratio of correctly classified cases) and the accuracy of imputation ($RMSE$), are presented in the following subsection. Each row contains averaged results of 20 runs of the same experiment with different random seeds. The amount of missing values and the type of missing values (MAR or MNAR) are shown as well. For the sake of space we report only results for the percentage of missing values 20%, 40%, and 60% for the MAR model, and various percentages for the MNAR model where we used the missing percentages 20%, 40%, and 60% as upper bounds for the percentage of missing values per attribute, but were not always able to delete that many values of the attribute due to the restriction of deleting only values bigger than the median. Let us note, that it may happen that the fraction of records with a value of an attribute bigger than its median might be arbitrarily small, e.g., when an attribute is almost constant. Moreover, in the results presented below, we show the average number of missing values taken over all attributes with missing values.

For the first dataset (Cover Type) we show the results from the Random Forest final model; for the remaining data sets and final model options we do

not show the results due to space limitations. A complete overview of the results, together with software and data used in our experiments, can be found in the public IARI repository, [17].

Each table contains several columns. The first two columns contain information about the percentage of missing values and the type of missingness. The next column, *Ref*, contains the accuracy of the model trained on the original complete dataset: either R^2 for regression problems or classification accuracy for classification problems. The following columns contain results of various imputation methods: Imputation by Mean, Imputation by Median, Imputation by Most Frequent, Predictive Value Imputation using 2-Nearest Neighbour over a dataset imputed by the Mean, Regression Imputation using Random Forests and last but not least, our own algorithm: IARI. Entries in boldface are significantly better than all other entries with the same settings. The significance is tested using the *t-test*, with significance level $p = 0.05$. The absence of a bold entry in the row means that none of the results were significantly better than the others.

5.1 Cover Type Dataset Results

In Table 1 and 2 the accuracy of the model (Accuracy Score) and the quality of imputation ($RMSE$) are shown for the imputation algorithms on 40.000 instances of the Cover Type dataset.

Table 1. Model Accuracy Score on the Cover Type Dataset with 40000 instances using Random Forests

Miss.%	Type	Ref.	Mean	Median	Freq.	PVI	NN	RI	IARI
6	MNAR	0.911	0.871	0.864	0.860	0.868	0.868	**0.881**
10	MNAR	0.911	0.815	0.809	0.803	0.806	0.805	**0.839**
12	MNAR	0.911	0.670	0.678	0.656	0.657	0.663	**0.693**
20	MAR	0.911	0.874	0.887	0.886	0.883	0.880	**0.899**
40	MAR	0.911	0.834	0.859	0.858	0.845	0.845	**0.878**
60	MAR	0.911	0.776	0.824	0.822	0.787	0.799	**0.847**

Table 2. Imputation Quality (RMSE) of each Imputation Algorithm on the CoverType dataset with 40000 instances

Miss.%	Type	Mean	Median	Freq.	PVI	NN	RI	IARI
6	MNAR	0.786	0.795	0.813	0.786	0.776	**0.760**
10	MNAR	0.848	0.852	0.867	0.847	0.838	**0.791**
12	MNAR	0.894	0.884	0.889	0.894	0.894	**0.877**
20	MAR	0.380	0.389	0.414	0.370	0.330	**0.266**
40	MAR	0.540	0.552	0.588	0.533	0.496	**0.422**
60	MAR	0.661	0.676	0.718	0.658	0.630	**0.564**

Table 3. Execution time of Imputation Algorithms on the Cover Type Dataset with values 50% MAR in seconds.

Mean	Median	Freq.	PVI	NN	RI	IARI
0.03	0.11	0.48	61.47	381.75	119.12	

From our test results we can observe that the maximum average amount of MNAR values we can delete from each attribute is around the 12%. Which implies that approximately 88% of the dataset is filled with values below or equal the median of each attribute (probably 0). In Table 3 the execution time for each algorithm is shown for the case of 50% values MAR, which is representative for all the tests on this dataset. Our approach is not the fastest, Replace by Median, Replace by Mean and Replace by Most Frequent are almost instant while PVI, RI and IARI are more complex and take some time. The execution time is mostly dependent on the size of the dataset and mainly on the amount of attributes, and not so much on the amount of missing values.

6 Conclusion

We presented a novel algorithm, IARI, for imputing missing values into training sets. IARI can handle both regression and classification problems. The key advantage of IARI over other imputation methods is the superior accuracy of the final models which are trained on the repaired training sets, and more accurate reconstruction of missing values. On the other hand, IARI requires much more computing resources than its alternatives: 2-3 orders of magnitude. Fortunately, the main algorithm behind IARI, Random Forest, can be efficiently distributed along multiple nodes, significantly reducing the real (wall clock) computation time.

In principle, IARI is a generic algorithm which can be configured in various ways by changing the measure of importance of attributes, ordering of attributes, and the base algorithm used for imputation. Also the initialization step, where only attributes with no missing values are used as a starting set of predictors, can be modified: sometimes adding to this set several attributes with just a few missing values and removing incomplete records from it, lead to better results.

During our experiments with IARI, we noticed that sometimes a simple imputation method may lead to better results than those obtained with IARI. This happens in case of the *Digits* dataset, where values were removed "not at random", see the *IARI repository* [17]. As expected, the quality of IARI approximations of missing values was always significantly better than those obtained by imputing means, but surprisingly, the opposite holds for the quality of the corresponding final models. This is probably caused by the nature of the classification problem and the fact that the Random Forest is not suitable for image classification. Almost in all other cases the IARI algorithm outperforms other imputation methods: both in terms of the accuracy of imputation and the accuracy of the final model.

In most real world cases it is difficult to determine how well a certain imputation algorithm will work. The quality of imputation depends a lot on the dataset and the reason of why values are missing. However, when we know little about a dataset, the IARI algorithm is probably the best choice.

References

1. PASCAL Machine Learning Benchmarks Repository - mldata.org. http://mldata.org/repository/data
2. Bache, K., Lichman, M.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2013)
3. Breiman, L.: Random forests. Machine learning pp. 5–32 (2001)
4. Carpenter, J.R., Kenward, M.G.: Multiple imputation and its application. Wiley, 1st edn. (2013)
5. Greenland, S., Finkle, W.: A critical look at methods for handling missing covariates in epidemiologic regression analyses. American journal of epidemiology 142(12) (1995)
6. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction. Springer, 2 edn. (2009)
7. Henry, A.J., Hevelone, N.D., Lipsitz, S., Nguyen, L.L.: Comparative methods for handling missing data in large databases. Journal of vascular surgery 58(5), 1353–1359.e6 (Nov 2013)
8. Howell, D.C.: The analysis of missing data. London: Sage (2007)
9. Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., Kolehmainen, M.: Methods for imputation of missing values in air quality data sets. Atmospheric Environment 38(18), 2895–2907 (Jun 2004)
10. Lakshminarayan, K., Harp, S.a., Samad, T.: Imputation of missing data in industrial databases. Applied Intelligence 11, 259–275 (1999)
11. Little, R.J.A., Rubin, D.B.: Statistical analysis with missing data. Hoboken N. J. Wiley, 2nd edition edn. (2002)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
13. Rubin, D.B.: Inference and Missing Data. Biometrika 63(3), 581 (Dec 1976)
14. Rubin, D.B.: Multiple imputation for nonresponse in surveys (2004)
15. Rubin, D.B., Schenker, N.: Multiple imputation in healthcare databases: An overview and some applications. Statistics in medicine 10, 585–598 (1991)
16. Seaman, S.R., White, I.R.: Review of inverse probability weighting for dealing with missing data. Statistical methods in medical research 22(3), 278–95 (Jun 2013)
17. Stein, B. van: Incremental attribute regression imputation. http://basvanstein.github.io/IARI/ (2015)
18. Žliobaitė, I., Hollmén, J.: Optimizing regression models for data streams with missing values. Machine Learning 99(1), 47–73 (2015)

Exploring the Impact of Ordering Models in Merging Decision Trees: A Case Study in Education

Pedro Strecht, João Mendes-Moreira, and Carlos Soares

INESC TEC/Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
{pstrecht,jmoreira,csoares}@fe.up.pt

Abstract. Merging decision trees models has been, so far, motivated as a way to avoid both transporting data sets on distributed locations or training very large data sets. This paper presents a novel rationale which is the need to generalize knowledge by grouping models consumed across different decision levels in a non-distributed environment and propose a methodology for this goal. The approach is evaluated using data from the University of Porto, in the context of predicting the success/failure of students in courses. The experiments focus mainly on the impact of the order of models on the overall performance of merged models. Directions of unexplored issues for future research are also discussed.

Keywords: decision tree merging, C5.0, prediction of failure

1 Introduction

Decision trees have the characteristic of not requiring previous domain knowledge or heavy parameter tuning making them appropriate for both prediction, exploratory data analysis. They are also quite good in terms of human interoperability. In this paper, we propose an approach to merge decision tree models based on previous research [1] by turning the focus to studying the impact of different ways of ordering models during the merging process. There are also improvements in the experimental set-up and measures of merged models evaluation. Therefore, this papers presents work in progress and identifies issues still open for research and future work.

The case study used for empirical evaluation uses data from the academic management information system of the University of Porto (U.Porto), Portugal. Due to limitations of space, this paper focuses on the process of merging trees. Therefore, some decisions which were based on domain-specific knowledge and preliminary experiments (e.g. variable selection, parameter setting) as well as some aspects of the results have not been discussed in depth.

The remainder of this paper is structured as follows. Section 2 presents related work of merging decision trees models. Section 3 describes the system architecture and methodology. Section 4 details results and discussion of applying the methodology in the case study. Finally, Section 5 presents the conclusions.

2 Related Work

The motivation to combine prediction models has its origins as a strategy to deal with building models from distributed data. Distribution can occur *naturally* if the data is initially collected in different locations. Every location produces its own data called, in such context, a *local data set*. An example is a company with multiple branches in which each holds its own data. Thus, each branch can have its own model to predict a specific variable of interest. Distribution can also occur when data, even if not geographically spread, is collected into different data sets to create models relating to business entities. An example is a centralized academic database in which student enrollments are grouped by courses (business entity) to allow the possibility of creating a model for each course separately. In this case, although data is initially centralized, it becomes *artificially* distributed to fulfill a business goal.

To build a global model encompassing all available data, each data set has to be transported to a single location and assembled to form a *monolithic data set*[1]. This may be unfeasible, either by security reasons (unsafe connections) or because transportation may be costly. Bursteinas and Long [2] address the problem of data being generated on distributed distant machines connected by "low transparency connections". Even if it is possible to gather all data sets into a monolithic data set, it may still be impossible to train a global model if the number of examples is too large for the available resources (or at least a very slow task). An example is given by Andrzejak, Langner and Zabala [3] which present distribution as a strategy to deal with data sets with "exceeding RAM sizes". Therefore, artificially distributed data appears as a strategy to avoid having very large data sets, as long as there is a way to build a global model from distributed data.

Fig. 1 shows n local data sets being transported over a channel to a specific location and assembled into a monolithic data set MD. In this centralized location, a model M is trained using all available data. This set-up highlights two problems: if, on one hand it is desired to avoid transporting data (#1), even if that is possible, chances are, that the resulting data set would end up being a very large one (#2). Model merging can be used as a technique to address both problems. To avoid transporting data sets, a local model is trained in each distributed location and then transported over a channel to a centralized location where they are merged to form a global model. This alternative set-up has the advantage of enforcing the need to create local models, while at the same time, reduces the amount of transported information (the models can be represented as a set of lines of text). Another application of model merging is to avoid training a model from a very large data set. The data is artificially splitted into different data sets according to some business criteria and then a model is trained for each. After all models are created they are merged together yielding a single model.

[1] a *monolithic data set* is a non-distributed data set situated in a single and specific known location

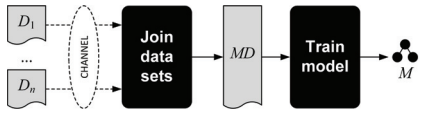


Fig. 1. Training a single model from distributed data

There have been different approaches to merge models, which can be be divided into two main categories: mathematical [4, 5], in which a function is used to aggregate models, and analytical [6–8, 2, 3, 1] in which the models are broken down into parts, combined and re-assembled to form a new model. While the former has been used scarcely, probably due to its complexity, the latter has been more explored. The basic idea is to convert decision trees from two models into decision rules by combining the rules into new rules, reducing their number and finally growing a decision tree of the merged model. The basic fundamentals of the process were first presented in the doctoral thesis of Williams [9]. Over the years, other researchers have contributed by proposing different ways of carrying out intermediate tasks. Table 1 summarizes research examples of this specific approach, specifying the problem (or motivation) and data sets used.

Table 1. Research examples of combination of rules approaches to merge models

Research	**Problem/motivation**	**Data sets**
Hall, Chawla and Bowyer [8]	Train model in a very large data set	Iris, Pima Indians Diabetes
Bursteinas and Long [2]	Mining data distributed on distant machines	UCI Machine Learning Repository
Andrzejak, Langner and Zabala [3]	Train models for distributed data sets and exceeding RAM sizes	UCI Machine Learning Repository
Strecht, Mendes-Moreira and Soares [1]	Generalize knowledge in course models at university level	Academic data from University of Porto

The results of each approach are not easy to evaluate, largely due to the fact that there is not yet a specified standard set-up to assess the quality of merged models. Hall, Chawla and Bowyer [8, 10] evaluated the accuracy of their merged models against a baseline model trained with all examples. A slight improvement of 1% was observed by using the merged model. Andrzejak, Langner and Zabala [3] use the same baseline case and then compare its accuracy by increasing the number of groups. Sixteen groups is the limit where the quality of predictions of the merged model still provides a good approximation to the baseline case. Bursteinas and Long [2] compare accuracy of the test set for the combined tree claiming it to be similar to the accuracy generated with the tree induced on the monolithic data set. Strecht, Mendes-Moreira and Soares [1] defined $\Delta F1$ as the possible gain in the predictive performance by using the group model instead of the individual models. A global improvement of 3% was observed under specific circumstances. A merging score measure is also introduced.

3 Methodology to Merge Models

3.1 Experimental set-up

The system architecture for merging models, presented in Fig. 2, encompasses four main processes. The data is spread (either naturally or artificially) over n data sets ($D_1, \ldots, D_n$). These are assumed to have been created by some data extraction process which is not part of the methodology.

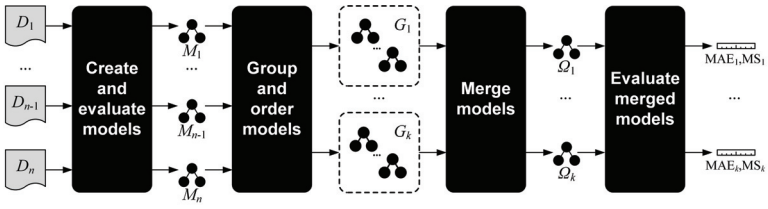


Fig. 2. System architecture of the combination of rules approach to merge models

The first process creates and evaluates n individual decision tree models (M_1, $\ldots$, M_n) for each data set and the corresponding generic evaluation measure ($\eta(M_1)$, $\ldots$, $\eta(M_n)$). The second process organizes the models into k groups ($G_1, \ldots, G_k$) according to some specific criteria. Models order within each group can be random or set by some specific algorithm. The third process merges all models in each group and creates the corresponding merged model. Therefore, for each group ($G_1, \ldots, G_k$) there is a corresponding merged model ($\Omega_1, \ldots, \Omega_k$). Finally, the fourth process evaluates each merged model using parts of the data sets ($D_1, \ldots, D_n$) that were not used to create the individual models. This results in two performance measures for each merged model, the mean absolute error (MAE), and the merging score (MS). It is mandatory that the evaluation measure are the same as the ones used to evaluate the individual models, to allow performance comparison between them.

3.2 Create and evaluate models

In the first process, a decision tree model M_i is created for each data set D_i. Although there are several algorithms to create decision trees, the most popular are CART [11] (Classification and Regression Trees) and C5.0 [12]). For decision tree merging, although recommended, it is not mandatory that all models are trained using the same algorithm. Nonetheless, it is essential to have access to the models themselves, which is a characteristic of decision trees. The result of evaluation correspond to a generic evaluation measure η (e.g., accuracy or F1 [13]), therefore, each model has its own value ($\eta(M_1)$, $\ldots$, $\eta(M_n)$).

3.3 Group and order models

In the second process, the models are gathered into groups and then ordered within each group. Although models can be grouped by any criterion, it is worthwhile to establish a distinction between two major cases: *domain-knowledge* in which models are grouped together according to the data sets meta-information, and *model-driven* in which models are grouped together according the characteristics of the models themselves, i.e., models meta-information. While the former involves business rules criteria, and may imply human judgment to create groups, the latter concerns model similarity, therefore clustering techniques can be used to automatically create groups. The order of models within each group is the main issue being explored in this paper as it is expected to affect the results of the merging process. One possibility under consideration is using the order provided by hierarchical clustering [13] within each group as it orders models according to some distance measure.

3.4 Merge models

In the third process, the models in each group are merged together yielding the *group model*, according to the experimental set-up presented in Fig. 3. A requirement for this process is that each model must be represented as a set of decision rules. This takes the form of a decision table, in which each row is a decision rule. Therefore, the first (M_1) and second (M_2) models are converted to decision tables and merged, yielding the ω_1 model, also in decision table form. Then, the third model (M_3) is also converted to a decision table and is merged with model ω_1 yielding the model ω_2. This process is replicated to all models in the group. The last merged model ω_{n-1} is converted to decision tree form and renamed Ω (referring to the group model).

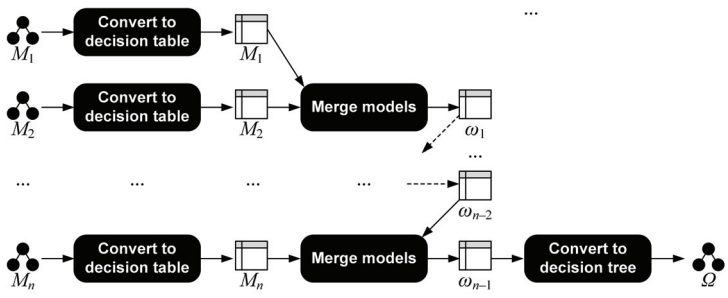


Fig. 3. Experimental set-up to merge all models in a group

In the first subprocess, a decision tree is transformed to a set of rules. Each path from the root to the leaves creates a rule with a set of possible values for

variables and a class. The set of values (nominal or numerical) is the domain of each dimension and each rule defines a region. It is worthwhile observing that all rules lead to non-overlapping regions that together cover the entire multi-dimensional space. Decision rules are represented in a *decision table* which is a linearization of a decision tree. Two special cases have to be considered. The first is when a variable has not appeared yet in a decision. This means that it can take any value without affecting the prediction, and is assigned the limits of 0 and $+\infty$. The second relates to the case of an "empty" model, i.e., one with a single decision region covering the whole space. Although an empty model does not have any rules to be tested, it does have a predicted class.

In the second subprocess, a pair of models (M_1 and M_2) is merged into one (ω) and encompasses four sequential tasks, as presented in Fig. 4. The result of each task is an intermediate model. *Intersection* combines both models yielding model α. This is submitted to *filtering* to remove disjoint regions yielding model β. Disjoint regions corresponds to all pairs of regions in the original models which intersection yields an empty set. In the absence of disjoint regions, β is a copy of α. A possible outcome of this task is that all the regions of α end up being disjoint. In such case, the models are regarded as *unmergeable*. Otherwise, the process moves on to *conflict resolution*. In this task, regions of M_1 and M_2 originally sharing the same space but with different assigned classes have to agree on which class to assign in the merged model. In the literature this issue has been referred to as "class conflict". Several heuristics have been proposed to address it, however, none have been considered flawless. Consequently it remains an issue still open for further research. The resulting model γ is devoided of class conflicts. Finally, the *reduction* task attempts a simpler model ω by identifying adjacent regions that can be joined together (if none is found, then ω equals γ).

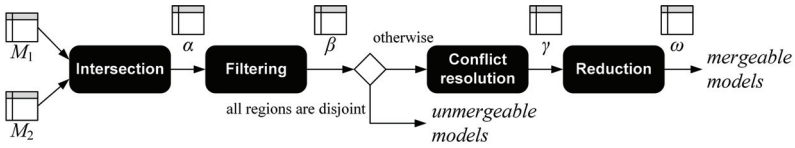


Fig. 4. Subprocess of merging two models

In the third subprocess, the last merged model of the group (ω_{n-1}), in decision table form, is converted to the decision tree representation. Usually, the process of converting a decision tree from a decision table is lossless, meaning that it can be reversed. In other words, it is possible to derive a decision tree by inspection of the corresponding decision table (each region in the table maps into a branch in the tree). However, considering that the decision table to be converted arises from the merging of several models it is no longer guaranteed that all regions of multidimensional space are covered. This is a consequence

of removing regions during the merging process and leads up to the inability to derive a decision tree directly from a decision table. This problem, that we name as "lack of full space coverage", can be addressed by generating a data set representative of the decision table. The idea is to infer the examples that could have yield the decision table, had it been obtained by decision tree induction and subsequent conversion. These are used as learning examples by the same algorithm used to create the initial individual models. The algorithm then induces a model that covers the whole multidimensional space. However, other approaches could be explored, making this an issue still open for further research.

3.5 Evaluate merged models

The purpose of the fourth process is to assess the quality of the merged models and the merging process overall. The merged model is evaluated for each data set, using one tenth of data of the data set (unused to train the original model). Then, we calculate the error made by predicting with the merged model relative to the original model for each data set (eq. 1). We evaluate the global performance of a merged model k by calculating the *mean absolute error* (MAE) [14] for all data sets corresponding to the models in the group (eq.2).

As described previously, one possible outcome of the merging process is the inability to merge two models, due to the lack of common regions. As the merge order is not commutative, is plays a significant role in the ability to merge models, particularly the number of models that is possible to merge within a group. For that purpose, we define the *merging score* (MS) of a merged model as the number of models that is possible to merge (m) divided by the number of pairs of models in the group ($n-1$) (eq. 3).

$$\Delta\eta_i = \eta(\Omega_i) - \eta(M_i) \quad (1) \qquad MAE_k = \frac{1}{n}\sum_{i=1}^{n} \Delta\eta_i \quad (2) \qquad MS_k = \frac{m}{n-1} \quad (3)$$

4 Case Study and Results

4.1 Motivation

Interpretable models for predicting the failure of students in university courses are important to support both course and programme managers. By identifying the students in danger of failure beforehand, suitable strategies can be devised to prevent it. Moreover, those models can give clues about the reasons that lead to student attrition, a topic widely studied in educational data mining [15]. Recently, in the University of Porto (UPorto) set as one of its goals to understand the reasons of this phenomena. The starting point has been to create models to predict if a student is going to pass or fail a course. This meant that a very large number of models was created, which raises problems on how to generalize knowledge in order to have a global view across the university instead of only a single course. Therefore, merging models appears in this context as a technique to address the need to have models at different decision levels.

4.2 Merge models application

The data sets were extracted from the academic databases of UPorto. These store a large amount of data on students, program syllabuses, courses, academic acts and assorted data related to a variety of subprocesses of the pedagogical process. The analysis done focuses on the academic year 2012/2013 with the extraction of 5779 course data sets (from 391 programmes). As a result, there is a data set for each course with student's enrollments described by a set of socio-demographic variables and approval (target variable).

The models trained are decision tree classifiers generated by C5.0 algorithm [12] and students are classified as having passed or failed a course. Experimental setup for evaluation uses k-fold cross-validation [16] with stratified sampling [17]. Failure is the positive class in this problem, i.e. it is the most important class, and thus, we use a suitable evaluation measure F1 [18].

Training, analysis and evaluation of models is replicated for each course in the data set, however, models were created only for courses with a minimum of 100 students enrolled. This resulted in creating 730 models (12% of the 5779 courses). The variables used in the models are age, marital status, nationality, type of admission, type of student, status of student, years of enrollment, and delayed courses. Delayed courses (41%) is the variable most often used, followed by age (16%) and years of enrollment (16%). The quality of the models varies significantly with only a quarter having F1 above 0.60.

Models were grouped in four different ways: scientific area (#1), number of variables (#2), variable importance (#3), and a baseline group containing all models (#4). The C5.0 algorithm measures the importance of variable I_v by determining the percentage of examples tested in a node by that variable in relation to all examples. For creating groups according to variable importance we used the k-means clustering algorithm [13], which created four groups (clusters) using only three of the most important variables, namely age, years of enrollment, and delayed courses.

4.3 Results and discussion

The methodology was applied several times combining different ways to perform the intermediate steps, as described in our previous research [1]. For this study, we used the combination that yielded the best results and explored the effects of ordering in both the mean absolute error and merging score on each of the four arrangements of grouping. For that purpose, we carried out four experiments ordering models by different criteria: random, number of variables, number of examples, and euclidean distance considering variables age, years of enrollment and delayed courses. All three cases were carried out by getting the order of applying hierarchical clustering in each group. For group set evaluation, we normalized MAE and MS of all groups by the number of models in each group, and to allow experiments comparison, we averaged MAE and MS of all group sets.

The results are presented in Table 4.3 and are somewhat surprising. Contrary to what one would expect, the order of models have a minimal effect on the

Table 2. Model merge order criterion comparison results

Order criterion	MAE	MS
Random	0.1927	0.7347
# variables	0.1877	0.7350
# examples	0.1299	0.7349
Age, years of enrollment, and delayed courses	0.1817	0.7347

overall values of MAE and MS. The average MAE across groups sets for using the merged models instead of the original models does not exceed 20%. It is observed that the order of the models does not have a expressive impact on this result. The MS is always 73%, showing again that the merge order does not influence results whatsoever. Given the fact that the merging operation is not commutative, one would expect that the results would present large variations. This suggests that further study is needed to understand the reasons for this lack of variation. In addition, the best results were obtained when no ordering is performed, which in this case is the order in which the data sets were captured in the database. As these results relate to work in progress, additional research will carry on to explore the models order issue and its impact on results.

5 Conclusions

The approach of merging models by combining decision rules is the most often found in the literature. This paper suggests a systematic methodology that can be used both for naturally distributed data or artificially distributed data.

The most suitable representation to merge models is working with decision tables, the combination of decision rules algorithm (the core of the whole process) is where the major differences are found. The main problem to deal with is class conflict in overlapping rules which has no consensual approach. Efforts to simplify the resulting merged model are always included mainly by attempting to reduce the number of decision rules. The final sub-process of growing a decision tree representation of the merged model also presents challenges and should be further explored in future research.

The case study explores the merge order of models and presents unexpected results. Although the merging operation is not commutative, variations in the order of models during merging did not affect the overall performance results of the process. The reasons for this will be explored in further research.

Acknowledgments. This work is funded by projects "NORTE-07-0124-FEDER-000059" and "NORTE-07-0124-FEDER-000057", financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

References

1. P. Strecht, J. Mendes-Moreira, and C. Soares, "Merging Decision Trees: A Case Study in Predicting Student Performance," in *Advanced Data Mining and Applications* (X. Luo, J. Yu, and Z. Li, eds.), Lecture Notes in Computer Science, pp. 535–548, Springer International Publishing, 2014.
2. B. Bursteinas and J. Long, "Merging distributed classifiers," in *5th World Multiconference on Systemics, Cybernetics and Informatics*, 2001.
3. A. Andrzejak, F. Langner, and S. Zabala, "Interpretable models from distributed data via merging of decision trees," *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, Apr. 2013.
4. H. Kargupta and B. Park, "A fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 216–229, 2004.
5. K. Y. Gorbunov and V. a. Lyubetsky, "The tree nearest on average to a given set of trees," *Problems of Information Transmission*, vol. 47, pp. 274–288, Oct. 2011.
6. F. Provost and D. Hennessy, "Distributed machine learning: scaling up with coarse-grained parallelism," in *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, vol. 2, pp. 340–7, Jan. 1994.
7. F. Provost and D. Hennessy, "Scaling up: Distributed machine learning with cooperation," in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 74–79, 1996.
8. L. Hall, N. Chawla, and K. Bowyer, "Combining decision trees learned in parallel," *Working Notes of the KDD-97 Workshop on Distributed Data Mining*, pp. 10–15, 1998.
9. G. Williams, *Inducing and Combining Multiple Decision Trees*. PhD thesis, Australian National University, 1990.
10. L. Hall, N. Chawla, and K. Bowyer, "Decision tree learning on very large data sets," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 2579 – 2584, 1998.
11. Breiman, Friedman, Olshen, and Stone, *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.
12. M. Kuhn, S. Weston, N. Coulter, and R. Quinlan, "C50: C5.0 Decision Trees and Rule-Based Models. R package version 0.1.0-16," 2014.
13. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann, 2011.
14. C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005.
15. G. Dekker, M. Pechenizkiy, and J. Vleeshouwers, "Predicting students drop out: a case study," in *2nd International Educational Data Mining Conference (EDM09)*, pp. 41–50, 2009.
16. M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B*, vol. 36, no. 2, pp. 111–147, 1974.
17. R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Conference on AI (IJCAI)*, (San Mateo, CA), pp. 1137–1145, Morgan Kaufmann, 1995.
18. N. Chinchor, "MUC-4 Evaluation Metrics," in *Proceedings of the 4th Message Understanding Conference (MUC4 '92)*, pp. 22–29, Association for Computational Linguistics, 1992.

Learning an Optimized Deep Neural Network for Link Prediction on Knowledge Graphs

Wilcke W.X.*

Department of Computer Science
VU University Amsterdam
Amsterdam, The Netherlands
w.x.wilcke@vu.nl

Abstract. Recent years have seen the emergence of graph-based Knowledge Bases build upon *Semantic Web* technologies, known as *Knowledge Graphs* (KG). Popular examples are *DBpedia* and *GeoNames*. The formal system underlying these KGs provides inherent support for deductive reasoning. Growing popularity has exposed several limitations of this ability, amongst which are scalability and uncertainty issues, as well as coping with heterogeneous, noisy, and inconsistent data. By supplementing this form of reasoning with Machine Learning algorithms, these hurdles are much more easily overcome. Of the existing research in this area, only a handful have been considering a *Deep Neural Network*. Moreover, only one of these studies has addressed the problem of hyper-parameter optimization, albeit under specific conditions. To contribute to this area of research, we propose a research design that will investigate a *Deep Neural Network* with optimized hyper-parameters for its effectiveness to perform link prediction on real-world KGs.

Keywords: Knowledge Graphs·Semantic Web·Relational Data·Machine Learning·Deep Learning·Neural Networks·Hyper-Parameters

1 Introduction

In 2001, Tim Berners-Lee introduced the world to his vision of a Semantic Web (SW) [1]; a network of semantically annotated and interconnected information, which are interpretable by both man and machine. At the time of writing, this vision has become a reality, with a tremendous amount of information having been made available as such. This information is structured as a graph, called a Knowledge Graph (KG), in which factual information is encoded as relations (edges) between entities (vertices). Well-known examples are *DBpedia*[1] and *GeoNames*[2], the first of which holds information extracted from *Wikipedia*[3], and the second

* Special thanks go to my supervisors Frank van Harmelen and Henk Scholten, as well as to my daily supervisors Victor de Boer, Niels van Manen, and Maurice de Kleijn.

[1] See DBpedia.org

[2] See GeoNames.org

[3] See Wikipedia.org

of which holds basic geographical information on most of the world's places. To ensure a correct interpretation of the facts encoded within such KGs, both their relations and entities are assigned semantic labels. In addition, the definitions of these labels are firmly fixed by shared ontological background knowledge.

Reasoning engines for KGs typically make use of their inherent deductive abilities. This form of reasoning is completely dependent on axiomatic prior knowledge. Hence, it is solely able to derive information that was already implicitly present in the data [2,3]. By supplementing deductive reasoning with methods from Machine Learning (ML), which reasoning inductively, it becomes possible to truly gain additional information [4,5]. Moreover, unlike deduction, these methods are generally able to cope with uncertainty, inconsistency, and noise, all of which are abundant in real-world data. In addition, they tend to suffer less from scalability issues.

Methods that learn from relation data fall under the fairly recent field of Statistical Relational Learning (SRL) [3], [6]. Of all research within SRL, only a small part involves learning from KGs. Currently-popular approaches are Inductive Logic Programming, logic and graph-based kernels [7,8], and matrix and tensor factorization [2], [9,10]. In contrast, despite having several potentially-useful characteristics, only limited attention appears to be have been given to Neural Networks (NN).

A typical NN consists of one or more hidden layers which, when trained, represent one or more latent features within a data set [11]. Latent-feature models are a sensible choice to learn from real-world data, due to their robustness towards noise and inconsistencies, as well their ability to cope with large-scale and high-dimensional data sets. Furthermore, NNs are universal function approximators, which allows them to model any arbitrary relational structure, given enough model complexity. Moreover, recent breakthroughs have made it possible to effectively learn deep NNs, which radically improves their ability to solve complex learning problems [12]. Together, these characteristics make for an interesting alternative to the currently-popular approaches mentioned above. Nevertheless, deep NNs have only been applied a handful of times to the learning problems we are considering in this paper, and even fewer have taken on the challenge of exploiting (ontological) graph features for improved predictive performance, even though this has been proven useful [13,14]. Moreover, to the best of our knowledge, only one of these studies has yet addressed the optimization of its model's hyper-parameters, despite the positive influence thereof on the performance.

In this paper, we propose a research design for investigating the effectiveness of a hybrid latent and graph-feature model capable of performing link prediction on real-world KGs. To this end, we intent to develop a deep feedforward NN with the ability to learn from complex relational data. For optimization purposes, we will additionally investigate the effectiveness of learning a set of (near) optimal hyper-parameters through Bayesian optimization. To this end, we will first look at relevant background knowledge in Sect. 2, followed by a discussion of our proposed research design and evaluation method in Sect. 3. Section 4 will continue that discussion by looking at the domain within which our experiments will take

place. This is followed by relating our design choices to state-of-the-art research and development in Sect. 5. Finally, Sect. 6 will end this proposal with several last remarks.

2 Background

For the purpose of this paper, we define a KG as a semantic graph $G = (\mathcal{E}, \mathcal{R})$, with $\mathcal{E} = \{e_1, \ldots, e_{n_\mathcal{E}}\}$ and $\mathcal{R} = \{r_1, \ldots, r_{n_\mathcal{R}}\}$ being the finite set of semantically-enriched entities and relations, respectively. In addition, each fact in G constitutes a triple $t_{ijk} = (e_i, r_k, e_j)$, which reflect a binary relation between two entities. Furthermore, we let $\mathcal{T}$ denote the set of all existing triples in G.

An example of a small KG is depicted in Fig. 1. There, vertices represent different entities from the domains of people, universities, and operating systems. These entities are related to each other through the graph's edges, with their labels reflecting the relationship they represent. For instance, we can observe that `Andrew S. Tanenbaum` worked at the `VU University Amsterdam` and that he is the creator of `Minix`. However, as a typical KG is subject to the Open World Assumption (OWA), the converse is not necessarily true. Hence, while there is no edge between `Andrew S. Tanenbaum` and the `University of Helsinki`, we cannot simply assume that he never had any affiliation with that institute; it is just unknown to us. However, through simple deduction, we do know his rival did.

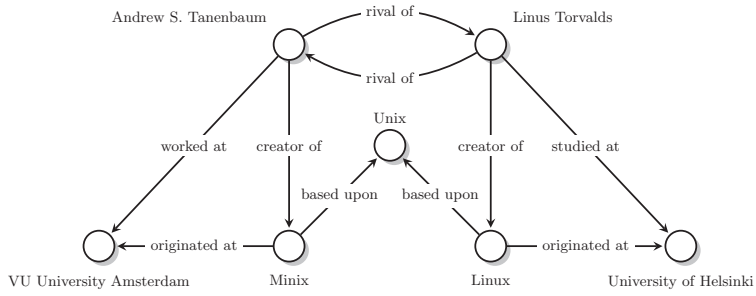


Fig. 1. Example of a small KG on software kernels. Entities and relations are represented by vertices and edges, respectively. Note that any ontological background knowledge has been omitted for reasons of clarity.

For the purpose of clarity, semantic properties of both entities and relations have been omitted from Fig. 1. Nevertheless, a possible ontology might specify that both the `VU University Amsterdam` and the `University of Helsinki` are of the class `university`. Moreover, the `university` class is likely to be a

subclass of yet-another class, e.g, `educational institute`. Through transitivity, we can thus infer that the `VU University Amsterdam` is an `educational institute` as well, despite it not being explicitly stated. Similarly, the relations `worked at` and `studied at` might both be associated with the broader-defined relation `affiliated with`, hence allowing us to infer that `Andrew S. Tanenbaum` is `affiliated with` an `educational institute`. Moreover, this broader relation might be subject to any number of constraints, e.g. requiring that its source and target entities are instances of the `people` and `organization` class, respectively.

Apart from their complex relational structure, KGs typically posses several other characteristics that make learning from them challenging [5], [9], [15,16]. Firstly, their data are often highly heterogeneous, ranging from textual and numerical to shapes or other abstract data types. Secondly, $\mathcal{R}$ generally holds only positive relations, which may limit the effectiveness of many learning methods. Moreover, as mentioned above, the framework underlying a KG is subject to the OWA, making it is incorrect to simply assume omitted relations as being false[4]. Together, these characteristics typically result in a rather low-density graph or, equivalently, a rather sparse data set.

2.1 Hyper-Parameter Optimization

Selecting suitable values for a set of hyper-parameters may considerably improve the performance of an learning algorithm [17]. Several hyper-learning algorithms exist by which this set of values can be optimized. These range from simple but naive algorithms, such as grid and random search, to the more-advanced algorithms which guide the search towards an optimum. A well-known example of the latter is Bayesian optimization.

Bayesian optimization is a global optimization methodology for noisy and expensive black-box models [18,19]. For this purpose, it relies on the construction of a relatively cheap probabilistic model that represents a distribution over loss functions for a given set of hyper-parameter values. The added value of this distribution is that it serves as a prior. As a result, finding (near) optimal values generally requires less iterations than alternative optimization methods.

Different models can be used for Bayesian optimization. An often-used model is the Gaussian process, which is simple and flexible. It is known, however, to scale cubicly, which renders it infeasible for optimizing the parameters of complex models. A well-established alternative is to use trees or, better still, random forests, which scale linearly [18]. Another model that scales linearly is a deep NN, which has shown to achieve state-of-the-art performance [19].

3 Research Design

Our high-level goal is to develop a predictive model to aid in a knowledge discovery (KD) process. Resulting predictions will be used for data enrichment,

[4] While generally true, there are examples of KGs that hold a Local OWA.

as well as for the generation of potential hypotheses. In addition, they can be used for data validation. For this purpose, we propose the use of a hybrid latent and graph-feature model, represented as a deep feedforward NN. The intended learning task involves link prediction, which constitutes estimating the possible existence of unknown relations between existing entities in KGs. To this end, we intent to use state-of-the-art theories and techniques from the field of ML, thereby placing emphasis on deep learning and Bayesian optimization. Moreover, in order to improve predictive performance, we plan to investigate the integration of ontological background knowledge.

3.1 Research Hypothesis

We define the following high-level research hypothesis:

> *Knowledge discovery by learning from large-scale KGs containing real-world data through means of a state-of-the-art deep feedforward NN provides significant advantage over comparable methods.*

The above hypothesis can broken apart into the following research questions:

1. Can a suitable propositionalization function be defined, such that it can translate an arbitrary KG to a NN, whereby the relational information that is lost, if any, is less or equal than that of comparable methods?
2. Do the following additions improve predictive performance, and is this improvement significant enough to justify the expected increase in the required computational and temporal resources?
   (a) The preprocessing of graph data prior to learning using various techniques, amongst which are partial and full materialization, as well as the generation of negative instances and ignoring very-frequent relations with a low-explanatory score.
   (b) The incorporation of state-of-the-art advances from the field of ML, thereby emphasizing deep learning for training purposes and Bayesian optimization for learning (near) optimal hyper-parameters.
   (c) The ability to, in addition to latent features, exploit (ontological) graph features from either or both the instance data and their corresponding ontological background knowledge.
3. How well are deep NNs able to cope with commonly-existing facets of real-world data, amongst which are heterogeneity, uncertainty, inconsistency, and noise, as well as often-seen issues of learning from KGs, amongst which are scalability problems, integrity constraints, and the imbalance between positive and negative examples in the corresponding data sets?
4. Are the resulting predictions relevant enough (i.e. non-trivial) to domain experts for them to be considered 'new' or 'useful knowledge', as well as accurate-enough to be considered trustworthy for usage in scientific research?

3.2 Methodology

Our projected research will consist of several phases which will gradually build up to the final model. To this end, we have begun the development of a deep NN and its corresponding learning method. Following extensive testing, we will extend this learning method to learn the network's (near) optimal set of hyper-parameters using Bayesian optimization. Upon completion, we will iteratively add additional layers of complexity to the model, with each one extending its ability to exploit (ontological) graph features.

The exploitation of various graph feature will be explored for their beneficial effect on the overall predictive performance. The algorithms that will extract these features will be developed by us during the course of this research, as well as adopted from recent literature. Examples within that literature are path and authority-ranking algorithms, as well as algorithms to determine various semantic associations between entities. Other examples are the exploitation of class hierarchies and integrity constraints, which are available as ontological background knowledge.

Model Description. Consider a fully-connected feedforward NN, with input vector $\boldsymbol{x}$ and output vector $\boldsymbol{y}$. Here, vector $\boldsymbol{x}$ will be constructed through a concatenation of two vectors $\boldsymbol{x}_{\text{lhs}}$ and $\boldsymbol{x}_{\text{rhs}}$. These vectors describe the local neighbourhood of the left-hand and right-hand side entity from a given triple, respectively. The rationale behind this decision is that we believe such a description to better reflect the relational context of an entity than merely that entity itself. Output vector $\boldsymbol{y}$ will hold the certainties of all possible relations in $\mathcal{R}$. More specific, $\boldsymbol{y}(k)$ will represent the certainty that the corresponding relation $r_k \in \mathcal{R}$ exists between the entities e_{lhs} and e_{rhs} as provided by the input vector. During the learning phase, we will draw training instances from $\mathcal{T}$. Hence, the target relation is known to us, allowing $\boldsymbol{y}$ to be regarded as a *one-hot* vector with real-numbered values. In contrast, during the testing phase, the certainty values in $\boldsymbol{y}$ will be estimated.

To allow for both single-label and multi-label prediction, we will introduce a certainty threshold τ, which defines a cut-off point under which predictions will be deemed untrustworthy. When $\tau = \mathbf{max}(\boldsymbol{y})$, a single-label prediction scheme will be maintained, whereas any other value will result in a multi-label prediction scheme. When $\tau \to 1.0$, this threshold can additionally be used as a means to guard the validity of subsequent predictions. During evaluation, we will iteratively refine the value of τ with the help of domain experts.

Learning Method. At first, the network's weights will be pre-trained layer-wise using a greedy unsupervised algorithm. To this end, each input layer will be treated as a restricted Boltzmann machine, as is the current *de facto* means for deep learning [12]. This is followed by training or, more accurately, fine tuning the weights using supervised back-propagation. For this purpose, we will employ an online-learning scheme, due to its effectiveness in solving large-scale

and difficult classification problems [11]. Furthermore, to guard validity, a k-fold cross-validation scheme will be utilized.

For learning the set of (near) optimal hyper-parameters, we intent to use Bayesian optimization. For this purpose, we will employ random forests, due to their low computational complexity [18]. Hyper-parameters specific to the two learning algorithms will be learned separately, with those of the unsupervised algorithm being kept static during optimization of the hyper-parameters of the supervised algorithm. To compensate for the increase in computational resources, we intent to parallelize this process.

3.3 Evaluation

Our evaluation of the method's effectiveness will make use of both quantitative and qualitative measures. To determine the former, we will calculate the area under the precision-recall curve (AUC-PR). We motivate our choice for this measure over the often-used area under the receiver-operating-characteristic curve, due to its better handling of imbalanced distributions [14]. The set of learning methods that will take part in the quantitative evaluation process will be composed of the final and intermediate models developed during the course of this research, as well as of comparable models from recent literature.

Qualitative measures will involve interviews and questionnaires, with which we intend to evaluate our model for its usefulness towards domain experts. For this purpose, we will organize workshops at regular intervals during which members from relevant communities will be asked to evaluate predictions made by our model on data relevant to their research. Their input will additionally be used to refine threshold parameter τ. At a later stage, we intent to integrate our method into a web-based service for analysing KGs. This will allow us to access a much larger audience.

4 Domain and Data

The effectiveness of our proposed method will be evaluated on several data sets within the domain of digital humanities. With the transition from traditional to digital means, a large number of data sets are becoming available. A number of these sets have since been converted to KGs, e.g. those of the Louvre[5] and the Rijksmuseum[6]. Researchers who are interested in studying those data are still hampered by the lack of effective KD tools.

Our decision to narrow the scope to digital humanities is motivated by the following aspects :

1. Methods underlying KD tools are in high demand within digital humanities, particularly those effective on KGs given the lack thereof.

[5] See louvre.fr
[6] See rijksmuseum.nl

2. Few studies have yet examined the effectiveness of such methods on KGs from digital humanities, as well as the relevance of their outcome to domain experts.
3. Evaluating predictions in cooperation with domain experts will provide a valuable measure of our method's predictive performance.

We will hold several general assumptions about KGs on which we will be evaluating our proposed method. These assumptions are based on interviews with various domain experts, as well as on a preliminary studies of literature and data. Firstly, we assume the KGs to contain real-world data, a large portion of which is uncurated. As a result, these data are assumed to be of a heterogeneous nature, as well as containing noise and inconsistencies. Moreover, we assume geospatial data to be well-represented within these KGs.

4.1 ARIADNE

A major use case in our research is European Union's Seventh Framework Programme ARIADNE[7]; a four-year project which aims at providing semantically-enriched archaeological information on an European level. For the most part, this involves the process of converting multilingual field reports, either on paper or stored digitally, to KGs. These reports consist of various types of data, including text, tables, figures, and photographs.

Another aspect of the research within ARIADNE involves the investigation of methods capable of performing KD on the created archaeological KGs. The research described in this proposal is a part of that study. As a consequence, we have direct access to the project's data, as well as to the archaeological community involved with the project.

5 Related Work

Learning a deep network for KGs has been investigated in several recent studies [20,21,22,23]. Common in all is the use of embedding structures, which utilize a dedicated mapping function that translates an entity or its label into a high-dimensional input vector, or an abstraction thereof. This was shown to work quite well in Natural Language Processing. An extension was proposed by [21], who included additional information on an entity into the embeddings, such as its local neighbourhood, its description, and its class. This is an approach similar to the descriptive vectors which we are proposing to use.

To allow the use of bilinear functions, instead of the conventional sigmoidal non-linearities as we are proposing to use in our model, [20] and [22] substituted one or more hidden layers by third-order tensors. However, this approach required a large number of hyper-parameters to be set, four of which were learned through an unspecified optimization method. Nevertheless, the sheer size of their

[7] Advanced Research Infrastructure for Archaeological Dataset Networking in Europe (ARIADNE). See ariadne-infrastructure.eu

solution space makes this approach virtually intractable when scaled up [13], [24]. This was partial solved by [23], who devised a method to translate third-order tensors into sigmoidal functions.

To the best of our knowledge, none of the deep networks for learning from KGs have explored the exploitation of graph features. In contrast, [13] has shown their usefulness with conventional NNs. For this purpose, they developed a stacking model, in which results from a path-ranking algorithm were fed to a single-layer feedforward NN. This was shown to improve performance. The same result was found with other latent-feature models [14].

6 Final Remarks

We have proposed a research design for investigating the effectiveness of a deep NN for link prediction on real-world KGs. Our study of recent literature indicated that only a handful of studies have been focussing on this topic, and even less have been considering the exploitation of graph features to improve predictive performance. Furthermore, to the best of our knowledge, only one of these studies has yet addressed the problem of hyper-parameter optimization, and only under specific conditions. Given these observations, we strongly believe that this area is in need of additional research.

Acknowledgements. This research is partially funded by the ARIADNE project through the European Commission under the Community's Seventh Framework Programme, contract no. FP7-INFRASTRUCTURES-2012-1-313193.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (May 2001) 29–37
2. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing YAGO: scalable machine learning for linked data. In: Proceedings of the 21st international conference on World Wide Web, ACM (2012) 271–280
3. Rettinger, A., Lsch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the semantic web. Data Mining and Knowledge Discovery **24**(3) (2012) 613–662
4. Tresp, V., Bundschus, M., Rettinger, A., Huang, Y.: Towards machine learning on the semantic web. Springer (2008)
5. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A Review of Relational Machine Learning for Knowledge Graphs: From Multi-Relational Link Prediction to Automated Knowledge Graph Construction. arXiv preprint arXiv:1503.00759 (2015)
6. Getoor, L., Taskar, B., eds.: Introduction to statistical relational learning. Adaptive computation and machine learning. MIT Press, Cambridge, Mass (2007)
7. d'Amato, C., Fanizzi, N., Esposito, F.: Classification and retrieval through semantic kernels. In: Knowledge-Based Intelligent Information and Engineering Systems, Springer (2008) 252–259
8. Losch, U., Bloehdorn, S., Rettinger, A.: Graph kernels for rdf data. Number 5781-5782 in Lecture notes in computer science, Lecture notes in artificial intelligence, Springer (2009)

9. Franz, T., Schultz, A., Sizov, S., Staab, S.: Triplerank: Ranking semantic web data by tensor decomposition. Springer (2009)
10. Lippert, C., Weber, S.H., Huang, Y., Tresp, V., Schubert, M., Kriegel, H.P.: Relation prediction in multi-relational domains using matrix factorization. In: Proceedings of the NIPS 2008 Workshop: Structured Input-Structured Output, Vancouver, Canada, Citeseer (2008)
11. Haykin, S.S.: Neural networks and learning machines. 3rd ed edn. Prentice Hall, New York (2009)
12. Bengio, Y.: Learning Deep Architectures for AI. Foundations and Trends in Machine Learning **2**(1) (2009) 1–127
13. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2014) 601–610
14. Nickel, M., Jiang, X., Tresp, V.: Reducing the Rank in Relational Factorization Models by Including Observable Patterns. In: Advances in Neural Information Processing Systems. (2014) 1179–1187
15. d'Amato, C., Fanizzi, N., Esposito, F.: Inductive learning for the Semantic Web: What does it buy? Semantic Web **1**(1) (2010) 53–59
16. Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data. Machine Learning **94**(1) (2014) 233–259
17. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In: Neural Networks: Tricks of the Trade. Springer (2012) 437–478
18. Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2013) 847–855
19. Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Ali, M., Adams, R.P., others: Scalable Bayesian Optimization Using Deep Neural Networks. arXiv preprint arXiv:1502.05700 (2015)
20. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). Volume 1631., Citeseer (2013) 1642
21. Huang, H., Heck, L., Ji, H.: Leveraging Deep Neural Networks and Knowledge Graphs for Entity Disambiguation. arXiv preprint arXiv:1504.07678 (2015)
22. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems. (2013) 926–934
23. Yu, D., Deng, L., Seide, F.: Large Vocabulary Speech Recognition Using Deep Tensor Neural Networks. In: INTERSPEECH. (2012)
24. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Learning Multi-Relational Semantics Using Neural-Embedding Models. arXiv preprint arXiv:1411.4072 (2014)

Toward Improving Naive Bayes Classification: An Ensemble Approach

Khobaib Zaamout[1] and John Z. Zhang[2]

[1] Department of Computer Science
University of Calgary, Calgary, AB Canada T2N 1N4
kzaamout@cs.ucalgary.ca
[2] Department of Mathematics and Computer Science
University of Lethbridge, Lethbridge, AB Canada T1K 3M4
zhang@cs.uleth.ca

Abstract. In this paper, we present our preliminary results toward improving the classification accuracy in naive Bayes classifiers using ensemble techniques. We show that using predictions of a naive Bayes classifier as input to another naive Bayes classifier trained on the same dataset will improve the accuracy of classification. We consider two variations of this approach, single-link chaining and multi-link chaining. Both variations include predictions of a trained naive Bayes classifier in the construction and training of a new one and then store these predictions for later inclusion. In both variations, the construction process continues until acceptable error reduction is achieved. The effectiveness of our proposed approach is demonstrated through a series of empirical experiments and discussions on real and synthesis datasets. But we leave the theoretical analysis of the approach as our future work.

Keywords: Machine learning, Naive Bayes classifier, Ensemble techniques

1 Introduction

Classification is one of the most important pattern-recognition tasks in machine learning and is a process to assign known class label(s) to an unknown object [2]. Essentially, an object is a collection of numerical and/or categorical features. In mathematical terms, the object's class is the output of a linear or nonlinear function of its features. Classification is an example of supervised learning where a machine learner would learn from a training set of correctly classified objects to be able to infer the classes of new ones.

Many algorithmic techniques have been developed over the past decades to automate the classification process. *Naive Bayes Classifier* (*NBC* for short) is amongst the best techniques used for classification [16]. NBC is a simple probabilistic classifier based on applying Bayes' Theorem with naive or strong independence assumptions among the features in a particular problem domain. In practice, such naive independence assumptions exist to a certain degree in many

problem domains, thus making NBC an effective classification approach to them. Examples include *textual information retrieval*, *medical analysis*, etc. Other classification techniques include *Logistic regression* [6], *Support vector machine* [5], *Neural Networks* [15], etc.

Since naive Bayes classifier is a simple probabilistic classifier which assumes conditional independence amongst all the observed variables, it is very scalable. However, due to NBC's simplicity, it is more understandable that other more sophisticated approaches, such as support vector machine, neural networks, and Bayesian networks, dwarfed NBC's predictive performance. Therefore there still remains natural interests in further increasing an NBC's classification power.

In our work, we focus on an ensemble of NBCs. The intuition behind our approach is concerned with combining the outputs of trained NBC on the same dataset or a subset of the dataset through some technique. Since NBCs will probably make errors on different areas of the input space of a problem domain, a good combination technique will yield an ensemble that is less likely to be at fault and is more error tolerant [9].

The paper is organized into the following structure. Section 2 discusses the related previous work along the same direction we are working. In Section 3, we mainly discuss our proposed ensemble approach aiming at improving the classification accuracy of NBC by grouping a set of NBCs in a novel way. For our experiments in Section 4, we present the datasets and the related preprocessing techniques. We also introduce some evaluation measures on classification. Section 5 is for our discussions on the performance of our proposed ensemble approach, where we attempt to discuss it from different perspectives. We conclude our presentation in Section 6, with remarks on our future work.

2 Previous work

In essence, improving the classification accuracy of a classifier can be achieved through various means. For instance, given a classification task, we could reduce the feature space or perform some preprocessing tasks to clean input data. Another approach that gains wide attractions is to utilize the outputs of a classification in an intelligent manner. Output utilization and boosting are the manipulations of the results of a classification algorithm such that the output is enhanced. The latter technique is generally known as ensemble.

Ensemble refers to the techniques of combining the outputs of a number of diverse classifier(s), through some gating function, in order to arrive at better classification of any individual member of the ensemble. [7] describes an ensemble of classifiers as the set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new objects. It is conjectured that ensembles overcome limitations of individual classifiers. Given the inherent diversity of ensembles they are more robust towards local optima problem and more tolerant to small training set sizes than the individual ensemble members.

Bagging [3] and Boosting [17] are popular examples of ensemble combining mechanisms. Bagging, or Bootstrap aggregating, is carried out by training various classifiers on different subsets of a dataset and combining their collective vote on a new object through averaging. Boosting refers to the process of incremental construction of the ensemble by training new classifiers on instances that were misclassified by preceding ensemble members and presenting the final output of the ensemble as the weighted sum of all ensemble members. Other examples of ensemble combining techniques are plurality, where the correct classification is the one that was agreed upon by the largest number of ensemble members. An example is majority voting, where the correct classification is the one voted on by more than half of the ensemble members, etc. [9].

Some combining mechanisms, such as Bagging, have been shown to consistently reduce error. Others, such as Boosting, have been shown to significantly reduce error but inconsistently, and suffer from overfitting in the presence of random noise [13].

3 Our Proposed Approach

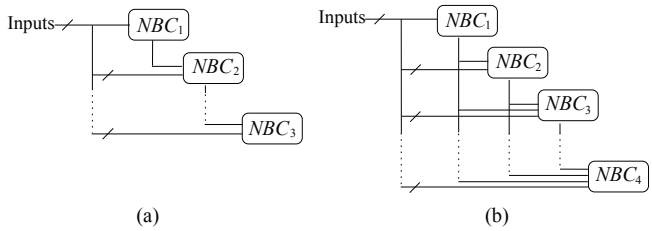


Fig. 1. (a) Single-Link Chaining. (b) Multi-Link Chaining.

In our work, we consider a novel ensemble approach using naive Bayes classifiers. We call our approach *Chaining Ensemble of Naive Bayes Classification* (*CENBC* for short). It aims to improve an NBC's predictions by including the predictions of the previously trained NBC(s) into its current training process, forming a chain-like ensemble. This approach has two variations, *single-link chaining* (*SLC* for short) and *multi-link chaining* (*MLC* for short). SLC, as shown in Figure 1 (a), trains a naive Bayes classifier on a dataset and then uses its predictions as input to another classifier along with the given dataset. The chaining process continues forming a chain of classifiers, i.e. *chain links*, until an acceptable error is achieved. Each classifier in the "chain" is trained on the original dataset and on the predictions of the classifier that immediately precedes it. This approach increases the number of features in the original dataset by only one, keeping computational cost of creating a new classifier feasible. *MLC*, as shown

in Figure 1 (b), is similar to the SLC variation. It differs in that each naive Bayes classifier in MLC is trained on the original dataset and on the predictions of all classifiers that precede it. This causes the creation of new classifier to become computationally expensive. It is easy to see that both variations undergo the same chain links generation process but differ in the way they use these links.

In general, the intuition behind both variations is that an NBC's predictions can be used to correct the predictions of the upcoming NBCs. This is because these predictions are resulted from the features that are indicative of the target classes of a given classification task. Therefore, the predictions are highly correlated with the target classes. Using these predictions is therefore expected to further improve the predictability of the classification process by NBC.

In particular, the intuition behind SLC is that an NBC in the chain may not need the predictions of all preceding NBCs in order to correct its classification. An NBC trained on the predictions of a previous NBC produces predictions influenced by that knowledge. Therefore, it seems reasonable that the predictions of the new NBC should replace that of the previous NBC in the dataset, thus, avoiding an unnecessary increase in calculations. The intuition behind MLC is that it may be necessary for an NBC to have access to all the predictions of previous NBCs. This way, it is left up to the training procedure to learn what it finds beneficial.

One critical issue arises in the SLC and MLC variations regarding the generations of the chain links, i.e., in determining the number of chain links required to reduce the overall error of the NBC(s) to the minimal. While we believe that it is difficult to conduct a formal analysis on these problems, we will definitely attempt to tackle them in our future investigations.

Actually the current proposed approach is part of our framework that attempts to introduce ensemble techniques into machine learning algorithms. We have done some work [19] on the same the ensemble technique but using *neural networks* (*NN* for short) instead of NBC. While the ensemble shows promising improvements over NN itself alone, the complex structure of NN, the uncertainty of many parameters in NN, the complexity of the learning algorithms in NN, etc., pose great difficulty for us to conduct formal theoretical analysis of our approach. At the moment, we are still working on the problem. But we hope that our work with the current ensemble using NBC(s), thanks to NBC's simplicity, would shed light on attempt on the problem.

4 Empirical Experiments

4.1 Setup and Datasets

Preprocessing refers to performing some work on the raw data in order to extract specific features to improve classification accuracy [18]. In our work, we make use of three preprocessing methods, namely, *Principal Component Analysis* (*PCA*), *Correlation-based Feature Selection* (*CFS*), and *ReliefF*. PCA is a multivariate analysis technique that takes as input a dataset of inter-correlated attributes and

produces a new smaller dataset of independent (i.e. orthogonal) attributes (i.e. principal components) that retain most of the original dataset properties [1]. CFS is a feature filtering algorithm. It selects a subset of attributes such that they are highly correlated with the class attributes while being the least correlated with each other [8]. ReliefF is another feature filtering algorithm that ranks attributes based on their relevance to class attributes. A selected attribute would contain values that distinguish for different classes and are similar for the same class [10, 11].

Four datasets are used to validate our proposed approach, namely *Cardiotocography* (*CDO* for short), *Steel Plates Fault* (*SPF* for short), *Chronic Disease Survey* (*CDS* for short), and *Spambase Dataset* (*SDB* for short). Their summaries are shown in Table 1.

Table 1. Summary of datasets

Dataset Name	# Attributes	# Instances	Class Type (#)	Data Characteristics
D_{CDO}	23	2126	Categorical(10)	Integer
D_{SPF}	27	1941	Categorical(7)	Continuous
D_{CDS}	8	2200	Categorical(6)	Integer-Continuous
D_{SDB}	57	4601	Categorical(2)	Numeric

D_{CDO} is obtained from the UCI machine learning repository [12]. It is a dataset for classification with the goal of predicting the magnitude of fatal heart rate. D_{SPF} is also obtained from [4]. It records various aspects of steel plates, such as type of steel, thickness, luminosity, etc., which allow predicting various faults in steel plates. D_{SDB} is a dataset in which the attributes encode different characteristics indicative of spam or non-spam emails [12]. D_{CDS} is a real-life dataset obtained from the government of Canada data portal [14] as part of the open data initiative adopted by the Canadian government. It contains data sources from every Canadian province and territory to estimate the incidence and prevalence of chronic conditions, as well as related risk factors, use of health services and health outcomes. D_{SDB} is a collection of spam emails came from postmasters and individuals that file spams. Classification is conducted to assign whether an email is a spam or not.

4.2 Evaluation Measures

Three evaluation measures are considered to evaluate the performance of the proposed approach: *precision*, *recall* and *F1-measure* [16]. Precision shows the fraction between the number of the objects whose predicted class labels matched with the target class compared to the total number of objects who are predicted as target class by a classifier. Recall presents the fraction between the number of the objects whose predicted class labels matched with the target class compared

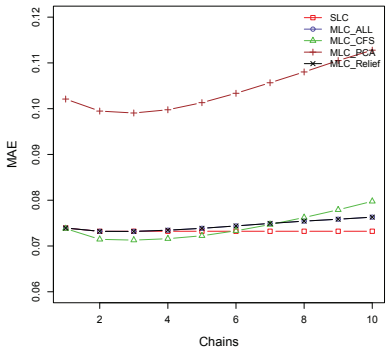


Fig. 2. For dataset D_{CDO}.

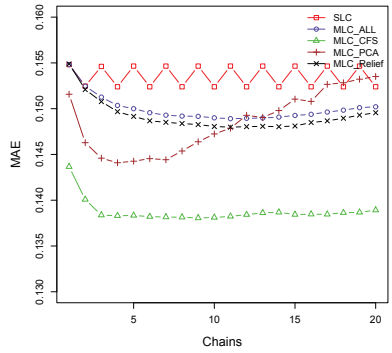


Fig. 3. For dataset D_{SPF}.

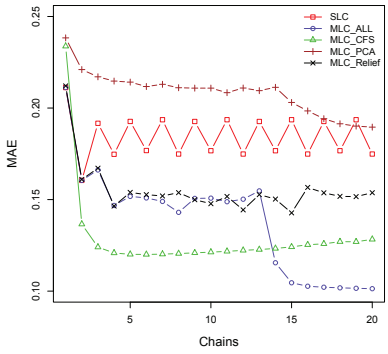


Fig. 4. For dataset D_{CDS}.

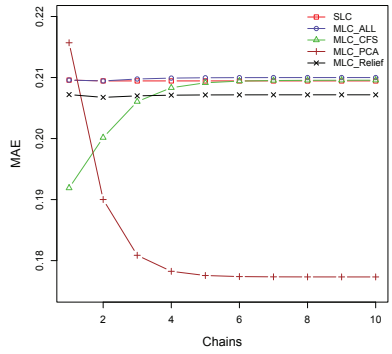


Fig. 5. For dataset D_{SDB}.

to the total number of objects who are of the target class. In other words, precision is the fraction of retrieved objects that are of the target class, while recall is the fraction of the objects with the target class that are retrieved. F1-measure is to consider the precision and recall measures of a classifier on a given task together, presenting a harmonic mean of the two.

5 Results and Discussions

All the experiment are conducted using 10-fold cross validation. In Figures 2, 3, 4, and 5, MAE stands for the standard *Mean Absolute Error*, the legend ALL means that all the features are used in the training and testing process, and others, such as CFS, express that the features used are obtained through the corresponding feature reduction techniques.

The results from our empirical experiments are encouraging, as shown in Table 2, where we summarize the performance of our proposed approach in D_{CDO}, D_{SPF}, and D_{CDS}. One observation shows that for real dataset D_{CDS}, our proposed CENBC sees the most significant reductions of errors, for both SLC and MLC, with an average error reduction of 72%. Their corresponding Precision, Recall and F1 measures are also promising, as in the table. Within a couple of links, we already see the reductions of errors.

Table 2. 10-fold cross validation Mean Absolute Error - MAE (third column) obtained by a typical NB classifier for each dataset (first column) along with the lowest MAE (fifth column) achieved by different chaining mechanisms (second column) and the percentage of error reduction achieved (sixth column) in the given number of chains (fourth column). We also provide the precision, recall, and F1 measures.

	Datasets	NB - MAE	Chain#	MAE	% Reduction	Precision	Recall	F1
	SLC	0.0739	2	**0.0732**	1.00%	0.725	0.637	0.651
	MLC ALL	0.0739	3	**0.0732**	1.02%	0.726	0.636	0.650
CDO	MLC CFS	0.0738	3	***0.0713***	3.53%	0.758	0.666	0.684
	MLC PCA	0.1021	3	**0.0991**	3.07%	0.639	0.540	0.546
	MLC RELIEF	0.0739	3	**0.0732**	1.02%	0.726	0.636	0.650
	SLC	0.1548	4	**0.1524**	1.59%	0.820	0.467	0.556
	MLC ALL	0.1548	11	**0.1489**	3.97%	0.826	0.489	0.578
SPF	MLC CFS	0.1437	9	***0.1381***	4.06%	0.827	0.530	0.606
	MLC PCA	0.1516	4	**0.1441**	5.19%	0.824	0.515	0.622
	MLC RELIEF	0.1549	11	**0.1480**	4.67%	0.827	0.494	0.585
	SLC	0.2112	2	**0.1605**	31.60%	0.956	0.544	0.633
	MLC ALL	0.2112	200	***0.0971***	117.46%	0.954	0.706	0.802
CDS	MLC CFS	0.2338	6	**0.1200**	94.75%	0.846	0.642	0.707
	MLC PCA	0.2383	42	**0.1848**	28.93%	0.781	0.436	0.493
	MLC RELIEF	0.2121	36	**0.1132**	87.32%	0.952	0.657	0.759

The other two UCI datasets, namely D_{SPF} and D_{CDO} also exhibit some reductions of errors, though not as much as the one in D_{CDS}, within a couple of links.

The next observation is that classification using only NBC never outperforms our proposed ensembles, as shown in the second column titled NB-MAE in Table 2. It is desirable to see this and it shows the potential of our approach.

In terms of the number of links in the final trained CENBC, for D_{CDS}, we also observe that it takes more links to achieve greater reduction of errors, as shown in Figure 4, while for the other datasets, it take less number of links. So far, it appears that there is no rule of thumb as what the number of link for a dataset is in order to achieve a better classification accuracy of our ensemble. We believe that this number is highly problem-specific and we keep it in our future investigation.

In Figure 5 which is about the performance of our approach on D_{SDB}, we do not see the similar error reductions as in other datasets (therefore not included in Table 2). Further, we see that for the situation where MLC is in conjunction with CFS, there is an increase in MAE, and that for others (except MLC in conjunction with PCA), there is little error reduction. We need to investigate as to why D_{SDB} has this observation. We believe that it is probably more related to the dataset itself, where more characteristics of the dataset should be explored.

There are also some interesting observations. In Figures 3 and 4, for the variation SLC, we see some oscillations in its MAE. We have not found a plausible reason to explain this situation. On the other hand, we do not see any increase or decrease in MAE in Figures 2 and 5. It appears that the variation SLC require more investigation in our future work to understand why this behavior occurs.

6 Conclusion

In this paper, we present our on-going work on an ensemble of naive Bayes classifiers. Empirical experiments show the effectiveness of our approach. We observe significant error reductions on many datasets in our experiments. Furthermore, the number of links we need in our ensemble seems reasonable, making our approach practical with real-life tasks.

Our future plan is to include more real-life datasets in our experiments. More importantly, we need to conduct formal investigations to explore as why our proposed ensemble achieves such significant error reductions and, due to the simplicity of NBC, we hope to provide some performance guarantee of our ensemble through theoretical analysis. In addition to this, we observe that our ensemble runs extremely fast, usually with seconds or minutes. We also desire to conduct formal time complexity analysis on our approach.

References

1. H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

2. S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification : A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.
3. L. Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.
4. M. Buscema and S. Terzi W. Tastle. A new meta-classifier. In *North American Fuzzy Information Processing Society*, pages 1–7, 2010.
5. C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
6. D. R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 215–242, 1958.
7. T. G. Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.
8. M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. pages 359–366, 2000.
9. L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
10. K. Kira and L. A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, ML '92, pages 249–256, 1992.
11. I. Kononenko. Estimating Attributes: Analysis and Extensions of RELIEF. volume 784, pages 171–182, 1994.
12. M. Lichman. UCI machine learning repository, 2013.
13. R. Maclin and D. Opitz. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 1999.
14. Public Health Agency of Canada. Canadian chronic disease surveillance system summary 1999-2009, 01 2014.
15. F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
16. S. Russell and P. Norvig. A modern approach: Artificial intelligence. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25, 1995.
17. R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.
18. A. J. C. Sharkey. On combining artificial neural nets. *Connection Science*, 8:299–313, 1996.
19. K. Zaamout and J. Z. Zhang. Improving neural networks classification through chaining. In *The 22nd International Conference on Artificial Neural Networks*, pages 288–295, 2002.

A Metalearning Framework for Model Management

Mohammad Nozari Zarmehri and Carlos Soares

INESC Porto
Faculdade de Engenharia
Universidade do Porto
Rua Dr. Roberto Frias, 378, 4200–465
Porto, Portugal
{mohammad.nozari,csoares}@fe.up.pt

Abstract. The problem of selecting the best algorithm arises in a wide variety of situations. Organizations are more interested in having a specific model for distinct part of data instead of a single model for all data. From the business perspective, data can be divided naturally in different dimensions. This problem is getting worse when besides selecting the suitable algorithm, the selection of the best level of granularity is also involved. We propose a metalearning framework which recommends the best level of granularity in which, by applying a recommended algorithm by the framework, high performance is expected with high probability. The proposed framework is evaluated using two different datasets. The experiments show that the framework is very well suited for different problems including classification and regression problems.

1 Introduction

Traditionally, DM algorithms were applied at the global level and a single model is created for all data. For example, a single model is generated to predict the trip duration or to make sales predictions for all products. However, as more data is collected and the data characterizes objects at a finer level, there is a growing interest in more specific models, that represent groups or individual entities [11].

The first arisen question is how to split the data. We believe that Business Intelligence (BI) can be used for that purpose because a lot of effort has been invested into identifying the data dimensions that are relevant for the business (i.e. implemented as the data cubes). The second question is the granularity of the split. In BI the values of a dimension are organized hierarchically. The best models for a given subset of the data may be obtained by training with data from other, related subsets (e.g. if the amount of the data available for a given product is small, a more reliable model may be obtained by training with data from other products in the same category) [6].

In addition, a data hierarchy has been carefully designed by experts. The corresponding subsets are meaningful from a business perspective. Therefore,

subsets defined by Data Warehouse (DW) dimensions are generally expected to represent partitions of the data which may be useful to improve learning processes. However, since there are multiple levels, finding the best subset for learning a model by DM is a crucial task.

One solution can be metalearning [14]. It models the relationship between the characteristics of the data with the performance of the algorithms. It is often used to select the best algorithm for a specific problem, such as classification or regression. In this paper we address the problem of selecting the right level of granularity, as defined by DW dimensions, to model a DM problem. We use a metalearning approach, in which the characteristics of the data are mapped to the performance of the learning algorithms at different levels of granularity.

2 Background

In this section, we introduce the case studies and then we summarize the metalearning approaches.

2.1 Error Detection in Foreign Trade Statistics

Foreign trade statistics are important to describe the state of the economy of countries [9]. They are usually estimated by the different national statistics institutes based on data provided by companies. However, this data often contains errors because companies do not always appreciate the importance of providing accurate information. If undetected, these errors may, in some cases, have a significant impact on the value of the statistics. Therefore, national statistics institutes, such as the Portuguese Institute of Statistics (Instituto Nacional de Estatistica – INE), apply a combination of automatic and manual procedures to identify and correct those errors. Their goal is to detect as many errors as possible – to maximize the quality of the statistics – with as little manual effort as possible – to minimize the cost.

Some of the previous work on error detection have used outlier detection, classification and clustering approaches (e.g., [9, 6]). In general, satisfactory results have been obtained as some approaches were able to detect most of the erroneous transactions by choosing a small subset of suspicious transactions for manual inspection. However, this was not true for all products. This is partly due to the fact that some products have very few transactions. Given that each product is analyzed individually, the decision can be based on a very small set of data.

In [6], investigation of improvement of previous results by aggregating the data from different products based on the product taxonomy was done. The INE data contains the transactions for months 1, 2, 3, 5, 6, 8, 9, 10 in 1998 and months 1, 2 in 1999. The products are organized in a 4-levels taxonomy. An example of such a taxonomy can be: Food (Level 4), Bread (Level 3), Sliced bread (Level 2), Pack of 16 slices (Level 1). Each product is presented with a unique 8-digits product code (Level 1). Grouping the transactions at a higher

level of the product taxonomy may help obtaining better results when compared to an analysis at the product level (Level 1) itself, especially in the cases where the amount of data at this level is too small. According to previous work, the best results are obtained at different levels of the taxonomy for different products (Figure 1). For example, the best results for the products on the right leaf are obtained at the third level of product taxonomy while for the products at the middle leaf, the best results are obtained at the second level (black models in Figure 1). In spite of the fact that their results show that the aggregation is generally useful, they also show that the best results for different products are obtained at different levels of granularity.

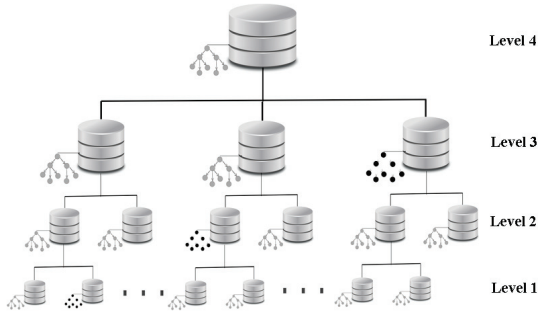


Fig. 1. Illustration of a hierarchy in datasets: for each category the best performance (black model) is obtained at different levels

2.2 Trip Duration

There has been a significant amount of research on trip duration prediction. Kwon et al. [5] use the flow and occupancy data from single loop detectors and historical trip duration information to forecast trip duration on a freeway. Using real traffic data, they found out that simple prediction methods can provide a good estimation of trip duration for trips starting in the near future (up to 20 minutes). On the other hand, for the trips starting more than 20 minutes away, better predictions can be obtained with historical data. The same approach is used by Chien et al. [3]. Zhang et al. [16] propose a linear model to predict the short-term freeway trip duration. In their model, trip duration is a function of departure time. Their results show that for a small dataset, the error varies from 5% to 10% while for a bigger dataset, the variation is between 8% and 13%.

Support Vector Regression (SVR) is used for prediction of trip duration by Wu et al. [15]. They utilize real highway traffic data for their experiments. They

suggest a set of SVR parameter values by trial-and-error which lead to a model that is able to outperform a base-line model. Balan et al. [1] propose a real-time information system that provides the expected fare and trip duration for passengers. They use historical data consisting of approximately 250 million paid taxi trips for the experiment.

Considering the rapid change of behavior of vehicular networks, using the same algorithm for forecasting the travel time over a long period and for different vehicles, will eventually end in unreliable predictions. Therefore, it is important to find the best algorithm for each context. One possibility is to use a trial-and-error approach. This approach would be very time consuming, given the amount of alternatives available. One alternative approach is metalearning which is still missing.

2.3 Metalearning

The algorithm selection problem was formally defined by Rice in 1976 [7]. The main question was to predict which algorithm has the best performance for a specific problem. The first formal project in this area was MLT project [4]. The MLT project creates a system called *Consultant-2* which can help to select the best algorithm for a specific problem. Over the years, metalearning research has addressed several issues [8]. It may be important to select the best base-level algorithm not for the whole dataset, but rather for a subset of the examples [2] or even for individual examples [12]. Tuning the parameters of base-level algorithms is another task that metalearning can be helpful to (e.g. the kernel width of SVM with Gaussian kernel [10, 8]. Rijn et al. [13] have investigated the use of metalearning for algorithm selection on data streams. The metafeatures are calculated on a small data window at the start of the data stream. Metalearning uses this metafeatures to predict which algorithm is the best in the next data windows.

3 Methodology

3.1 Database

Traditional method: Suppose the available data consists of n_1 entities, $\{E_i, \forall i \in \{1, ..., n_1\}\}$. In traditional data mining scheme, each entity E_i has some associated features, C_i, and there is a target variable, Y_i. So the dataset used for the traditional data mining is like $DB = \{E_i, C_i, Y_i\}, \forall i \in \{1, ..., n_1\}$ while C_i is a vector of features. So the traditional scheme is unidirectional scheme.

Our metalearning method: The possibility of categorizing entities at upper level adds another dimension to the dataset. So for each entity, instead of having just one vector of features (C_i), there are more features at different levels, $C_1^1, C_1^2, C_1^3, ..., C_1^k$, where k is the number of existing levels or categories. So the dataset for using in the data mining process is $DB = \{E_i, C_i^j, Y_i\}, \forall i \in$

$\{1, ..., n_1\}, \forall j \in \{1, ..., k\}$. In general, C_i^j is the features for entity i at level j. In addition, the number of entities at higher levels (bigger j) is higher:

$$w < v \iff L^w < L^v \tag{1}$$

According to Formula 1, for 2 different levels w and v where $w < v$, the number of entities in level w, L^w, is less than number of entities in level v, L^v. The proposed model used in this article is shown in Figure 2. At the lowest level,

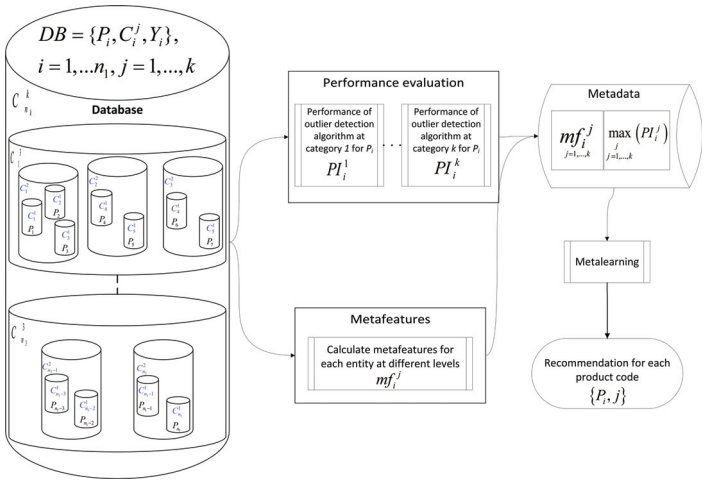


Fig. 2. Proposed methodology used for Metalearning

level 1, each entity creates a unique category, $C_i^1, \forall i \in \{1, ..., n_1\}$. But at the higher levels, levels $(2, 3, ..., k)$, several entities join to create a category. For example, category 1 at level 2 (C_1^2) consists of 3 different entities: E_1, E_2, and E_3, while category 2 in the same level (C_2^2) consists of 2 different entities: E_4 and E_5. So the number of categories at the levels 1, 2, 3, ..., and k are n_1, n_2, n_3, ..., and n_k, respectively. In this step, the $DB = \{E_i, C_i^j, Y_i\}, \forall i \in \{1, ..., n_1\}, \forall j \in \{1, ..., k\}$ is delivered to the learning process. In the model, there are g algorithms. Having different algorithms and different levels for each entity, each algorithm will be evaluated for each entity at each level. As result, for each entity, there are different performance indicators: P_{i1}^1, ..., P_{i1}^k, where P_{ig}^k means the performance of the algorithm g at level k for entity i. On the other hand, the metafeatures calculation for DB is made in the other side of the model. The metafeatures are calculated for each entity and at different levels. In general mf_i^j is the calculated metafeatures for entity i at the level j.

3.2 Metadata

The dataset used for metalearning is called metadata. For each entity, the best performance obtained from the performance evaluation part is selected according to the Eq. 2:

$$P_{best\,i} = \max_{w,j}(P_{iw}^{j}) \ , \ \forall w \in \{1, ..., g\}, \forall j \in \{1, ..., k\} \tag{2}$$

So the metadata for each entity is consisted as metafeatures for different levels plus the best performance obtained from the Eq. 2. As an example Eq. 3 shows the general form of the metadata which is used for metalearning:

$$Row\ i \rightarrow E_i,\ mf_i^1,\ mf_i^2,\ mf_i^3, \ldots,\ mf_i^k,\ P_{best\,i} \tag{3}$$

Therefore, the metadata has n_1 rows which is equal to the number of entities. The main idea in metalearning is to find out the best algorithm and the best level to apply the algorithm depending on the metafeatures obtained at different levels. Consequently, the metalearning maps the extracted features from the original datasets to the best performance obtained at different levels by applying different algorithms on the original dataset. Our model recommends a level and an algorithm for each entity in which, applying the recommended algorithm on the recommended level produces the best performance with high probability (see Eq. 4).

$$Output\ :\ \{\ \underbrace{E_i}_{entity}\ ,\ \underbrace{j}_{recommended\,level}\ ,\ \underbrace{g}_{recommended\,algorithm}\ \} \tag{4}$$

4 Evaluation

In this section, the proposed framework is evaluated by two different case studies: INTRASTATS dataset (Section 4.1) and VANETs dataset (Section 4.2).

4.1 Case Study 1: INTRASTATS Dataset

The dataset obtained from foreign trade statistics (Section 2.1) is used for this case study.

Methodology For this case study, our model is reduced to just predict the best level of hierarchy for a given algorithm. To adopt our model, each product code is selected as an entity. Then the DB is equal to $\{E_i, C_i^j, Y_i\} = \{P_i, C_i^j, Y_i\}, \forall i \in \{1, ..., n_1\}$ where n_1 is the number of unique products. In performance evaluation block in our model, the first line is only relevant for this scenario. So the performance of only outlier detection algorithm is evaluated for each product at different levels ($P_{i1}^1 \rightarrow P_{i1}^k$) where k is the number of different levels in this case. Then the best performance is calculated according to the Eq. 5:

$$P_{best\,i} = \max_{j}(P_{i1}^{j}) \ , \ \forall j \in \{1, ..., k\} \tag{5}$$

The metadata is then created using the best performance obtained from the performance evaluation block plus extracted metafeatures from the original datasets. An example of the metadata is: P_i, mf_i^1, mf_i^2, $mf_i^3, \ldots,$ mf_i^k, P_{best_i}.

Level Prediction In this section, only the prediction of the best level is evaluated. In our model, two algorithms are applied on metadata: Decision Tree (ML-Tree) and Random Forest (ML-RF). Figure 3 shows the comparison of two metalearning approaches with the baseline accuracy. It is clear that the random forest model applied on metadata, is outperformed the base-line. But the accuracy of the decision tree is not as well as the accuracy of the random forest model. Although, the decision tree shows better results than the baseline in the last month. Instead of applying the selected algorithm on several levels and

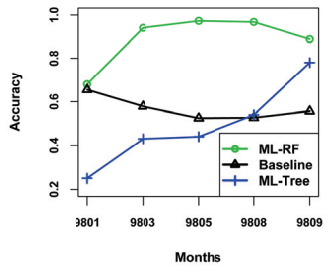


Fig. 3. Comparing metalearning approaches with the baseline: accuracy

compare their performance to find the best one, we just need to calculate the metafeatures and apply the outlier detection algorithm to find the proper level. In the Section 4.2, a complete evaluation to find both the proper algorithm and associated level is investigated.

4.2 Case Study 2: Taxi Dataset

In this section, our model is completely evaluated by recommending both the algorithm and the level which are proper for trip duration prediction for a given taxi. Instead of classification task in Section 4.1, here the task is a regression problem.

Dataset The dataset is obtained from a realistic and a large-scale scenario. The scenario is the city of Porto, which is the second largest city in Portugal, sum

up an area of 41.3 km^2. There are 63 taxi stands in the city and the main taxi union has 441 vehicles. Each taxi has an on-board unit with the GPS receiver and collect the travel log. The dataset is consist of five months in 2013 for all the vehicles. The dataset related to one month has 13 variables. The objective in this study is to predict the best level of hierarchically to apply the best algorithm to predict the trip duration.

Methodology According to our model, the entity (E_i) is replaced by a taxi (T_i). Having near 440 taxis in the city, the total number of unique entities in the first level in our model is 440 (C_i^1, $\forall i \in \{1, ..., 440\}$). In this study, the total number of levels are considered 2 levels: taxi itself (level 1) and the whole data in a month (level 2). So the dataset which is delivered to performance evaluation and metafeatures extraction blocks, in our model (Section 3) is $DB = \{E_i, C_i^j, Y_i\} = \{T_i, C_i^j, Y_i\}, \forall i \in \{1, ..., 440\}, \forall j \in \{1, 2\}$. The algorithm space contains 4 algorithms: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and Linear regression (LM). Each algorithm is evaluated for each taxi at 2 different levels Eq. 6.

$$P_{iw}^j : \ \forall w \in \{1, \ldots, 4\}, \forall j \in \{1, 2\}, \forall i \in \{1, \ldots, 440\} \tag{6}$$

And the best performance among the P_{iw}^j is selected according to the Eq. 7.

$$P_{best\,i} = \underset{w,j}{max}(P_{iw}^j) \ , \ \forall w \in \{1, \ldots, 4\}, \forall j \in \{1, 2\}, \forall i \in \{1, \ldots, 440\} \tag{7}$$

Finally the metadata structure is consist of the taxi identification, metafeatures for the first and the second level and the best performance obtained from performance evaluation block (see Eq. 8).

$$T_i, \ mf_i^1, \ mf_i^2, \ P_{best\,i} \tag{8}$$

Level Prediction The first analysis is done for just predicting the best level for each unseen observation and a given algorithm. So a specific algorithm is selected. Then by using our model, the best level for applying the algorithm to have the best prediction of trip duration is recommended. Figure 4 shows the results of this analysis. When the SVM algorithm is selected, the recommended level is more accurate than other algorithms. The accuracy in this case is 91% on average which is 7% more than the LM algorithm.

Level and Algorithm Prediction In this section, the complete result of simultaneously predicting both an algorithm and a level is analyzed. In Figure 5, the gray bars show the algorithm with the best performance at the base-level (BL). The prediction at meta-level (ML) is represented by blue bars (darker bars). Each algorithm is evaluated in two levels, for example RF algorithm is applied at level 1 (rf1) and level 2 (rf2). This plot shows that the algorithm with the best performance at the base-level is not always the same and can be varied. In addition, metalearning almost follows the base-line prediction without doing analysis at the base-level.

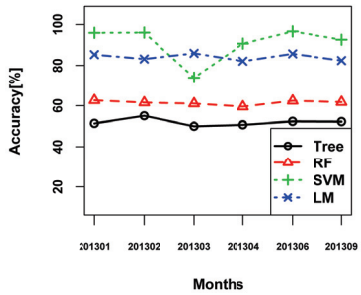


Fig. 4. Accuracy of recommendation of the level for an individual algorithm

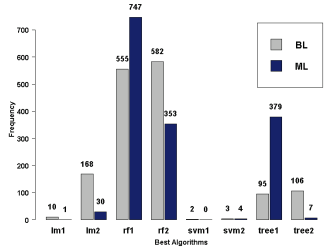


Fig. 5. The best algorithm and associated level

5 Conclusion

Metalearning approaches aim at assisting users to select appropriate learning algorithm for the particular data mining task. This problem is even worse when considering the existing hierarchy in the datasets. In this paper, we proposed a new metalearning framework to predict the best level of granularity to apply the recommended algorithm. The basic idea is to reduce the computational costs for applying different algorithms at different levels of granularity to reach the best performance. Our model recommends an algorithm and a level of granularity to obtain the best performance with high probability. The proposed model has been applied on different datasets: Statistical dataset and Taxi dataset. Extensive experimental results have illustrated the improvement of accuracy of metalearning approaches comparing to the base-line for both case studies.

References

1. Balan, R.K., Nguyen, K.X., Jiang, L.: Real-time trip information service for a large taxi fleet. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. pp. 99–112. MobiSys '11, ACM, New York, NY, USA (2011)
2. Brodley, C.: Recursive automatic bias selection for classifier construction. Machine Learning 20(1-2), 63–94 (1995)
3. Chien, S.I.J., Kuchipudi, C.M.: Dynamic travel time prediction with real-time and historic data. Journal of transportation engineering 129(6), 608–616 (2003)
4. Kodratoff, Y., Sleeman, D., Uszynski, M., Causse, K., Craw, S.: Building a machine learning toolbox (1992)
5. Kwon, J., Coifman, B., Bickel, P.: Day-to-day travel-time trends and travel-time prediction from loop-detector data. Transportation Research Record: Journal of the Transportation Research Board 1717(1), 120–129 (2000)
6. Nozari Zarmehri, M., Soares, C.: Improving Data Mining Results by taking Advantage of the Data Warehouse Dimensions: A Case Study in Outlier Detection. In: Encontro Nacional de Inteligência Artificial e Computacional. UFMG, LBD, São Carlos, Brazil (2014)
7. Rice, J.R.: The algorithm selection problem. Advances in Computers, vol. 15, pp. 65 – 118. Elsevier (1976)
8. Rossi, A.L.D., de Leon Ferreira de Carvalho, A.C.P., Soares, C., de Souza, B.F.: MetaStream: A meta-learning based method for periodic algorithm selection in time-changing data. Neurocomputing 127(0), 52–64 (2014)
9. Soares, C., Brazdil, P., Costa, J., Cortez, V., Carvalho, A.: Error Detection in Foreign Trade Data using Statistical and Machine Learning Algorithms. In: Mackin, N. (ed.) Proceedings of the 3rd International Conference and Exhibition on the Practical Application of Knowledge Discovery and Data Mining. pp. 183–188. London, UK (1999)
10. Soares, C., Brazdil, P.B., Kuba, P.: A meta-learning method to select the kernel width in support vector regression. Machine learning 54(3), 195–209 (2004)
11. Soulié-Fogelman, F.: Data Mining in the real world: What do we need and what do we have? In: Ghani, R., Soares, C. (eds.) Proceedings of the KDD Workshop on Data Mining for Business Applications. pp. 44–48 (2006)
12. Todorovski, L., Džeroski, S.: Combining classifiers with meta decision trees. Machine learning 50(3), 223–249 (2003)
13. van Rijn, JanN. and Holmes, Geoffrey and Pfahringer, Bernhard and Vanschoren, J.: Algorithm Selection on Data Streams. Lecture Notes in Computer Science, Springer International Publishing 8777, 325–336 (2014)
14. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial Intelligence Review (1997), 77–95 (2002)
15. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-time prediction with support vector regression. Intelligent Transportation Systems, IEEE Transactions on 5(4), 276–281 (2004)
16. Zhang, X., Rice, J.A.: Short-term travel time prediction. Transportation Research Part C: Emerging Technologies 11(3), 187–210 (2003)

Passive-Agressive bounds in bandit feedback classification

Hongliang Zhong[1] and Emmanuel Daucé[2]

1- Laboratoire d'Informatique Fondamentale-CNRS UMR 7279
Aix-Marseille Université, Ecole Centrale Marseille, France
hongliang.zhong@lif.univ-mrs.fr
2- Institut de Neurosciences des Systèmes-INSERM UMR 1106
Aix-Marseille Université, Ecole Centrale Marseille, France
emmanuel.dauce@ec-marseille.fr

Abstract. This paper presents a new online multiclass algorithm with bandit feedback, where, after making a prediction, the learning algorithm receives only partial feedback, i.e., the prediction is correct or not, rather than the true label. This algorithm, named Bandit Passive-Aggressive online algorithm (BPA), is based on the Passive-Aggressive Online algorithm (PA) proposed by [2], the latter being an effective framework for performing max-margin online learning. We analyze some of its operating principles, and we also derive a competitive cumulative mistake bound for this algorithm. Further experimental evaluation on several multiclass data sets, including three real world and two synthetic data sets, shows interesting performance in the high-dimentional and high label cardinality case.

1 Introduction

Online learning is an effective way to deal with large scale applications, especially applications with streaming data. Algorithm PA provides a generic framework for online large-margin learning, with many applications[4, 5]. PA uses hypotheses from a set of linear predictors. But it only works in the conventional supervised learning paradigm, in which, the learner has access to the true labels of data after making its prediction. In contrast, there is an other partially supervised learning problem: the multiclass prediction with bandit feedback[1]. Unlike the conventional supervised learning paradigm, it focuses on applications where the learner only receives bandit feedback. "Bandit feedback" means partial feedback: the learner only receives "correct" or "not correct" about its prediction. As we know, full information is rarely revealed in the real world. So bandit feedback could apply to lots of domains, including many web-based applications, such as an online recommender system as mentioned by [1]. It is said that when user makes a query to a recommender system, the system gives a suggestion under its former knowledge about the user; then this user responds to the suggestion by either clicking or not clicking it. Nevertheless the system does not know what would happen if it would provide other suggestions as substitutions. Essentially,

we formalize the problem as follows: the learning algorithm gets an input feature vector x_t at each round t; then based on the obtained information from the former round, it makes a prediction and assigns a label $\hat{y}_t$ to the input; finally, according to its prediction and the true label of the input x_t, it receives a partial feedback telling whether its prediction was correct or not. In contrast, conventional online supervised learning would disclose the true label y_t to the user at each consecutive round. So, with bandit feedback, this kind of problems is harder than conventional supervised learning problems.

Related work. Several classification algorithms exist that address the bandit feedback setting. Banditron [1], based on the Perceptron algorithm, is the most "classical" one, having a number of mistakes asymptotically bounded. For the case where the data is linearly separable, the number of mistakes is bounded as $O(\sqrt{T})$ in T rounds, and has a $O(T^{2/3})$ regret in the non-linearly separable case. To handle the difficulties of utilizing the negative feedback, Banditron uses an exploitation-exploration scheme. In some exploratory rounds, it makes a prediction uniformly with probability $\mathbb{P}(\tilde{Y} = i|\hat{y}_t)$ from the full set of labels instead of choosing the most probable label given the current learner belief.

Another bandit algorithm, named "Confidit", was proposed by [3]. This algorithm trades off exploration and exploitation via upper-confidence bounds, in a way that is somewhat similar to the work of [12]. In confidit approach, the bound of regret is improved from $O(T^{2/3})$ to $O(\sqrt{T}\log T)$.

In this paper, we discuss a new algorithm: Bandit Passive-Aggressive Online algorithm(BPA), i.e., we adapt PA approach[3] to the bandit setting. With PA's advantage, BPA should in principle perform a max-margin with bandit feedback.

In next sections, we will discuss this new bandit algorithm, including its update rules, and we provide some experiments to compare the cumulative loss on two synthetic and three real-world data sets.

2 Preliminaries

Online learning is applied in a sequence of consecutive rounds. On round t, the learner is given an instance vector $x_t \in \mathbb{R}^d$ and is required to predict a label out of a set of multiclass $[k] = \{1, ..., k\}$. We denote by $\hat{y}_t$ the predicted label. In the general setting, after its prediction, it receives a correct label associated with x_t, which we denote by $y_t \in [k]$. In the bandit setting, the feedback contains a partial information $\delta_{(\hat{y}_t = y_t)}$, where $\delta_{(\hat{y}_t = y_t)}$ is 1 if $\hat{y}_t = y_t$, and 0 otherwise.

The prediction at round t is chosen by a hypothesis $h_t : \mathbb{R}^d \to [k]$, where h_t is taken from a class of hypothesis $\mathbb{H}$ parameterized by a $k \times d$ matrix of real weight w, and is defined to be:

$$\hat{y}_t = h_t(x_t) = \underset{i \in [k]}{argmax} < w_i, x_t > \tag{1}$$

where $w_i \in \mathbb{R}^d$ is the i^{th} row of the matrix $w \in \mathbb{R}^{k*d}$.

Consistently with [3]'s writing, a feature function: $\Phi(x, i)$ is a $k \times d$ matrix which is composed of k features vectors of size d. All rows of $\Phi(x, i)$ are zero ex-

cept the i^{th} row which is set to x_t. It can be remarked that $< \Phi(x,i), \Phi(x,j) >= \| x \|^2$ if $i = j$ and 0 otherwise.

3 The algorithm Passive-Aggressive with bandit feedback

In this section, we introduce a new online learning algorithm, which is a variant of the Passive-Aggressive Online algorithm adapted to the bandit setting.

3.1 Passive-Aggressive Online learning

The goal of online learning is to minimize the cumulative loss for a certain prediction task from the sequentially arriving training samples. PA achieves this goal by updating some parameterized model w in an online manner with the instantaneous losses from arriving data $x_{t,t\geqslant 0}$ and corresponding responses $y_{t,t\geqslant 0}$. The losses $l(w;(x_t,y_t))$ can be the hinge loss. The update of PA derives its solution from an optimization problem:

$$w_{t+1} = \underset{w\in\mathbb{R}^{k\times d}}{argmin} \frac{1}{2} \| w - w_t \|^2 \; s.t. \; l(w;(x_t,y_t)) = 0 \tag{2}$$

Namely, each instance x_t is associated with a single correct label $y_t \in \mathbb{Y}$ and the prediction $\hat{y}_t$ extends by Eq. 1. A prediction mistake occurs if $y_t \neq \hat{y}_t$. The update w of PA in Eq. 2 has the closed form solution,

$$w_{t+1} = w_t + \frac{l(w_t;(x_t,y_t))}{\| \Phi(x_t,y_t) - \Phi(x_t,\hat{y}_t) \|^2} (\Phi(x_t,y_t) - \Phi(x_t,\hat{y}_t)) \tag{3}$$

Intuitively, if w_t suffers no loss from new data, i.e., $l(w_t;(x_t,y_t)) = 0$, the algorithm passively assigns $w_{t+1} = w_t$; otherwise, it aggressively projects w_t to the feasible zone of parameter vectors that attain zero loss.

3.2 Passive-Aggressive algorithm in bandit setting

We now present BPA in Algorithm 1, which is an adaptation of PA for the bandit case. Similar to PA, at each round it outputs a prediction $\hat{y}_t$ to be the label with the highest score of $\langle w_i, x_t \rangle$, to make a reference to Eq. 1. Unlike the conventional learning paradigm, if $\hat{y}_t \neq y_t$, it is difficult to get a PA update because the true label's information is not supported. So we need to perform an exploration, i.e. sample a label randomly $[k]$ with parameter γ and contrast this random prediction with a bandit return $\delta_{(\tilde{y}_t = y_t)}$, where $\tilde{y}_t$ is the result of a random draw from a certain distribution $\mathbb{P}(\tilde{Y}|\hat{y}_t)$:

$$\mathbb{P}(\tilde{Y} = i|\hat{y}_t) = \delta_{(i=\hat{y}_t)} \cdot (1-\gamma) + \frac{\gamma}{k} \tag{4}$$

The above intuitive argument is formalized by defining the update matrix U_t to be a function of the random prediction $\tilde{y}_t$.

We redefine the instantaneous loss by the following function,

$$l_t = [1 + (1 - 2\delta_{(\tilde{y}_t = y_t)}) \cdot \langle w, \Phi(x_t, \tilde{y}_t)\rangle]_+ \tag{5}$$

with $(1-2\delta_{(\tilde{y}_t=y_t)})$ equal to -1 when $\tilde{y} = y$ and 1 elsewhere. This loss is the standard hinge loss $[1 - \langle w, \Phi(x_t, \tilde{y}_t)\rangle]_+$ when the prediction is correct: it stays at 0 for $\langle w, \Phi(x_t, \tilde{y}_t)\rangle \geq 1$ and then increases for decreasing values of $\langle w, \Phi(x_t, \tilde{y}_t)\rangle$. In contrast, when the prediction is incorrect, the loss is equal to $[1+\langle w, \Phi(x_t, \tilde{y}_t)\rangle]_+$, i.e. stays at 0 for $\cdot\langle w, \Phi(x_t, \tilde{y}_t)\rangle < -1$ and then increases for increasing values of $\langle w, \Phi(x_t, \tilde{y}_t)\rangle$.

The linear classifiers are updated at each trial using the standard tools from convex analysis [6]. If $l_t = 0$, w_t satisfies the constraint in Eq. 2 , otherwise it should to satisfy the constraint optimization problem defined in Eq. 2 by the Lagrangian,

$$L(w, \tau) = \frac{1}{2} \parallel w - w_t \parallel^2 + \tau([1 + (1 - 2\delta_{(\tilde{y}_t = y_t)}) \cdot \langle w, \Phi(x_t, \tilde{y}_t)\rangle]_+) \tag{6}$$

$$w = w_t + \tau \cdot (2\delta_{(\tilde{y}_t = y_t)} - 1)\Phi(x_t, \tilde{y}_t)$$

Taking the derivative of $L(\tau)$ with respect to τ and also setting it to zero, we get that:

$$\tau = \frac{l_t}{\parallel \Phi(x_t, \tilde{y}_t) \parallel^2}$$

$$\Rightarrow w = w_t + (2\delta_{(\tilde{y}_t = y_t)} - 1) \cdot \tau \cdot \Phi(x_t, \tilde{y}_t) \tag{7}$$

Algorithm 1 The Bandit Passive-Aggressive online learning

Require: $w_1 = \mathbf{0} \in \mathbb{R}^{k \times d}$
1: **for** each $t = 1, 2, ..., T$ **do**
2: Receive $x_t \in \mathbb{R}^d$;
3: Set $\hat{y}_t = \underset{r \in [k]}{argmax}(< w_t, \Phi(x_t, r) >)$
4: **for all** $i \in [k]$ **do**
5: $\mathbb{P}(\tilde{Y} = i|\hat{y}_t) = (1 - \gamma) \cdot \delta_{(i=\hat{y}_t)} + \frac{\gamma}{k}$;
6: **end for**;
7: draw $\tilde{y}_t$ randomly
8: Receive the feedback $\delta_{(\tilde{y}_t = y_t)}$
9: $l_t = [1 + (1 - 2\delta_{(\tilde{y}_t = y_t)}) \cdot < w_t, \Phi(x_t, \tilde{y}_t) >]_+$
10: Update: $w_{t+1} = w_t + (2\delta_{(\tilde{y}_t = y_t)} - 1) \cdot \frac{l_t}{\|\Phi(x_t, \tilde{y}_t)\|^2} \cdot \Phi(x_t, \tilde{y}_t)$
11: **end for**

Considering for instance the common phenomenon of label noise, a mislabeled example may cause PA to drastically change its classifiers in the wrong direction. To derive soft-margin classifiers [13] and a non-negative slack variable

ξ is introduced into the optimization problem in Eq. 2. Accordingly with [2], the variable can be introduced in two different ways.

$$\begin{cases} w_{t+1} = \underset{w\in\mathbb{R}^{k\times d}}{argmin} \frac{1}{2} \parallel w - w_t \parallel^2 + C\xi \;\; s.t. \;\; l(w;(x_t,y_t)) \leqslant \xi \; and \; \xi \geqslant 0 \\ w_{t+1} = \underset{w\in\mathbb{R}^{k\times d}}{argmin} \frac{1}{2} \parallel w - w_t \parallel^2 + C\xi^2 \;\; s.t. \;\; l(w;(x_t,y_t)) \leqslant \xi \end{cases} \tag{8}$$

By these optimization problems, we get the corresponding optimization solutions:

$$\begin{cases} w_{t+1} = w_t + (2\delta_{(\tilde{y}_t=y_t)} - 1) \cdot min\left\{C, \frac{l_t}{\|\Phi(x_t,\tilde{y}_t)\|^2}\right\} \cdot \Phi(x_t,\tilde{y}_t) \\ w_{t+1} = w_t + (2\delta_{(\tilde{y}_t=y_t)} - 1) \cdot \frac{l_t}{\|\Phi(x_t,\tilde{y}_t)\|^2 + \frac{1}{2C}} \cdot \Phi(x_t,\tilde{y}_t) \end{cases}$$

4 Analysis

In this section, we prove the cumulative squared loss has a upper bound. To simplify, we note $l(w_t;(x_t,y_t))$ as l_t and $l(u;(x_t,y_t))$ as l_t^*.

Theorem 1. *Let $(x_1,y_1),...,(x_T,y_T)$ be a sequence of separable examples where $x_t \in \mathbb{R}^d$, $y_t \in [k]$ and $\parallel x_t \parallel \leqslant R$ for all t, and $u \in \mathbb{R}^{k\times d}$. Then, the cumulative squared loss of this algorithm is bounded by,*

$$\sum_{t=1}^{T} l_t^2 \leqslant R^2 \cdot \parallel u \parallel^2 \tag{9}$$

Proof. Define Δ_t to be:

$$\Delta_t = \parallel w_t - u \parallel^2 - \parallel w_{t+1} - u \parallel^2$$

By summing Δ_t over all t from 1 to T, that $\sum_t \Delta_t$ is a telescopic sum which collapses to,

$$\sum_{t=1}^{T} \Delta_t = \sum_{t=1}^{T} \left(\parallel w_t - u \parallel^2 - \parallel w_{t+1} - u \parallel^2 \right) = \parallel w_1 - u \parallel^2 - \parallel w_{t+1} - u \parallel^2$$

By the initiation of $w_1 = \mathbf{0}$,

$$\sum_{t=1}^{T} \Delta_t = \parallel u \parallel^2 - \parallel w_{t+1} - u \parallel^2 \leqslant \parallel u \parallel^2 \tag{10}$$

Using the definition of update in Eq.7,

$$\Delta_t = -2\left\langle (w_t - u), (2\delta - 1)\frac{l_t}{\parallel \Phi(x_t,\tilde{y}_t) \parallel^2}\Phi(x_t,\tilde{y}_t)\right\rangle - \left(\frac{l_t}{\parallel \Phi(x_t,\tilde{y}_t) \parallel^2}\Phi(x_t,\tilde{y}_t)\right)^2$$

With $l_t = [1 + (1 - 2\delta_{(\tilde{y}_t = y_t)}) \cdot \langle w_t, \Phi(x_t, \tilde{y}_t)\rangle]_+$ and $l_t^* = [1 + (1 - 2\delta_{(\tilde{y}_t = y_t)}) \cdot \langle w^*, \Phi(x_t, \tilde{y}_t)\rangle]_+$, So,

$$\Delta_t = 2\frac{l_t^2 - l_t l_t^*}{\parallel \Phi(x_t, \tilde{y}_t) \parallel^2} - \left(\frac{l_t}{\parallel \Phi(x_t, \tilde{y}_t) \parallel^2} \parallel \Phi(x_t, \tilde{y}_t) \parallel\right)^2$$

$$\Delta_t = \frac{l_t^2 - 2l_t l_t^*}{\Phi(x_t, \tilde{y}_t) \parallel^2}$$

If all examples are separable, $\exists u$ such that $\forall t \in [1, ..., T]$, $l_t^* = 0$, following the Eq. 10,

$$\Rightarrow \parallel u \parallel^2 \geqslant \sum_{t=1}^{T} \Delta_t \geqslant \sum_{t=1}^{T} \left(\frac{l_t^2}{\parallel \Phi(x_t, \tilde{y}_t) \parallel^2}\right)$$

$$\Rightarrow \sum_{t=1}^{T} l_t^2 \leqslant \parallel u \parallel^2 \cdot \parallel \Phi(x_t, \tilde{y}_t) \parallel^2$$

$$\sum_{t=1}^{T} l_t^2 \leqslant R^2 \cdot \parallel u \parallel^2$$

Theorem 2. *Let $(x_1, y_1), ..., (x_T, y_T)$ be a sequence of non-separable examples where $x_t \in \mathbb{R}^d$, $y_t \in [k]$ and $\parallel x_t \parallel \leqslant R$ for all t. Then for any vector $u \in \mathbb{R}^{k \times d}$ the cumulative squared loss of this algorithm is bounded by:*

$$\sum_{t=1}^{T} l_t^2 \leqslant \left(R \parallel u \parallel + 2\sqrt{\sum_{t=1}^{T} (l_t^*)^2}\right)^2$$

Proof. By the proof of Theorem 1,

$$\sum_{t=1}^{T} l_t^2 \leqslant R^2 \cdot \parallel u \parallel^2 + 2\sum_{t=1}^{T} l_t l_t^*$$

To upper bound the right side of the above inequality, and denotes $L_t = \sqrt{\sum_{t=1}^{T} l_t^2}$ and $U_t = \sqrt{\sum_{t=1}^{T} (l_t^*)^2}$,

$$2(L_t U_t)^2 - 2(\sum_{t=1}^{T} l_t l_t^*)^2 = \sum_{i=1}^{T}\sum_{j=1}^{T} l_i^2 (l_j^*)^2 + \sum_{i=1}^{T}\sum_{j=1}^{T} l_j^2 (l_i^*)^2 - 2\sum_{i=1}^{T}\sum_{j=1}^{T} l_i l_j l_i^* l_j^*$$

$$= \sum_{i=1}^{T}\sum_{j=1}^{T} (l_i l_j^* - l_j l_i^*)^2 \geqslant 0$$

$$\sum_{t=1}^{T} l_t^2 \leqslant R^2 \cdot \parallel u \parallel^2 + 2\sum_{t=1}^{T} l_t l_t^* \leqslant R^2 \cdot \parallel u \parallel^2 + 2L_t U_t$$

$$L_t \leqslant U_t + \sqrt{R^2 \parallel u \parallel^2 + U_t^2}$$

Using the fact that $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$,

$$L_t \leqslant R \parallel u \parallel + 2U_t$$

$$\sum_{t=1}^{T} l_t^2 \leqslant \left(R \parallel u \parallel + 2\sqrt{\sum_{t=1}^{T} (l_t^*)^2} \right)^2$$

5 Experiments

In this section, we evaluate our algorithm with experimental results on three bandit algorithms and two supervised algorithms over two synthetic and three real world data sets. Their characteristics are summarized in Table 1. The cumulative loss is presented for each data sets.

5.1 Data sets

The first data set, denoted by SynSep, is a 9-class, 400-dimensional synthetic data set of size 10^5. More details about the method to generate this data set can be found in [1]. The SynSep idea is to have a simple simulation of generating a text document. The coordinates represent different words in a small vocabulary of size 400. We ensure that SynSep is linearly separable.

The second data set, denoted by SynNonSep, is constructed the same way as SynSep except that a 5% label noise is introduced, which makes the data set non-separable.

The third data set is collected from the Reuters RCV1-v2 collection[7]. The original data set is composed by multi-label instances. So we make some preprocessing likes [8]. First, its label hierarchy is reorganized by mapping the data set to the second level of RCV1 topic hierarchy. The documents that have labels of the third or forth level only are mapped to their parent category of the second level; Second, all multi-labelled instances have been removed. This RCV1-v2 is a 53-class, 47236-dimensional real data set of size 10^5.

Table 1. Summary of the five data sets, including the numbers of instances, features, labels and whether the number of examples in each class are balanced.

Data	Instances	Features	Labels	Balanced
SynSep	100 000	400	9	Y
SynNonSep	100 000	400	9	Y
RCV1-v2	100 000	47236	53	N
Letter	20 000	16	26	N
Pen-Based	13 200	16	10	N

The fourth and fifth data sets are collected from [9, 10]. The fourth data set is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20000 unique stimuli. Each stimuli was converted into 16 primitive numerical attributes (statistical moments and edge counts). It forms a 26-class, 16-dimensional real data set of size 20000. The fifth data set is a digit data base made by collecting 250 samples from 44 writers, using only (x,y) coordinate information represented as constant length feature vectors, which were resampled to 8 points per digit (therefore the data set contains 8 points $\times$ 2 coordinates = 16 features). This one is a 10-class, 16-dimensional real data set of size 10992.

5.2 Algorithms

Five algorithms are evaluated:

Perceptron[11] and PA[2], they work in the full information setting and no parameters are needed.

Banditron[1] working in bandit feedback, its simulations are run for different γ values from interval $[0.01, 0.99]$ to determine the best value for each data set.

Confidit[3] working in bandit feedback, to simplify the computing process, we replaced the multiplier of $x_t^T A_{i,t-1}^{-1} x_t$ in the definition of $\epsilon_{i,t}^2$ (see [3]) with some constant η.

Our algorithm, BPA works in bandit feedback, different simulations are run to choose the best γ value for each data, set like Banditron.

5.3 Results

Figures 1 and 2 show the experimental results on two synthetic data sets and three real data sets. For SynSep, a separable linear data set, all algorithms except Banditron obtain a good performance; with the non-separable SynNonSep data, Confidit and BPA outperform the other algorithms, even the algorithms having a full feedback.

With the three real data sets, the algorithms with full information, despite their competitive advantage with respect to the ones with bandit feedback, do not significantly depart from BPA and Confidit, with classification results that clearly outperform Banditron. While having a lower computational complexity, BPA approach is even found to outperform Confidit in the most challenging situation, i.e. the high-dimensional case with a large number of classes (RCV1-v2 data set).

The γ parameter represents the exploration rate in Banditron and BPA algorithms. We compare on Figure 3 the average error rates obtained on the two algorithms for different values of γ on the different data sets. In contrast with Banditron, BPA shows that γ has a very little influence on the final error rate, indicating a capability to deal with small exploration rates.

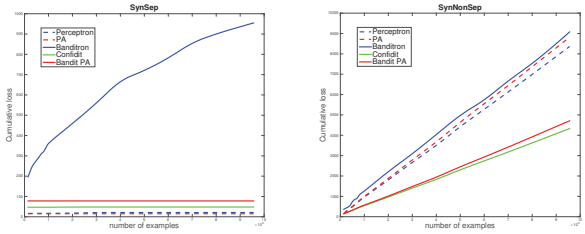


Fig. 1. Cumulative Errors on the synthetic data sets: SynSep and SynNonSep.

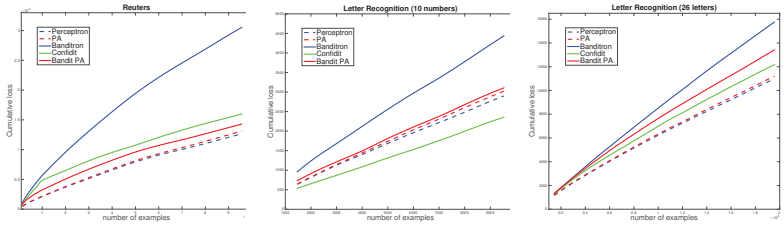


Fig. 2. Cumulative Errors on the real data sets: RCV1-v2 (53 classes), Letter Recognition (10 numbers) and Letter Recognition (26 Letters).

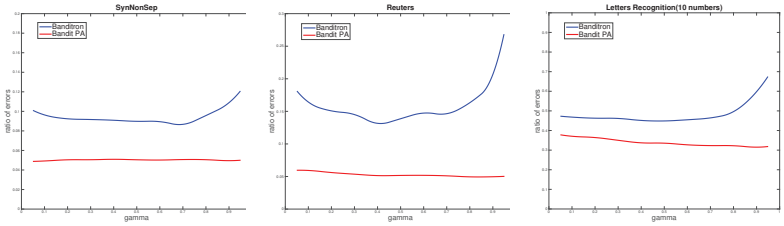


Fig. 3. Average error on Banditron and BPA for parameter's value γ.

6 Conclusion and Open Questions

In this paper, we proposed a novel algorithm for online multiclass with bandit feedback. By the advantage of PA max-margin principle, BPA appears effective to address the bandit online learning setting. Its main advantage is its linear complexity in space that allows to deal with high dimensional data sets and a large number of classes, on the contrary to second-order methods. The practicability of this algorithm is verified theoretically by showing a competitive loss bound.

Moreover, experimental evaluation shows that BPA performs better than other algorithms on some real sets, even better than the algorithms with full feedback on the data sets non-separable.

Ongoing research, we will take BPA to deal with data sets non-linear by combining the Kernel method. Otherwise, Algorithm PA could be adapted to the task of multilabel classification. So our work could be extended by the problem of "Multilabels in bandit setting" which is proposed by [14].

ACKNOWLEDGEMENT

This work is partially supported by the ANR-funded project GRETA – Greediness: theory, algorithms (ANR-12-BS02-004-01) and China Scholarship Council.

References

1. Kakade, Sham M., Shalev-shwartz, Shai, et Tewari, Ambuj: Efficient bandit algorithms for online multiclass prediction. In: *Proceedings of the 25th international conference on Machine learning*, ACM, p. 440-447.(2008)
2. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer: Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*. (2006)
3. K. Crammer, C. Gentile: Multiclass Classification with Bandit Feedback using Adaptive Regularization. *ICML* . (2011)
4. Ryan McDonald, Koby Crammer, Fernando Pereira: Online Large-Margin Training of Dependency Parsers. In: *ACL*. (2005)
5. David Chiang, Yuval Marton, Philip Resnik: Online Large-Margin Training of Syntactic and Structural Translation Features. In: *EMNLP*. (2008)
6. S.Boyd, L.Vandenberghe: Convex Optimization.*Cambridge University Press*,2004.
7. D.D. Lewis, Y. Yang, T.G. Rose, and F. Li: RCV1: A new benchmark collection for text categorization research. In: *JMLR*, vol.5, p.361-397.(2004)
8. Ron Bekkerman and Martin Scholz: Data weaving: Scaling up the state-of-the-art in data clustering.*In Proceedings of CIKM*, p.1083-1092.(2008)
9. J. Alcala-Fdez, L. Sanchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernandez, F.Herrera: KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. In:*Soft Computing*, vol.13, issue.3, p.307-318.(2009)
10. J. Alcala-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. Garcia, L. Sanchez, F. Herrera: KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. In: *Journal of Multiple-Valued Logic and Soft Computing*, vol.17, issue.2-3, p.255-287.(2011)
11. F. Rosenblatt:The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Rev*, vol.65, p.386-407.(1958)
12. P. Auer: Using confidence bounds for exploitation- exploration trade-offs.*JMLR*, vol.3.(2003)
13. V.N. Vapnik: Statistical Learning Theory. *Wiley*.(1998)
14. C. Gentile, F. Orabona: On multilabel classification and ranking with bandit feedback. *The Journal of Machine Learning Research*, vol.15, no.1, p.2451-2487.(2014)

ECMLPKDD 2015 Doctoral Consortium was organized for the second time as part of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD), organised in Porto during September 7-11, 2015. The objective of the doctoral consortium is to provide an environment for students to exchange their ideas and experiences with peers in an interactive atmosphere and to get constructive feedback from senior researchers in machine learning, data mining, and related areas. These proceedings collect together and document all the contributions of the ECMLPKDD 2015 Doctoral Consortium.

ISBN 978-952-60-6443-7 (pdf)
ISSN-L 1799-4896
ISSN 1799-4896 (printed)
ISSN 1799-490X (pdf)

Aalto University
School of Science

www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS

Aalto-ST 12/2015