OCTOBER 13TH TO 15TH, 2015

PETRÓPOLIS - LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA - LNCC

PROCEEDINGS OF THE 3RD SYMPOSIUM ON KNOWLEDGE DISCOVERY, MINING AND LEARNING

ALEXANDRE PLASTINO, SANDRA DE AMO, LEANDRO BALBY MARINHO (EDS.)

3rd SYMPOSIUM ON KNOWLEDGE DISCOVERY, MINING AND LEARNING

October 13th to 15th, 2015
Petrópolis – RJ – Brazil

PROCEEDINGS

Organization
Fluminense Federal University – UFF
National Laboratory for Scientific Computing – LNCC
Federal Center of Technological Education of Rio de Janeiro – CEFET/RJ

Local Organization Chair
Alexandre Plastino, UFF

Program Committee Chairs
Sandra de Amo, UFU
Leandro Balby Marinho, UFCG

Steering Committee Chairs
André Ponce de Leon F. de Carvalho, ICMC-USP
Wagner Meira Jr., UFMG

Support
Brazilian Computer Society – SBC
International Association for Statistical Computing – IASC

ISSN: 2318-1060

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da Universidade Federal Fluminense

S989 Symposium on Knowledge Discovery, Mining and Learning
(3.: 2015: Petrópolis, RJ).
Proceedings / 3rd Symposium on Knowledge Discovery, Mining and Learning ; Alexandre Plastino, Sandra de Amo, Leandro Balby Marinho, editors ; Universidade Federal Fluminense, Laboratório Nacional de Computação Científica, Centro Federal de Educação Tecnológica do Rio de Janeiro, organizadores. – Petrópolis : [s.n.], 2015.

97 p.
Evento realizado de 13 a 15 de outubro de 2015.
ISSN 2318-1060

1. Mineração de dados. 2. Aprendizagem de máquina. 3. Ciência da Computação. I. Plastino, Alexandre. II. Amo, Sandra de. III. Marinho, Leandro Balby. IV. Universidade Federal Fluminense. V. Laboratório Nacional de Computação Científica. VI. Centro Federal de Educação Tecnológica do Rio de Janeiro. VII. Título.

CDD 005.741 (21. ed)

Editorial

The Symposium on Knowledge Discovery, Mining and Learning (KDMiLe) aims at integrating researchers, practitioners, developers, students and users to present theirs research results, to discuss ideas, and to exchange techniques, tools, and practical experiences – related to Data Mining and Machine Learning areas.

KDMiLe is organized alternatively in conjunction with the Brazilian Conference on Intelligent Systems (BRACIS) and the Brazilian Symposium on Databases (SBBD). This year, in its third edition, KDMiLe will be held in Petrópolis, a city in the state of Rio de Janeiro, from the 13th to 15th of October in conjuntion with SBBD.

The KDMiLe program includes two short courses, which will be presented by two experts in each topic: "Introduction to Machine Learning", by André Ponce de Leon F. de Carvalho (ICMC-USP), and "Four Paradigms in Data Mining", by Wagner Meira Jr. (UFMG).

KDMiLe will also offer a tutorial on "Mining Data in Cognitive Era", by Ana Paula Appel and Heloisa Candello, both from IBM Research Brazil, and a panel, coordenated by Luciana Alvim Santos Romani (Embrapa), where topics such as data science, big data and cognitive computing will be discussed.

The program committee evaluated 44 submissions and selected 13 papers, which corresponds to an acceptance rate of 30%. These papers were organized in four technical sessions, where authors will present and discuss their work.

We thank SBBD Organization Committee for hosting KDMiLe at LNCC (Laboratório Nacional de Computação Científica) and also our sponsors for their valueable support. We are also grateful to the Program Committee members and external reviewers who carefully evautated the submitted papers and, mainly, to the authors who submitted their research work to KDMiLe and contributed to a high quality edition of this growing event in Data Mining and Machine Learning areas.

Petrópolis, 13th October, 2015

Alexandre Plastino, UFF
KDMiLe 2015 Local Organization Chair

Sandra de Amo, UFU
KDMiLe 2015 Program Committee Chair

Leandro Balby Marinho, UFCG
KDMiLe 2015 Program Committee Chair

3rd Symposium on Knowledge Discovery, Mining and Learning

October 13-15, 2015
Petrópolis – RJ – Brazil

Organization

Fluminense Federal University – UFF
National Laboratory of Scientific Computing – LNCC
Federal Center of Technological Education of Rio de Janeiro – CEFET/RJ

Support

Brazilian Computer Society – SBC
International Association for Statistical Computing – IASC
International Statistical Institute – ISI

KDMiLe Steering Committee

Alexandre Plastino, UFF
André Ponce de Leon F. de Carvalho, ICMC-USP
Leandro Balby Marinho, UFCG
Sandra de Amo, UFU
Wagner Meira Jr., UFMG

KDMiLe 2015 Committee

Local Organization Chair
Alexandre Plastino, UFF

Program Committee Chairs
Sandra de Amo, UFU
Leandro Balby Marinho, UFCG

Steering Committee Chairs
André Ponce de Leon F. de Carvalho, ICMC-USP
Wagner Meira Jr., UFMG

KDMiLe Program Committee

Sandra de Amo (UFF, Brazil, PC Chair)
Leandro Balby Marinho (UFCG, Brazil, PC Chair)

Adriana Bechara Prado (EMC Brazil R&D Center, Brazil)
Adriano Veloso (UFMG, Brazil)
Alexandre Plastino (UFF, Brazil)
Aline Paes (UFF, Brazil)
Ana L. C. Bazzan (UFRGS, Brazil)
Ana Paula Appel (IBM Research, Brazil)
Andre Carvalho (USP, Brazil)
Angelo Ciarlini (EMC Brazil R&D Center, Brazil)
Aurora Pozo (UFPR, Brazil)
Carlos Eduardo Pires (UFCG, Brazil)
Carlos Soares (University of Porto, Portugal)
Cícero Nogueira dos Santos (IBM Research, Brazil)
Edson Matsubara (UFMS, Brazil)
Elaine Faria (UFU, Brazil)
Elaine P. M. de Sousa (USP, Brazil)
Fabio Cozman (USP, Brazil)
Fernando Otero (University of Kent, UK)
Flavia Bernardini (UFF, Brazil)
Francisco de A.T. de Carvalho (UFPE, Brazil)
Gisele Pappa (UFMG, Brasil)
Herman Gomes (UFCG, Brasil)
Humberto Luiz Razente (UFU, Brazil)
Jose Alfredo Ferreira Costa (UFRN, Brazil)
Julio Cesar Nievola (PUC-PR, Brazil)
Kate Revoredo (UNIRIO, Brazil)
Leonardo Rocha (UFSJ, Brazil)
Luis Zárate (PUC-MG, Brazil)
Luiz Merschmann (UFOP, Brazil)
Marcelino Pereira (UERN, Brazil)
Marcelo Albertini (UFU, Brazil)
Marcelo Ladeira (UNB, Brazil)
Marcio Basgalupp (ICT-UNIFESP, Brazil)
Marcílio de Souto (LIFO/University of Orleans, USA)
Maria Camila Nardini Barioni (UFU, Brazil)
Maria Gatti (IBM Research, Brazil)
Nuno C. Marques (FCT/UNL, Portugal)
Ricardo Prudêncio (UFPE, Brazil)
Ronaldo Prati (UFABC, Brazil)
Rui Camacho (LIACC/FEUP University of Porto, Portugal)
Vasco Furtado (UNIFOR, Brazil)
Wagner Meira (UFMG, Brazil)

External Reviewers

Adriano Rivolli
André L.D. Rossi
Anisio Lacerda
Carlos Affonso
Christian Cesar Bones
Cláudio Rebelo de Sá
Eanes Pereira
Eduardo Corrêa
Elaine Faria
Fábio Paiva
Leandro Pasa
Marcos Cintra
Pedro Saleiro
Rômulo Pinho
Tiago Cunha

Table of Contents

A Social Approach for the Cold-Start Issue on Recommender Systems Based on the Extraction and Analysis of Web Resources

Antonio Ferreira Podgorski Bezerra, Julio Cesar Duarte

IME - Instituto Militar de Engenharia, Brazil
antonio@podgorski.com.br duarte@ime.eb.br

Abstract. In general, a recommender system helps people perform choices among several alternatives presented, trying to maximize the possibilities to find interesting and valuable information that may help with their decisions. In recent years, it is clearly observed that information in the web is exponentially growing, mainly from social interactions. However, its heterogeneous sources and how they are structured make their extraction and analysis a complex process, thus, several research areas are working actively in this theme. In this article, we present an approach to improve the collaborative filtering technique by expanding the user-item matrix, having as motivation this social information overload and classic recommender systems limited context acting. Our approach alleviates the cold-start problem, a common issue with normal recommender systems, by using this information to create models of social users, without interfering with privacy concerns of real users that collaborate to build the models. This expansion process can also be used to improve classic recommender systems with few ratings in the database. In a real case scenario, the observed results in the experiment showed an improvement in the quality of the predictions and recommendations of items in cold-start situations of about 161% and 309%, respectively, when compared to classic CF methods.

Categories and Subject Descriptors: H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*data types and structures*; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*

Keywords: cold-start, collaborative filtering,cross-domain,information retrieval, recommender systems, web resources

1. INTRODUCTION

The first computational recommender systems emerged in 90s [Goldberg et al. 1992], in order to automate the recommendation process and to help people choose between several alternatives. In other words, recommender systems try to maximize the possibilities of finding interesting and valuable information to support decisions. The goal of a recommender system is to generate suggestions about new items or to predict the utility of a specific item for a particular user [Sigroha and Rana 2012].

According to [Ricci et al. 2011], collaborative filtering (CF) is a widely used technique for recommender systems. It assumes that people who agreed in the past, will also agree in the future, where the similarity in taste of two users is evaluated based on the similarity in the rating history of the users.

In a basic CF implementation, the input data is a matrix (M) **user x item**, where each M_{ij} corresponds to a user's rating in a particular item. With this rating matrix, the similarity between two users can be evaluated generating neighborhoods of similar users. Finally, the ratings for unseen items are evaluated to make predictions for a target user.

The cold-start problem [Huang et al. 2004] refers to the situation in which a new user or item has

just entered the system, and a CF cannot generate useful recommendations for a new user due to a lack of sufficient previous ratings or the presence of new items with few users ratings. The sparsity in the matrix is one of the factors that contribute directly for this cold-start issue. In this work, we focused in items that suffer from cold-start, this phenomenon generally is observed in unpopular or brand-new items.

A cross-domain recommendation system [Cremonesi et al. 2011] tries to take advantage of the fact that users' preferences are spanned across different application areas. By recommending items on a domain A, one can use ratings or reviews made on a different auxiliary domain B. This approach is very suitable to determine recommendations for cold-start items, since we can find the items themselves or similar ones in this auxiliary domain. In this context, we can recommend, for instance, movies based on music reviews. Here, we use a relaxed definition for a cross-domain recommendation system, where domains A and B can be the same, this is similar to the experimental setup proposed by [Cremonesi et al. 2011].

Nowadays, we can clearly observe information overload in the web, mainly from social interactions. Its heterogeneous, decentralized and organic form make their extraction and analysis a complex process, thus, several research areas are working actively in this theme. The observed informative potential and social context involved are extremely relevant to improve classic recommender systems [Rodriguez et al. 2014]. In this context, many researchers have proposed personalized recommender systems based on social environments [Zhou et al. 2012] using techniques that have been largely applied like tag-based, context-based, social influence-based and trust-based recommendation system [Keikha et al. 2013].

Our approach, on the other hand, is focused in extracting and analyzing social environments in order to create models of social users. The social user models are composed by the retrieved information that could be extract based on user ratings in the auxiliary domains, that will expand the original matrix of interactions in the main domain, thus, improving the accuracy of classic CF recommender systems and allowing the integration and recommendation of new items for the main domain. The aim of this study is to alleviate the cold-start problem using the social overload information to create models of social users, recent literature defines this as a cross-domain approach.

The remainder of the paper is organized as follows. Section 2 presents the state-of-art and related works that uses different cross-domain techniques to improve recommendations. In section 3, we present our social approach methodology for expanding recommender systems with models of social users. In section 4, we create an instance to validate our methodology and constructed the data set for experiments. In section 5, we present the used evaluation metrics. In section 6, we explain the experiments' setup and results obtained in our experiments. Finally, in section 7, the conclusion of this study is presented and future work following this line and expanding this study are proposed.

2. STATE-OF-ART AND RELATED WORKS

Previous studies used different cross-domain techniques, demonstrating improvements in recommendation results if compared with classic CF methods.

In [Berkovsky et al. 2007], one of the first studies that presents this concept, four approaches were proposed: centralized prediction, distributed peer identification, distributed neighborhood formation, and distributed prediction. In this approach, overlapped users between the domains are necessary. [Heitmann and Hayes 2010] presents a different cross-domain approach using web semantic and structured web data for collections of information, by acquiring structured information from RDF, the user-item relations are transformed into an auxiliary user-item matrix that links cross-domain data. [Cremonesi et al. 2011] proposed a formal definition for the cross-domain technique addressing the evaluation of state-of-the-art algorithms and algorithms to perform on cross-domain scenarios that outperforms traditional CF algorithms. [Enrich et al. 2013] uses the tag information that overlaps

between domains to allow a cross-domain technique independent of domains that share some users. However, an additional information, the tag, is necessary to allow this approach beyond user-item matrices.

Hence, in our approach, we propose a methodology independent of overlapped users or additional information, like tags or latent shared factors between users to alleviate the cold-start issue. Observing that many real collaborative recommender systems don't contain additional information. So, we only use the traditional users' rating information in the user-item matrix, supported by social user models.

3. A SOCIAL APPROACH METHODOLOGY FOR EXPANDING RECOMMENDER SYSTEMS DATA SETS

In order to allow items that suffer from the cold-start issue to have sufficient interactions, and, thus, be recommended, we propose a new methodology based on the extraction and analysis of web resource content that expands the original interaction matrix, alleviating this cold-start effect and improving the performance of our recommender system.

The web resources used in our methodology can be divided in two forms: structured and unstructured. Structured resources can be defined as content where it is possible to establish or recognize some pattern, like a markup language (e.g. HTML, XML). In this case, applying techniques such as web scraping are possible in order to search and retrieve the desired content. Unstructured resources, nevertheless, have little or no organization that makes it possible to establish a pattern to follow, the content can be searched and retrieved through web scraping techniques or the use of a provided service like an API, commonly available in social networks. Techniques like NLP or sentiment analysis must be used then, to infer an analysis of the value-content retrieved. In both cases, the process is started using an item that is unpopular or brand-new to serve like a seed for the process chain.

As shown in figure 1, the Social Core (SC) starts with a cold-start item as input, known as seed item (SI), with which we desire to obtain more knowledge based on a target web resource. The expected output is made of models of social users related with this SI. The SC has two main modules, the Search Module (SM) and the Preprocessing and Analysis Module (PAM). SM is responsible to interface with web resources, this interaction consists in a two-step process. The first step consists of using the SI to retrieve all users on the web resource, or a subset of them, that interacted with the item. Before starting the second step, in order to minimize the number of requests, it is necessary to build a pool of requests, which will be treated as virtual threads, making it possible to perform asynchronous requests for each user, avoiding duplicity. The responsible module for this activity is the Structure Data Module (SDM). By doing this, the second step consists of SM sending users in the pool to the web resource and getting all users' interactions, or a subset of them.

The PAM is an auxiliary module. Depending on the nature of the information retrieved, in this module more analysis can be necessary to build more accurately social user-model. So, the social user-models are returned from the SC, allowing us to build the auxiliary domain. Although our methodology contemplates the use of both structured and unstructured data, our experiments in this article use only data extracted from structured resource. It is worth to note here, that our methodology is algorithm-independent and can work in any recommender scenario.

4. SOCIAL USER MODELS RETRIEVAL AND DATA SET CONSTRUCTION

In order to evaluate the performance of our methodology, a structured web resource was chosen to be used in our experimental process. Since recommender systems are widely used in movie rating systems, we chose IMDb website[1] for our experiments. Inside the IMDb website, there is a section

[1] http://www.imdb.com

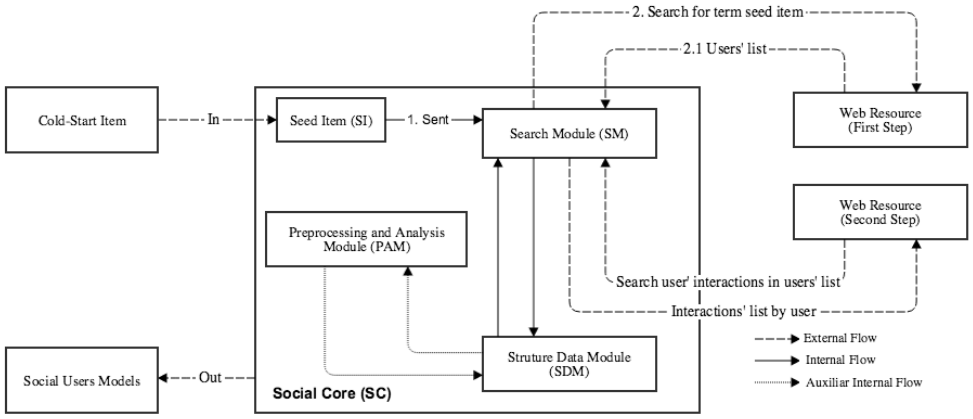


Fig. 1. Social Core Workflow in Two-Step Process

Table I. Month Groups' Distribution.

Month/Year	**Seed Items(Coming Soon)**	**Retrieved User Models (All)**
October/14	50	3,258
September/14	45	2,179
August/14	33	2,090
July/14	28	1,317
June/14	28	1,993
May/14	26	2,324
Total	**210**	**13,161**

called coming soon, which presents the movies that will be released during a month. This scenario allows us to validate our methodology in a simulated cold start situation with real brand-new items. We consider these movies inside the coming soon section as seed items for the SC. From May to October (2014), these movies served as seeds to the SC, which, then, retrieves different IMDB users to build the social user models, as illustrated in table I.

In order to evaluate the results, it is necessary to simulate the cross-recommendation scenario, by splitting the social users' models retrieved by the SC into our main and auxiliary domains. Randomly, we divide the user models in equal proportions for each domain. In other words, we create, for each month from the coming soon section, two independent domains without user intersection to perform our experiments, obtaining the main and auxiliary domain, respectively, domains A and B, as presented in figure 2.

5. EVALUATION

We use two metrics to evaluate our methodology, RMSE and F1-Score. The RMSE is an excellent general-purpose error, while F1-Score is good for information retrieval. The experiments were performed using the user-user algorithm, also known as k-NN. [Herlocker et al. 2002] presents k values between 20 and 50 as reasonable to define the user's neighborhood using the Pearson Correlation. Using a k value smaller than 20 should result in few neighbors and over-fitting estimations. On the other hand, a higher k value can adding noise to it. With that in mind, in this article, the k value is defined as 30.

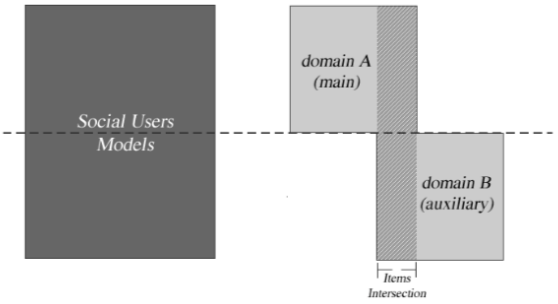


Fig. 2. Dataset Construction

5.1 RMSE

The first metric used to evaluate the hypotheses is the root mean square error (RMSE) [Jannach et al. 2010]. We observed this as the most popular metric in recommender systems, both in academic researches and commercial scenarios. As the comparison of different hypotheses using an absolute basis is difficult to achieve, we propose then a metric to compare how far our hypothesis (H) is, in respect to our best and worst cases to be presented in section 6. We call this metric the RMSE-GAP. It uses a relative basis rather than an absolute one, to make easier to evaluate our results, $RMSEGAP(H) = \frac{RMSE(Best\ Case) - RMSE(H)}{RMSE(Best\ Case) - RMSE(Worst\ Case)}$. This measure represents in terms of percentage how our methodological case is closer to the best case or worst case. If the result tends to be 0%, it means that the methodological case is near the best case, if the result is near 100% mean the methodological case is near the worst case.

5.2 F1-Score

The second metric chosen to evaluate the hypotheses is the F1-Score. Despite the fact that the RMSE metric is more popular in literature, [Cremonesi et al. 2010] showed that improvements in RMSE do not necessarily perform as expected in terms of top-n recommendation tasks. The main idea to use this measure is to consider that the purpose of a real world recommender system is to produce a top-n list of recommendations and evaluate, depending on how well they can classify items as being recommendable, looking at our recommendation task as a classification problem. Then, we can use well-known measures for classifier's evaluation such as precision and recall [Jannach et al. 2010].

The F1-Score measure is frequently used in information retrieval. It is evaluated as the harmonic-mean between precision and recall, giving the same weight for both. The F1-Score final results oscillate between zero and one. The closer the result tends to 1, the better the items are predicted. [Cremonesi et al. 2011] proposed the approach used in our experiments. The F1-Score formula is adapted to the top-n items, where n is a number of items that a recommender system will suggest to a user.

Based in our methodology each user model inside the social core was retrieved through at least one seed item, a cold-start. However, in order to fairly evaluate F1-Scores on top n items, the same user must have enough cold-start candidates' ratings in the testing data set.

Since we deal with cold-start items, this number of candidates is usually small per month. So, we grouped four months in sequence, as shown in table II, generating new main and auxiliary domains, allowing us to produce eligible users for the metric. In other words, to be considered eligible, a user in the main domain must have reviewed a minimum number of cold-start items in the testing data set. For each month group (MG), we create the domains A and B. In domain A, there are cold start items that are not possible to be recommended to any user in this domain.

Table II. Month Groups' Distribution and total eligible users with minimum cold start items

	May	June	July	Aug.	Set.	Oct.	Eligible Users	
							>=10 items	>=20 items
MG01	✓	✓	✓	✓			78 users	21 users
MG02		✓	✓	✓	✓		84 users	18 users
MG03			✓	✓	✓	✓	79 users	22 users

6. EXPERIMENTAL SETUP AND RESULTS

In order to validate the effectiveness of our methodology related to RMSE metric, we evaluate our approaches against two baselines of comparison. In these hypotheses, we simulate the behavior of a recommender system that uses only CF basic algorithms in the main domain A without the new ratings from the auxiliary domain B. We call them, "worst" and "best" case. The use of quotation marks is needed because they are directly influence by the choice of CF technique, and depending on the algorithms used, small variations in the results can be presented. Our intention with these two hypotheses is to simulate extreme situations in a specific scenario of a classic recommender system. We used MyMediaLite [Gantner et al. 2011] due be a lightweight, multipurpose library of recommender system algorithms in our experiments. We call our approach the methodological case, where we apply a recommender system by expanding the main domain A matrix with the auxiliary domain B, appending our social users' models in the original matrix, and comparing our results with the previous ones obtained in the worst and best case, using the same algorithms.

These three hypotheses are tested in each month. Our aim is to evaluate how accurately and predictive each hypothesis will perform in practice, so we propose to divide these ratings in training and testing data sets. To simulate the hypothesis, we divide the ratings in different ways. To maintain the temporal rating behavior, we only consider, for each month, ratings until the last day of each month.

In worst case, we intend to simulate the cold-start situation to all respective months released items. Meanwhile, in the training data set, we used ratings provided for the movies that were released in the previous months, and as testing data set we used only the ratings provided from movies released during the time frame. So, the user average rating was used as baseline. Meanwhile, in best case we intend to split the ratings provided for the cold-start movies between our training and testing data set. So, we use the k-fold cross-validation method for this purpose, where we divided these ratings randomly and partitioned them into k equal size sub-samples. In our case k equals 10, using each sub-sample as testing exactly one time. After all test folds are computed, an average is evaluated. Worst case is the situation where we have no information to help us with the cold-start items, whereas in best case, we have these information and we can treat the items as normal ones. These scenarios represent standard CF techniques.

On the other hand, in our methodological case we intend to apply our social user models from auxiliary domain B to expand the original matrix in domain A. So the training and testing data sets are similar to the worst case scenario, however, we add in the training data set all ratings provided from the social user models generated from the respective coming soon month. Differently from the best case, this is a real scenario for a recommendation system that can deal with cold-start items. The results in tables III and IV shown an improvement in predictions when applied our social user models. In table III the negative RMSE-GAP shown our methodology outperform the best case in some months. While in table IV all month group outperform the best case.

In order to validate the effectiveness of our methodology related to F1-Score metric, we apply the social users' models from domain B as an extension of the ratings matrix from domain A, as in methodological case. Thus, these experiments divide the eligible users in two groups, the first one,

Table III. RMSE Hypothesis and RMSE-GAP per Coming Soon.

	RMSE			
	Best Case	**Worst Case**	**Methodological Case**	**RMSE-GAP**
May/14	1.1045	1.1698	1.0815	-35.22
June/14	1.0137	1.1250	1.0339	18.15
July/14	1.1292	1.1812	1.0881	-79.04
August/14	1.0083	1.1276	1.0103	1.68
September/14	1.2162	1.2513	1.2180	5.13
October/14	1.2464	1.2745	1.2366	-34.88

Table IV. RMSE Hypothesis and RMSE-GAP per Month Groups.

	RMSE			
	Best Case	**Worst Case**	**Methodological Case**	**RMSE-GAP**
MG01	1.019	1.1052	0.9571	-71.81
MG02	1.0295	1.1219	0.9729	-61.26
MG03	1.0763	1.1625	1.0328	-50.46

Table V. Baseline and F1-Score Results.

	>=10 Items			>=20 Items		
	top 1	**top 3**	**top 5**	**top 1**	**top 3**	**top 5**
Baseline	0.070	0.197	0.327	0.045	0.128	0.187
MG01	0.308	0.551	0.664	0.238	0.571	0.629
MG02	0.393	0.571	0.636	0.444	0.556	0.600
MG03	0.392	0.595	0.656	0.273	0.485	0.591

users that have at least ten possible cold-start items unseen by the user, the second one, at least twenty. For the top-n list of recommendations, we chose n to be 1, 3 or 5 items. To compare the results obtained using the F1-Score (n), we define as a baseline random algorithm that makes random top-n recommendation, similar as the worst case. The random baseline was adopted, since most of the algorithms implementations using classic CF uses the average of all evaluations already made by user to generate the prediction to a cold-start item. Thus, as all eligible items are cold-start items, they would have the same predicted evaluation, making it impossible to build a valid ranking. We then perform the F1-Score evaluation on these selected users obtaining the results described in table V. The results show, despite the fact that the average F1-score with more cold-start items are smaller in absolute values, a high percentage improvement if compared to the random baseline, which allows us to make a better recommendations' list.

7. CONCLUSION AND FUTURE WORK

The main goal of a recommender system is to provide users new ways to interact with them, in a way that his overall satisfaction is always improved. These recommendations are usually based in previous interactions of the user with the system, which makes difficult the integration of new items to the process. In this work, we presented a cross-domain approach using overloaded information from the web to build models of social users in order to expand the information used from the recommendation and alleviate the cold-start situation in a main domain.

The novelty of this work is the proposition of a cross-domain technique for cold-start items recommendation that doesn't need additional information. The experimental results of our methodology, performed in a structured web resource, proved that is possible to improve the quality of the predic-

tions and recommendations of items in cold-start situations of about 161% and 309%, respectively, when compared to the classic CF methods in proposed baselines. Our main contributions are the independent methodology of overlapped users or additional information to apply in a cross-domain technique, and an instance of this, using structured web resources in the movies' domain. We consider cross-domain and overloaded information from the web promising, and since there are many open questions to new researches in both areas, we motivate and propose future work in real world cross-domain scenario.

As observed in [Sahebi and Brusilovsky 2013], models' size can improve directly the results, so we believe more studies are necessary to allow computationally cross-domain approaches to be smoothly applied in real world tasks. Moreover, [Bao and Zhang 2014] proposed a simultaneous ratings and reviews exploiting for recommendations, motivating a hybrid approach to construct the users' model that allows the use of no-rating reviews. Finally, we plan to apply our methodology in an unstructured domain, as a social network like Twitter, to validate it. Of course, we expect worst results from unstructured domains, but they can provide us with reviews from completely brand-new items.

We believe the experiments and improvements proposed will allow the definition of better approaches and techniques in the cross-domain research's area dealing to cold start items.

REFERENCES

Bao, Y. and Zhang, H. F. J. Topicmf: Simultaneously exploiting ratings and reviews for recommendation, 2014.

Berkovsky, S., Kuflik, T., and Ricci, F. Cross-domain mediation in collaborative filtering. In *User Modeling 2007*. Springer, pp. 355–359, 2007.

Cremonesi, P., Koren, Y., and Turrin, R. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, pp. 39–46, 2010.

Cremonesi, P., Tripodi, A., and Turrin, R. Cross-domain recommender systems. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, pp. 496–503, 2011.

Enrich, M., Braunhofer, M., and Ricci, F. Cold-start management with cross-domain collaborative filtering and tags. In *E-Commerce and Web Technologies*. Springer, pp. 101–112, 2013.

Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, 2011.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35 (12): 61–70, Dec., 1992.

Heitmann, B. and Hayes, C. Using linked data to build open, collaborative recommender systems. In *AAAI spring symposium: linked data meets artificial intelligence*. pp. 76–81, 2010.

Herlocker, J., Konstan, J. A., and Riedl, J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval* 5 (4): 287–310, 2002.

Huang, Z., Chen, H., and Zeng, D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)* 22 (1): 116–142, 2004.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. *Recommender Systems: an Introduction*. Cambridge University Press, 2010.

Keikha, F., Fathian, M., and Gholamian, M. R. Comparison and evaluation of recommendation systems on social networks. *Journal of Basic and Applied Scientific Research* 3 (10): 52–58, 2013.

Ricci, F., Rokach, L., and Shapira, B. *Introduction to Recommender Systems Handbook*. Springer, 2011.

Rodriguez, M. G., Gummadi, K., and Schoelkopf, B. Quantifying information overload in social media and its impact on social contagions. *arXiv preprint arXiv:1403.6838*, 2014.

Sahebi, S. and Brusilovsky, P. Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation. In *User Modeling, Adaptation, and Personalization*, S. Carberry, S. Weibelzahl, A. Micarelli, and G. Semeraro (Eds.). Lecture Notes in Computer Science, vol. 7899. Springer Berlin Heidelberg, pp. 289–295, 2013.

Sigroha, D. and Rana, C. Survey Paper on Analysis of Various Recommendation Algorithms. *Journal of Computer Science* 3 (2): 3406–3408, 2012.

Zhou, X., Xu, Y., Li, Y., Josang, A., and Cox, C. The state-of-the-art in personalized recommender systems for social networking. *Artificial Intelligence Review* 37 (2): 119–132, 2012.

Predicting Student Dropout: A Case Study in Brazilian Higher Education

Allan Sales, Leandro B. Marinho, Adalberto Cajueiro

Universidade Federal de Campina Grande, Brazil
allan.melo@ccc.ufcg.edu.br, {lbmarinho,adalberto}@dsc.ufcg.edu.br

Abstract.

Students' dropout is a major concern of Brazilian higher education institutions because it can result in waste of resources and hence decrease the graduation rates. Most of the dropouts occur in the initial semesters of a course, especially in the first one, where students are still uncertain about the career they want to follow. Thus, the early detection of students with high probability of dropping out, as well as understanding the underlying causes, are crucial for defining more effective actions towards preventing this problem. In this paper, we cast the dropout detection problem as a classification problem. We use a large sample of academic records of students across 130 courses from a public University in Brazil in order to select informative features for the employed classifiers. Considering only first semester students as targets, we conduct a thorough evaluation of several state-of-the-art classification models and show that good results can be achieved considering only a small, but informative, number of features.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.2.6 [**Artificial Intelligence**]: Learning

Keywords: Dropout, Education, Educational Data Mining, Higher education institutions, Learning analytics, Machine Learning

1. INTRODUCTION

With the creation of several public policies towards expanding the access to Brazilian higher education, the number of enrollments has notably increased in recent years. In 2013, for example, more than 7 million enrollments were registered and this number is continuously growing up [de Estudos e Pesquisas Educacionais Anísio Teixeira 2013]. However, it is estimated that only 62.4% of these enrollments succeeds in getting a degree [de Estudos e Pesquisas Educacionais Anísio Teixeira 2010], which suggests a high rate of dropout students.

The student dropout problem occurs widely in several levels of education around the world. The most common reasons associated with this problem are poor grades, bad teaching or badly structured subjects, getting a job before or during the studies, lack of employment perspective, family issues and lack of aptitude for the course [GAIOSO 2005; Barroso and Falcão 2004; ADACHI 2009; Andriola et al. 2006]. Many studies have pointed out that the occurrence of dropouts is larger in the beginning of the courses, due to some of the aforementioned reasons [Dekker et al. 2009; Pal 2012]. Considering the dataset we used in our experiments (see Section 4 for more details) comprising 130 higher education courses in the Federal University of Campina Grande (UFCG) - Brazil, for example, we observed, through the cumulative distribution of dropouts per semester depicted in Figure 1, that more than 60% of the dropouts occurs in the first three semesters. This observation has motivated us to focus our investigation on first semester students.

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES), funded by CNPq and FACEPE, grants 573964/2008-4 and APQ-1037-1.03/08.

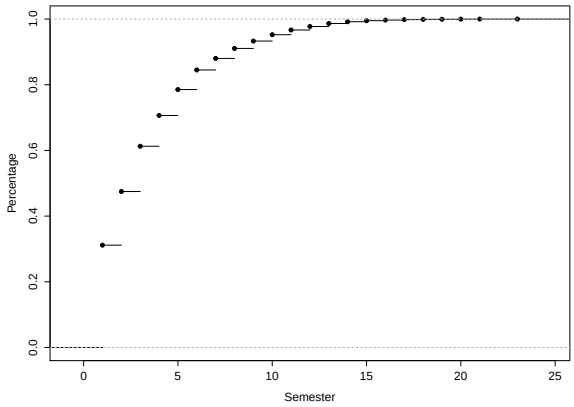


Fig. 1: Dropouts per semesters enrolled

In this paper we formulate the student dropout detection problem as a supervised learning problem using features extracted from academic records of students. To this end, and considering only the first semester students as targets, we employ classification models that categorizes these students into two different classes: 'dropout' or 'continue'. I.e., we want to identify, among the first semester students, the ones that will continue in the University after the first semester is over. We performed feature selection and evaluated many different classifiers with a variety of features, and discovered that a small subset of these features is sufficient for achieving good results.

Our approach is inserted in the field known as Educational Data Mining (EDM)[Romero and Ventura 2013] which has been a powerful tool to help educational institutions to devise better corrective and preventive actions such as improving the allocation of resources and staff or advising students identified as potential dropouts. Some works have appeared recently proposing to apply machine learning to detect students' dropout (cf. Section 2). We extend these works with the main following contributions:

—We use a large and comprehensive dataset of student's academic records from 130 different courses of a public Brazilian University;
—We focus on first semester students only;
—We conduct a feature selection analysis in order to discover which features have the highest impact in the classifiers' performance;
—We compare several state-of-the-art classification algorithms for the problem addressed in this paper.

2. RELATED WORK

There are several works that address the students' dropout problem, each one approaching a different perspective of the problem. Below we briefly describe the research works most related to ours.

Márquez-Vera et al. [Márquez-Vera et al. 2013] investigate students failure at high school in a city of Mexico. They use several popular classification algorithms and propose a genetic algorithm approach that considers cost-sensitive learning and class imbalance techniques. We consider the dropout problem in Brazilian public higher education which is a related but different problem in comparison to dropout in high school.

Mustafa et al. [Mustafa et al. 2012] exploits whether registration data of students (e.g., financial support, age, gender and disabilities) in the courses of Computer Science and Engineering at the University of Chittagong, are good features for predicting dropout. The authors use decision trees

classifiers and conclude that the most important features to predict dropout are financial support, age and gender. It is further stated that the accuracy of the trees were only 38.10%. It means that financial support, age and gender have some impact on the prediction performance but using them alone is not enough.

Pal [Pal 2012] proposes to predict dropout before the students start their first academic year. To accomplish that, the author tests four classification algorithms using socio-economical data and pre-university data (e.g., student grades in high school) features. The models vary the accuracy rate from 67.7% to 85.7%. He concludes that the performance of students in high school is the most discriminative feature in the classification model. Our model differs from this approach in the sense that we consider students already coursing their first academic semesters. Moreover, we do not have access to socio-economic or pre-university information about the students.

Dekker et al. [Dekker et al. 2009] investigate the dropout detection problem in Electrical Engineering courses after or before the first academic semester. To accomplish this goal, they used student's data during their first academic semester and pre-university data as input to eight classification algorithms. They used cost-sensitive learning for handling class imbalance and evaluate the algorithms before and after this treatment. They measure accuracy, true positive, true negative, false positive and false negative rates. The conclusion of their work is that the pre-university data was not effective and that the grades in linear algebra and calculus subjects were important for predicting the progress of students in the rest of the course. We follow a similar approach, but we consider 130 different courses.

Balaniuk et al. [Balaniuk et al. 2011] address the dropout prediction problem using data from 11,495 students in three courses (Journalism, Law and Psychology) of a higher education institute in Brasilia, Brazil. Three classification algorithms were used to classify students into "dropout"and "graduate", and as input for training the models, they used both socio-economic information and academic information of the students. They concluded that it is possible to identify students with high risk of dropping out with an accuracy of 80.6%.

Manhães et al. [Manhães et al. 2014] propose a similar approach as [Balaniuk et al. 2011] with the key difference that, similarly to us, only features extracted from academic information are used. They used five classification algorithms and data of six courses of the Federal University of Rio de Janeiro, Brazil: Civil Engineering, Mechanical Engineering, Production Engineering, Law, Physics and Pharmacy. Their approach showed accuracy of at least 87% for each course. Our work is very similar to this in terms of the approach used, but we consider only the students of the first semester as classification targets and also consider more courses in our evaluation.

3. PROBLEM FORMULATION

As mentioned in previous sections, we formulate the student's dropout problem as a classification problem. Classification typically considers a set of m-dimensional feature vectors $X \in \mathbb{R}^m$, a set of positive and negative classes $Y = \{+, -\}$ (in our case 'dropout' and 'continue'), and a training set of the form $D^{train} = \{(\vec{x}_1,y_1),\ldots,(\vec{x}_n,y_n)\}$ where $\vec{x}_i$ is a vector of attributes and $y_i \in Y$ represents the class which $\vec{x}_i$ belongs to. The idea is to find a classification function $\hat{y} : X \to Y$ that minimizes the error in the test set $D^{test} = \{(\vec{x}_1,y_1),\ldots,(\vec{x}_p,y_p)\}$, that is unavailable during training, i.e., $D^{test} \cap D^{train} = \emptyset$. More formally, the goal is to minimize:

$$err(\hat{y}; D^{test}) = \frac{1}{|D^{test}|} \sum_{(\vec{x},y)\in D^{test}} l(y, \hat{y}(\vec{x})) \quad (1)$$

where $l : Y \times Y \to \mathbb{R}$ is a loss function measuring, for any test instance $(\vec{x}, y) \in D^{test}$, the misfit between the true y and the predicted value $\hat{y}(\vec{x})$. Since the test is unavailable, the aim is then to minimize the loss in the training assuming that both training and test come from the same population. The specific error functions we use in this paper are defined in Section 5.

4. DATA PREPARATION AND ANALYSIS

The dataset used in our experiments was kindly provided by the administration of UFCG which is also partially sponsoring this research. The dataset consists of academic records of UFCG students from 2002 to 2014 across 130 different courses. This represents 12.5 years of data (or 25 semesters) with around 40,873 students enrolled during this period, from which 38,864 have enrollments in the first semester. From these students, 5,142 have dropped out by the end of the first semester. Table I enumerates and describes the data fields available in this dataset.

As depicted in Figure 2, the percentage of dropouts after the first semester is much lower than the percentage of students who continue in their respective courses. This represents a problem known in the classification literature as class imbalance, a scenario where the classification may be biased towards classifying all test instances with the majority class [He and Garcia 2009].

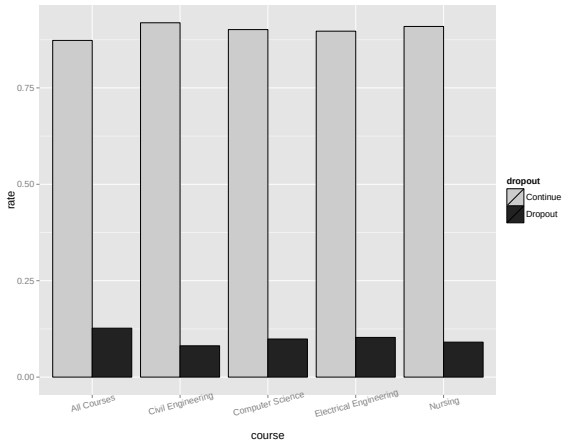


Fig. 2: Percentage of dropouts on all courses combined as well as on some randomly selected courses. Notice that the class imbalance problem affects all the courses, either combined or in isolation.

Column	Description
Enrollment id	Unique identifier of the student
Course id	Unique identifier of the course
Semester id	Identifier of the semester (e.g. 2014.1 means the first semester of 2014)
Entry semester	Semester when the student enrolled in the course
Last semester	Last semester the student was enrolled before dropout
Subject id	Identifier of the subject
Credits	Weight of the subject to the course (based on number of class hours)
Grade	Grade of the student in the subject in the [1,10] range
Situation	Situation of the student in the subject (approved, failed by grade, by attendance or stopped out)
Dropout code	A code that identify the type of dropout (e.g. dropout by abandonment and by conclusion)

Table I: Data fields description.

4.1 Data Preprocessing

Before analysing which features should be used as input in our models, it was necessary to preprocess the data described at Table I in order to deal with the following situations:

—Dropout code. There are several codes used in UFCG to justify/explain the dropout of a student in a course, e.g., dropout by abandonment, by university transfer or by death. In this paper, we

Type/Value	Feature	Description
Id code (string)	Student id	Student's identifier
Id code (string)	Course id	Course's identifier
{1 to n} (numeric)	Semester id	Number of semesters the student has already coursed (in our case always one)
{0 or 1} (string)	Dropout code	Target variable
{0 to n} (numeric)	N.APPR	Number of subjects approved in the semester (subject status approved)
{0 to 10} (numeric)	MEAN.APPR	Average grade of the approved subjects in the semester
{0 to n} (numeric)	N.FAIL	Number of subjects with status equal to fail by attendance and fail by grade
{0 to n} (numeric)	N.ABFL	Number of subjects with status equal to fail by attendance
{0 or 1} (numeric)	STATUS.SEM	The semester status (Defined with 0 if the student failed all subjects, 1 otherwise)
{0 to 10} (numeric)	SEM.MEAN	Mean of all subjects the student is enrolled in the semester
{0 to 10} (numeric)	GPA	Harmonic mean with Grade and Credits in the semester

Table II: Features selected after the Wilcoxon-Mann-Whitney test

Course	STATUS.SEM	N.APPR	N.ABFL	MEAN.APPR	GPA	SEM.MEAN	N.FAIL
Electrical Engineering	0.3670063	0.2818295	0.2140925	0.3670063	0.1951427	0.2608216	0.2803262
Nursing	0.5878980	0.2849662	0.4822433	0.5878980	0.3036385	0.2769646	0.3430081
Computer Science	0.15240790	0.16359879	0.14833515	0.15240790	0.16690989	0.16690989	0.14590964
All courses	0.475435488	0.192050101	0.201618507	0.343461764	0.165301614	0.168236975	0.115688568

Table III: Information Gain

want to predict the students in eminence of dropping out after the first semester, so we mapped all the dropout codes, except for dropout by death, into one class.

—Semester calculation. We calculated the current semester of every student using the semester id and the entry semester just counting the number of semesters that has passed since his/her entry semester until the semester id. In this paper we have used only the data of first semester students.

—Course re-entrance. In most of Brazilian public universities (including UFCG), it is possible for a student to re-enter in the course he/she is already enrolled at through the Brazilian High School National Exam known as ENEM. This results in a new enrollment id where his/her academic records will contain only the subjects in which he/she was approved while using the old id. We handle this situation by identifying these students and creating a new student id aggregating the records spread over all the possible past enrollment ids associated to him/her. This will eliminate the so called fake freshman.

4.2 Feature Selection

Using the resulting dataset from the preprocessing step, we now turn to select the most important features for dropout detection. For doing that, we first considered all the 31 features introduced by Manhães et al. [Manhães et al. 2014]. We then applied the Wilcoxon-Mann-Whitney statistical test on the features considering two samples on the training set: the features associated with instances of class 0 and 1 respectively. If the test results indicate that the features in the different samples come from the same population, we concluded that these features are not discriminative and should be discarded. Table II shows the remaining features after this filtering.

We also computed the information gain ratio of each feature in order to investigate whether the importance of the features vary across the courses.

The results of this analysis are illustrated in Table III. We only show three courses and all the courses combined due to space constraints. We found out that the importance of the features indeed vary across the courses. While STATUS.SEM and MEAN.APPR appear as the most important features (according to information gain) for Electrical Engineering, MEAN.APPR, STATUS.SEM and N.ABFL are the most important for Nursing and GPA and SEM.MEAN for Computer Science. Considering all the courses combined, STATUS.SEM and MEAN.APPR appear as the most important ones which is in line with the intuition that students typically loose interest in their courses when they get low

grades or fail many subjects.

5. EVALUATION

For evaluation we created different train/test splits as follows. We used a sliding window over time where for each considered semester, the students in the first semester of the previous semesters are used as training and the ones in the semester considered as the current one as test. For predicting dropout candidates for the semester 2004.1 (the first semester of 2004), for example, we used all the students in the first semesters of 2002.1, 2002.2, 2003.1 and 2003.2 as training. We did this for all the semesters from 2003.1 to 2013.2.

As mentioned in Section 4 our data suffers from class imbalance. To handle this problem we used random undersampling such that instances of the majority class were randomly discarded until we reached the proportion of 40% of instances of the dropout class and 60% of the other class. In future works we plan to use more sophisticated class imbalance approaches.

For each course and each classification algorithm we selected the subset of features that are significantly better than the others according to information gain. We used the FSelector package of the R project [R Core Team 2015] for statistical computing for doing that[1]. We have evaluated the following well known classification algorithms: Naive Bayes [James et al. 2013], C5.0 [James et al. 2013], SVM [James et al. 2013], Logistic Regression [James et al. 2013] and the Multilayer Perceptron [Russell and Norvig 2010]. With this comparison we want to answer the following research questions:

—RQ1: Does classification pay off for predicting first semester students dropouts?
—RQ2: Are the results consistent considering different classifiers?

We have used e1071[2], C50[3], RSNNS[4] and the stats[5] packages of R to run the algorithms. As evaluation metrics we considered precision, recall (aka true positive rate), f-measure and accuracy. Table IV summarizes the results.

The overall results are encouraging, specially if we consider the recall values that represents the true positive rates, i.e., the percentage of students who were correctly identified as dropouts. Also notice that the accuracy of the classifiers is still higher, in most of the cases, than a classifier that predicts always the majority class (here called Frequent Class) due to the high true positive rates detected by the classifiers.

It is worth noticing that the classifiers used for the "All courses"sample used only STATUS.SEM and MEAN.APPR as predictors. These two features alone are already sufficient for achieving good results, such as more than 70% of F-measure.

In Figure 3 we illustrate the results on the Logit algorithm on all the courses combined per test semester. Note that the results tend to improve in time until some point where they do not improve anymore. This happens because at each subsequent semester we have more training data to use until a point where new training data does not help to improve the results anymore.

Now answering the research questions presented at the beginning of this section, we notice that for RQ1 classification indeed pays off with F-Measure values higher than 70% considering all the courses combined. Concerning RQ2, we notice that the results are consistent considering all the compared classifiers.

[1] https://cran.r-project.org/web/packages/FSelector/
[2] https://cran.r-project.org/web/packages/e1071/e1071.pdf
[3] https://cran.r-project.org/web/packages/C50/C50.pdf
[4] https://cran.r-project.org/web/packages/RSNNS/RSNNS.pdf
[5] https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html

	Algorithm	Precision	Recall	F-measure	Accuracy
Electrical Engineering	Naive Bayes	0.721 ± 0.095	0.719 ± 0.074	0.695 ± 0.061	0.926 ± 0.024
	C5.0	0.721 ± 0.095	0.719 ± 0.074	0.695 ± 0.061	0.926 ± 0.024
	SVM	0.721 ± 0.095	0.719 ± 0.074	0.695 ± 0.061	0.926 ± 0.024
	MLP	0.721 ± 0.095	0.719 ± 0.074	0.695 ± 0.061	0.926 ± 0.024
	Logit	0.677 ± 0.088	0.73 ± 0.075	0.678 ± 0.059	0.919 ± 0.024
	Frequent Class	0	0	-	0.891 ± 0.008
Nursing	Naive Bayes	0.774 ± 0.119	0.827 ± 0.089	0.777 ± 0.070	0.960 ± 0.017
	C5.0	0.72 ± 0.14	0.815 ± 0.102	0.731 ± 0.088	0.943 ± 0.030
	SVM	0.803 ± 0.11	0.835 ± 0.09	0.802 ± 0.072	0.963 ± 0.016
	MLP	0.729 ± 0.128	0.844 ± 0.086	0.754 ± 0.085	0.956 ± 0.016
	Logit	0.687 ± 0.146	0.825 ± 0.077	0.721 ± 0.095	0.939 ± 0.037
	Frequent Class	0	0	-	0.925 ± 0.005
Computer Science	Naive Bayes	0.524 ± 0.102	0.936 ± 0.061	0.647 ± 0.092	0.894 ± 0.035
	C5.0	0.553 ± 0.117	0.932 ± 0.06	0.661 ± 0.098	0.908 ± 0.028
	SVM	0.593 ± 0.109	0.913 ± 0.062	0.694 ± 0.093	0.922 ± 0.026
	MLP	0.585 ± 0.105	0.921 ± 0.063	0.692 ± 0.085	0.917 ± 0.03
	Logit	0.564 ± 0.109	0.923 ± 0.062	0.674 ± 0.093	0.909 ± 0.032
	Frequent Class	0	0	-	0.928 ± 0.009
All courses	Naive Bayes	0.710 ± 0.085	0.823 ± 0.024	0.744 ± 0.048	0.935 ± 0.012
	C5.0	0.710 ± 0.085	0.823 ± 0.024	0.744 ± 0.048	0.935 ± 0.012
	SVM	0.710 ± 0.085	0.823 ± 0.024	0.744 ± 0.048	0.935 ± 0.012
	MLP	0.710 ± 0.085	0.823 ± 0.024	0.744 ± 0.048	0.935 ± 0.012
	Logit	0.710 ± 0.085	0.823 ± 0.024	0.744 ± 0.048	0.935 ± 0.012
	Frequent Class	0	0	-	0.905 ± 0.007

Table IV: Classification results

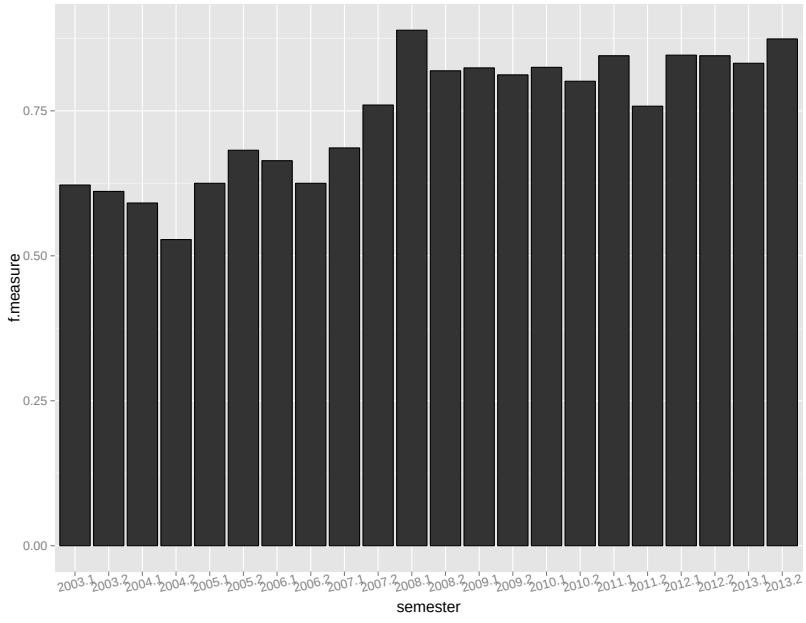


Fig. 3: F-measure per Test Semester considering the Data of all courses.

6. CONCLUSIONS AND FUTURE WORKS

In this paper we cast the students' dropout problem as a classification problem. We evaluated several classifiers on a large sample of academic records from a public Brazilian federal University. To the

best of our knowledge this is the first study considering this amount of data and variety of courses. From this work, we can draw the following important conclusions:

—Classification pays-off for the task of dropout classification showing encouraging results.
—Features extracted from academic records alone carry a strong signal about dropout occurrence.
—Features importance vary according the target course and in many cases a small number of them are sufficient for achieving good results.

As future work, we intend to extend this approach to handle students from other semesters. Our hypothesis is that the set of factors that impacts the identification of students' dropout in the first academic semester might not be the same factors in affecting dropout in subsequent semesters. Therefore, a general model may be created as an hybrid approach of models for each semester. We also intend to investigate the problem per course, since there are variations in terms of the predictors used and results achieved. Finally, we will deploy this model in the Academic Management System of UFCG in order to help administrators, professors and students to identify and prevent dropout.

REFERENCES

ADACHI, A. A. C. T. *Evasão e Evadidos nos Cursos de Graduação da Universidade Federal de Minas Gerais. 2009. 214 f.* Ph.D. thesis, Dissertação (Mestrado em Educação). Faculdade de Educação–Programa de Pós-Graduação em Educação. Universidade Federal de Minas Gerais. Belo Horizonte, 2009.

Andriola, W. B., Andriola, C. G., and Moura, C. P. Opiniões de docentes e de coordenadores acerca do fenômeno da evasão discente dos cursos de graduação da universidade federal do ceará (ufc). *Ensaio: aval. pol. públ. Educ*, 2006.

Balaniuk, R., do Prado, H. A., da Veiga Guadagnin, R., Ferneda, E., and Cobbe, P. R. Predicting evasion candidates in higher education institutions. In *Model and Data Engineering*. Springer, pp. 143–151, 2011.

Barroso, M. F. and Falcão, E. B. Evasão universitária: O caso do instituto de física da ufrj. *ENCONTRO NACIONAL DE PESQUISA EM ENSINO DE FÍSICA* vol. 9, pp. 1–14, 2004.

de Estudos e Pesquisas Educacionais Anísio Teixeira, I. N. Ensino superior mantém tendência de crescimento e diversificação, 2010.

de Estudos e Pesquisas Educacionais Anísio Teixeira, I. N. Censo da educação superior 2013, 2013.

Dekker, G., Pechenizkiy, M., and Vleeshouwers, J. Predicting students drop out: A case study. In *EDM*, T. Barnes, M. C. Desmarais, C. Romero, and S. Ventura (Eds.). www.educationaldatamining.org, pp. 41–50, 2009.

GAIOSO, N. P. d. L. *A evasão discente na Educação Superior no Brasil: na perspectiva de alunos e dirigentes.* Ph.D. thesis, Dissertação (Mestrado)-Universidade Católica de Brasília, Brasília-DF, 2005.[Links], 2005.

He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Trans. on Knowl. and Data Eng.* 21 (9): 1263–1284, Sept., 2009.

James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning*. Springer, 2013.

Manhães, L. M. B., da Cruz, S. M. S., and Zimbrão, G. Evaluating performance and dropouts of undergraduates using educational data mining. In *Proceedings of the Twenty-Ninth Symposium On Applied Computing*, 2014.

Márquez-Vera, C., Cano, A., Romero, C., and Ventura, S. Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Applied intelligence* 38 (3): 315–330, 2013.

Mustafa, M. N., Chowdhury, L., and Kamal, M. Students dropout prediction for intelligent system from tertiary level in developing country. In *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*. IEEE, pp. 113–118, 2012.

Pal, S. Mining educational data to reduce dropout rates of engineering students. *International Journal of Information Engineering and Electronic Business (IJIEEB)* 4 (2): 1, 2012.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

Romero, C. and Ventura, S. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 3 (1): 12–27, 2013.

Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

An Effective Strategy for Feature Selection in High-Dimensional Datasets

Mariana Tasca[1], Alexandre Plastino[1], Celso Ribeiro[1], Bianca Zadrozny[2]

[1] Universidade Federal Fluminense, Brazil
{mlobo, plastino, celso}@ic.uff.br
[2] IBM Research, Brazil
biancaz@br.ibm.com

Abstract. Feature subset selection is an important preprocessing step for the classification task, specially in the case of datasets with high dimensionality, i.e., thousands of potentially predictive attributes. There is an extensive literature on methods for performing FSS, but most of them do not apply to datasets with high dimensionality because of the prohibitive computational cost. This paper proposes a feature subset selection algorithm which is suitable for datasets with high dimensionality. Our proposal is based on the execution of a constructive procedure followed by a local search strategy, in just one iteration. We conducted experiments using a variety of high-dimensional datasets, showing that the proposed method can reach, in most cases, better accuracies – with a much lower computational cost – than some well-known algorithms.

Categories and Subject Descriptors: I.5.2 [**Pattern Recognition**]: Feature evaluation and selection

Keywords: classification, feature selection, high-dimensional datasets

1. INTRODUCTION

One of the most studied and applied tasks in data mining is the classification task, which aims at estimating the class of an instance based on the available set of attributes. One method to improve the performance of the classification process is to perform a feature subset selection (FSS) procedure, an important step in the data mining process, which aims at choosing a subset of attributes that can represent the important information within the data, based on some criteria [Liu and Motoda 1998]. The use of this procedure is strongly recommended, especially if the dataset has a huge dimensionality, because most of the data mining algorithms may require a large computational effort if a large number of attributes is used. The use of an FSS procedure can provide: (a) improvement in classification performance, eliminating useless attributes and those that can deteriorate the results, (b) simpler classification models, reducing the computational cost of executing this models and providing a better understanding of the obtained results, and (c) smaller datasets to be handled.

Because of the exponential (2^n) search space in terms of the number n of attributes, performing FSS through exhaustive search is intractable. For this reason, several approximation strategies were proposed to solve this problem. FSS algorithms are composed of a search method and a strategy to evaluate the candidate solution [Liu and Yu 2005]. There are a number of different search strategies such as ranker, sequential search, incremental search and metaheuristics, which are reviewed in the next section. The evaluation of candidates can be performed in two ways: the *filter* approach, which uses a relevance measure to estimate the goodness of attributes or sets of attributes, and the *wrapper* approach, which estimates the merit of candidates by the accuracy values obtained using a classifier.

This work was supported by CAPES, CNPq and FAPERJ research grants.

We present in this work a feature subset selection algorithm – called Local Search Based (LSB) strategy. LSB combines a construction phase followed by a local search, in only one iteration. Because of the reduced number of evaluations of candidate solutions, this strategy is well-suited to high-dimensional datasets, where some of the most popular FSS methods cannot be applied because of prohibitive computational costs. The information gain of individual attributes is used in the first phase of LSB to produce a ranking of attributes. Based on this information, an initial candidate solution is generated and, in the next step, its neighborhood is explored in order to find better solutions.

This paper is organized as follows: Section 2 presents previous work on FSS. Section 3 describes the proposed algorithm. In Section 4, the experiments conducted over nine datasets are showed, and the analysis about the results is presented. Finally, the conclusions about this work and some ideas for the future are discussed in Section 5.

2. PREVIOUS WORK ON FEATURE SELECTION

There are some different strategies in the literature which can be applied to the feature subset selection purpose.

Ranker approaches take into account the individual merit of attributes (with respect to their capacity of identifying the class) to create a ranking of attributes [Blum and Langley 1997; Guyon and Elisseeff 2003]. The first k attributes of the resulting ranking are selected to compose the candidate solution. These algorithms are very fast (linear complexity in terms of dataset dimensionality), but because interactions between attributes are not considered, the quality of candidates may be degraded. Moreover, it may be difficult to select an ideal value for k.

Sequential search algorithms are very simple: at each iteration, the inclusion/exclusion of each attribute is evaluated and those that generate the highest improvement in the solution quality are added/removed. Thus, the complexity of worst case is $O(n^2)$. The most common sequential strategies are *Sequential Forward Selection* (SFS) – which starts with an empty solution and adds attributes one by one – and *Sequential Backward Selection* (SBS) – which starts with all attributes and removes one by one [Kittler 1978].

Incremental search strategies also add one attribute per iteration. However, these algorithms use an initial ranking of the attributes, based on their individial merit. Thus, at each iteration, the attribute at the top of the ranking is selected to be added in the candidate solution, and only this new candidate is evaluated ($O(n)$ complexity) [Ruiz et al. 2006; Bermejo et al. 2010].

Metaheuristics like GRASP [Feo and Resende 1995; Resende and Ribeiro 2014], Tabu Search [Glover and Laguna 1997], Genetic [Goldberg 1989] and Memetic algorithms [Moscato 2003] have been used in the FSS context. For many optimization problems, metaheuristic techniques have proved to be very effective and feasible. However, their computational cost may be extremely high in the context of high-dimensional datasets. Some FSS algorithms that employ metaheuristic approaches can be found in [Yang and Honavar 1998; Inza et al. 2000; Yusta 2009; Esseghir 2010]. Despite the good results achieved, those approaches were applied to low-dimensional datasets (less than 100 attributes).

Some of the most popular FSS methods cannot be applied to high-dimensional datasets because of prohibitive computational costs. For example, for methods that are based on *wrapper* approaches, which require execution of the classifier for each candidate evaluation, it may become infeasible to execute a large number of evaluation steps.

In the last few years, some hybrid algorithms which combine *filter* and *wrapper* approaches have been proposed with the idea of reducing the number of attributes before the *wrapper* evaluation. Some of these approaches can be found in: [Ruiz et al. 2006; Flores et al. 2008], which incrementally explores the attributes by following the ranking obtained by a *filter* measure; [Gutlein et al. 2009],

which applies a *wrapper* sequential forward search but only over the first k attributes in the *filter* ranking; [Bermejo et al. 2010; Ruiz et al. 2008], which uses the *filter*-based ranking for a better organization of the search process; [Bermejo et al. 2011], which presents a GRASP with the main goal of speeding up the FSS process, by reducing the number of *wrapper* evaluations to carry out; [Bermejo et al. 2014], which proposes to embed the classifier into the FSS algorithm instead of using it as a black-box only for evaluating the candidate solutions; and [Moshki et al. 2015], which proposes a GRASP with an extended version of a simulated annealing algorithm for local search. Our strategy also follows a *filter-wrapper* approach in the sense that we use a *filter* in the constructive phase (to rank the attributes and than proceed a pruning on the original list) and we use the *wrapper* to evaluate the candidate solutions (both in construction phase and in the local search).

3. THE PROPOSED ALGORITHM

The proposed heuristic – LSB – is a combination of a construction procedure and a local search. On the initialization, two steps are performed: (i) the list of attributes E from the dataset are ranked by an individual relevance measure and (ii) the generated ranking is prunned so that only the first k attributes from the ranking (represented by R) are considered in the next phases. The value of k is controled by a parameter p which defines a percentage of the whole list of attributes. This pruning step is necessary on the context of high-dimensional datasets, because the evaluation of the attribute selection algorithm with the whole set of attributes is impracticable.

The construction phase produces a viable solution S from the prunned ranking R. S is represented by a vector $S[i]$, $1 \leq i \leq |R|$, where if $S[i]$=0, it means that the *i-th* attribute from R does not belong to S; on the other hand, if $S[i]$=1, the *i-th* attribute belongs to the solution S. Then the local search phase tries to improve the quality of S by searching for better neighbors in the $N(S)$ neighborhood. This combination *construction + local search* is executed only once and the final solution is the best neighbor found in the local search procedure. Pseudo-code of the proposed algorithm is presented in Figure 1.

In line 01, a ranking E' of the attributes from E is generated. The evaluation function used to evaluate the individual attributes was Information Gain [Quinlan 1993], since it is a well-known measure in the context of feature selection. In line 02, the number of attributes k which will be considered for the algorithm is calculated as $p\%$ of the total number of attributes on the dataset. Line 03 represents the pruning step. R is filled with the first k attributes from the ranked list E'. This step speeds up the algorithm since a reduced number of attributes (k) are considered in the constructive and local search phases.

In line 04, S is iniciated with an empty subset. The loop represented in lines 05 to 14 performs the construction of a solution by traversing all the elements of $|R|$.

In line 06, a restricted candidate list (RCL) is generated. The RCL is a list of attributes whose fitness belongs to the range $[max - \alpha * (max - min), max]$, where min and max are the lowest and highest fitness values to R, respectively, and α is a parameter which controls the size of this restricted list. In line 07, one attribute e is randomly selected from RCL to be incorporated, in line 08, in the current solution S. In line 09, the current solution is evaluated by a *wrapper* strategy, using the Naive Bayes classifier, with internal 5-fold cross-validation.

In lines 10 to 12, the fitness of the new solution S' is compared with the fitness of S. If S' outperforms S, it becomes the current solution S. The last step of the iteration is presented in line 13, when the evaluated attribute e is removed from R.

For the local search procedure (lines 15 to 24), the solution S generated by the constructive phase is taken as starting point. While a complete iteration of the local search finds a neighbor $S_i \in N(S)$ which outperforms S, a new iteration is performed by taking the best neighbor S_i as the current

```
procedure LSB(E, dataset, p, α)

//Initialization
01.  E' ← ranking of atributes from E;
02.  k ← |E|*p/100;
03.  R ← first k attributes from E' //(pruning step);

//Constructive phase
04.  S ← φ;
05.  while R <> φ do
06.    Generates RCL from R based on α;
07.    e ← randomly selected attribute from RCL;
08.    S' ← e ∪ S;
09.    f(S') ← fitness of solution S';
10.    if f(S') > f(S) then
11.      S ← S';
12.    end if;
13.    R ← R − e;
14.  end while;

//Local search phase
15.  do
16.  LS-improvement ← false;
17.    for each S_i ∈ N(S) do
18.      if f(S_i) > f(S) then
19.        S ← S_i;
20.        LS-improvement ← true;
21.      end if
22.    end for
23.  while LS-improvement is true;
24.  return S;
end.
```

Fig. 1. Pseudo-code of the proposed feature subset selection heuristic

solution S. The neighborhood $N(S)$ used is made up of all the n subsets $\{S_1,S_2,...S_n\}$, n=$|R|$, where the *i-th* bit $S_i[i]$, $1 \leq i \leq n$, is inverted. In other words, if $S[i]$=0, then the neighbor $S_i[i]$=1 and viceversa. This type of neighborhood takes into account insertions (when $S[i]$ is inverted from 0 to 1) and removals (when $S[i]$ is inverted from 1 to 0) of attributes in S. When none of the neighbors $S_i \in N(S)$ presents $f(S_i) > f(S)$, the local search ends and returns, in line 24, the best-fitness solution found.

4. EXPERIMENTS AND RESULTS

Datasets used in the experiments were obtained from public repositories and have hundreds or thousands of attributes. Table I presents these datasets showing their dimensionality and the number of instances. Datasets are split in 10 folds to enable an external 10-fold cross-validation. Thus, accuracy values for each experiment represent the average of 10 executions of the algorithm for the same dataset.

Since the algorithm uses a random function during the constructive phase to select an attribute from the RCL, it is necessary to define an initial seed value for each execution. We conducted 10 independent executions of each experiment, with 10 different initial seeds. Thus, the presented values in the next section represent the average of 10 independent executions on each dataset, each of them using a 10-fold cross-validation.

We tried some different values for the α and p parameters, and the best results were produced when α=0.2 and p=2%.

Table I. High-dimensional datasets used in the experiments

Dataset	# of attributes	# of instances
Leukemia	7130	72
DLBCL	4027	47
Lymphoma	4027	96
Madelon	500	2600
Colon	2001	62
Dexter	19999	600
Lung	12534	181
Prostate	12600	136
Gisette	5000	6000

WEKA (Waikato Environment for Knowledge Analysis) [Hall et al. 2009] is a powerful open-source Java-based machine learning workbench. Among the techniques available within WEKA, we selected four feature subset selection algorithms to make a comparision with LSB: *Best-First* (BF), *LinearForwardSelection* (LF), *SubsetSizeForwardSelection* (SS) and *RankSearch* (RS). The first three algorithms follow the sequential approach and the last one follows the incremental approach. Our aim is to compare LSB with some well-known available algorithms for feature selection.

Best-first [Ginsberg 1994; Russell and Norvig 2003] searches the space of attribute subsets by greedy hill-climbing augmented with a backtracking facility. The idea is to select the most promising candidate generated which has not already been expanded. The backtracking level is controled by a parameter which defines the number of non-improving candidates allowed. *Best-first* may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward. It is also possible start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point).

In the classical *Sequential Forward Selection* approach, the number of evaluations grows quadratically with the number of attributes: the number of evaluations in each step is equal to the number of remaining attributes that are not in the currently selected subset. This quadratic growth can be problematic for datasets with a large number of attributes. Trying to mitigate this problem, [Gutlein et al. 2009] propose a technique to reduce the number of attribute expansions in each forward selection step. *LinearForwardSelection* is an extention of *BestFirst*. In this approach, they limit the number of attributes that are considered in each step so that it does not exceed a certain user-specified constant. This drastically reduces the number of evaluations, and therefore improves the runtime performance of the algorithm.

SubsetSizeForwardSelection is an algorithm which determines the subset size to be reached in forward selection to combat overfitting, where the search is forced to stop at a precomputed subset size. In [Gutlein et al. 2009], they show that this technique reduces subset size while maintaining comparable accuracy with the *LinearForwardSelection* approach.

Rank-Search [Hall and Holmes 2003] is a forward search approach which works in two steps. In the first step, all attributes are ranked by a *filter* or a *wrapper* method. In the second step, the algorithm builds n attribute subsets: the first set is the top-ranked attribute, followed by the two top-ranked attributes, the three top-ranked attributes, and so on. These subsets are evaluated using the *wrapper* or a *filter* method that can evaluate sets of attributes.

Some different parameter combinations were tested for each of these algorithms. The combination with the best performance (regarding the solution quality) for each one was used in the comparison with LSB. It was not possible to conclude the experiments with the RS algorithm for two datasets,

because they exceeded the 10 hours per fold time limit, defined for these experiments. For this reason, results for RS were not reported, because we considered that it showed not to be suitable for high-dimensional datasets. For *BestFirst* and *LinearForwardSelection* algorithms, the best accuracy values were obtained with the default parameters from WEKA. For *SubsetSizeForwardSelection*, we performed a ranking using the *wrapper* with Naive Bayes and considered 100 attributes from this ranking.

At first, we analyzed the accuracy values obtained with Naive Bayes (NB) [Duda and Hart 1973] classifier, by submitting the selected subset by each evaluated algorithm. NB is a probabilistic classifier based on the assumption of conditional independence among the predictive attributes given the class. In spite of this hard independence assumption, NB is a competitive classifier, working quite well in many classification tasks [Fang 2013].

Table II presents the accuracy values obtained in this experiment. Values in brackets represent the position in the ranking which compares the four algorithms, for each dataset. The best accuracies in each line are marked in bold. The last row in the table presents the sum of the ranking positions (SRP) for each strategy. Considering that position 1.0 represents the best accuracy for the given dataset and position 4.0 represents the worst result, the optimum value for SRP would be 9.0 (when the algorithm is the top ranking for all datasets) and the worst value would be 36.0 (when the algorithm gets the fourth position for all datasets). LSB presented the best behaviour among the evaluated algorithms, as it has the lowest sum of ranking positions (SRP).

Table II. Accuracy values obtained for each evaluated algorithm

Dataset	BF	LF	SS	LSB
Leukemia	88.57 (4.0)	91.61 (2.0)	90.00 (3.0)	**95.15** (1.0)
Dlbcl	80.00 (4.0)	88.00 (2.5)	88.00 (2.5)	**88.36** (1.0)
Lymphoma	78.00 (2.0)	74.89 (3.0)	70.67 (4.0)	**79.14** (1.0)
Madelon	61.19 (3.0)	60.12 (4.0)	61.23 (2.0)	**61.34** (1.0)
Colon	80.48 (3.0)	**83.81** (1.0)	83.81 (2.0)	76.67 (4.0)
Dexter	81.67 (4.0)	84.33 (3.0)	85.17 (2.0)	**88.24** (1.0)
Lung	94.53 (4.0)	97.25 (2.0)	95.58 (3.0)	**98.73** (1.0)
Prostate	71.92 (3.0)	73.46 (2.0)	70.60 (4.0)	**79.80** (1.0)
Gisette	**93.80** (1.0)	88.25 (4.0)	89.03 (3.0)	90.17 (2.0)
Sum of Ranking Positions (SRP)	28.0	23.5	25.5	13.0

To analyze if the results are statistically significant, we applied the non-parametric *Friedman* test [Friedman 1937], which enables a multi-algorithm multi-dataset comparision. The null-hypothesis for the *Friedman* test is that there are no differences between the algorithms. If the null-hypothesis is rejected, we can conclude that at least two of the algorithms are significantly different from each other, and the *Nemenyi* post-hoc test can be applied to identify these differences [Demšar 2006]. According to the *Nemenyi* test, the performances of two algorithms are significantly different if their corresponding medium ranking are different at least for a determined critical value.

Friedman test results for the evaluated algorithms (p-$value$ = 0.0336) rejected the null-hypothesis, so the *Nemenyi* test was performed (critical value = 1.5634) and detected a significant difference between BF and LSB, LF and LSB, and between SS and LSB, which shows that LSB outperforms the other three algorithms with statistical significance.

We also ranked the strategies computational costs based on the CPU time. Figure 2 presents the sum of the ranking positions for each evaluated strategy. LSB also obtained the best result on this issue. For the nine evaluated datasets, LSB obtained the first position, taking the shortest CPU time to perform the FSS.

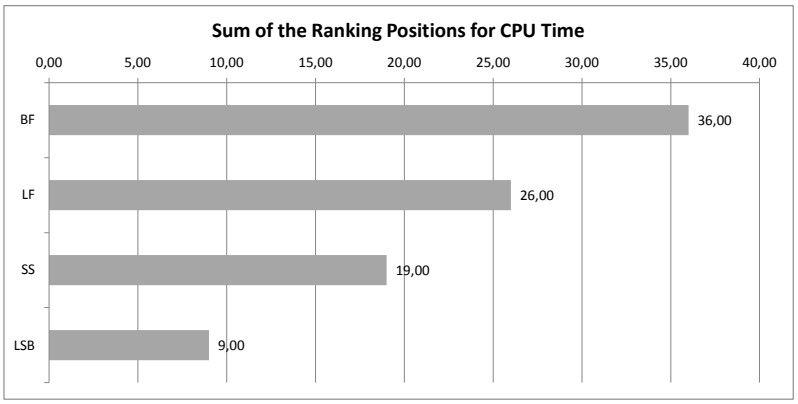


Fig. 2. Evaluation of Computational Time

With respect to the size of selected subsets, LSB proved to be very efficient to reduce the datasets dimensionality, in the same manner as the other evaluated algorithms. The selected subsets represent an average of 0.24% of all dataset attributes. BF, LF and SS generated, respectively, solutions with an average size of 0.37%, 0.49% and 0.19% of all dataset attributes.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we proposed a feature subset selection algorithm, based on a simple combination of a construction procedure and a local search phase. We are focused in the context of high-dimensional datasets, since the most popular FSS methods are not applied to this context, specially if the evaluation methods of candidates are based on *wrapper* approaches. Our proposal aims at simplicity and efficiency, generating solutions which produces good accuracies and reducing significantly the number of attributes in the dataset, with a low computational cost.

We have compared LSB with some important available FSS algorithms in WEKA over nine high-dimensional datasets. Results showed that LSB is a very competitive proposal. It produces, in most cases, better accuracies with a lower computational cost. We are already working on a comparative study among our algorithm and other more recent and sophisticated approaches for feature subset selection in high-dimensional datasets, like the one proposed in [Bermejo et al. 2011].

For future work, one idea is to investigate some parameter modifications, like changing the relevance measure for generating the initial ranking of attributes and trying different percentual values for the pruning step. We also intend to investigate some technique to intensificate the search, like path relinking [Glover 1996], aiming at finding better solutions.

REFERENCES

Bermejo, P., Gámez, J. A., and Puerta, J. M. Improving incremental wrapper-based feature subset selection by using re-ranking. In *Proceedings of the 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems - Volume Part I*. IEA/AIE'10. Springer-Verlag, Berlin, Heidelberg, pp. 580–589, 2010.

Bermejo, P., Gámez, J. A., and Puerta, J. M. A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. *Pattern Recognition Letters* 32 (5): 701–711, 2011.

Bermejo, P., Gámez, J. A., and Puerta, J. M. Speeding up incremental wrapper feature subset selection with naive bayes classifier. *Knowledge-Based Systems* vol. 55, pp. 140–147, jan, 2014.

Blum, A. L. and Langley, P. Selection of relevant features and examples in machine learning. *Artificial Intelligence* vol. 97, pp. 245–271, 1997.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* vol. 7, pp. 1–30, dec, 2006.

Duda, R. O. and Hart, P. E. *Pattern Classification and Scene Analysis*. John Willey & Sons, New Yotk, 1973.

Esseghir, M. A. Effective wrapper-filter hybridization through GRASP schemata. *Journal of Machine Learning Research - Proceedings Track* vol. 10, pp. 45–54, 2010.

Fang, X. Inference-based naive bayes: Turning naive bayes cost-sensitive. *Knowledge and Data Engineering, IEEE Transactions on* 25 (10): 2302–2313, oct, 2013.

Feo, T. and Resende, M. Greedy randomized adaptive search procedures. *Journal of Global Optimization* vol. 6, pp. 109–133, 1995.

Flores, M. J., Gámez, J. A., and Mateo, J. L. Mining the ESROM: A study of breeding value classification in manchego sheep by means of attribute selection and construction. *Computers and Electronics in Agriculture* 60 (2): 167–177, 2008.

Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32 (200): 675–701, dec, 1937.

Ginsberg, M. *Essentials of Artificial Intelligence*. Morgan Kaufmann Pub. Inc., San Francisco, CA, USA, 1994.

Glover, F. Tabu search and adaptive memory programing: Advances, applications and challenges. In *Interfaces in Computer Science and Operations Research*. Kluwer, Dallas, TX, EUA, pp. 1–75, 1996.

Glover, F. and Laguna, M. *Tabu Search*. Kluwer Academic Pub., Norwell, MA, USA, 1997.

Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, USA, 1989.

Gutlein, M., Frank, E., Hall, M., and Karwath, A. Large-scale attribute selection using wrappers. In *CIDM-2009 IEEE Symposium on Computational Intelligence and Data Mining*. Nashville, TN, USA, pp. 332–339, 2009.

Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research* vol. 3, pp. 1157–1182, mar, 2003.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The WEKA data mining software: an update. *SIGKDD Explorations Newsletters* 11 (1): 10–18, nov, 2009.

Hall, M. A. and Holmes, G. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering* 15 (3): 1437–1447, 2003.

Inza, I., Larrañaga, P., Etxeberria, R., and Sierra, B. Feature subset selection by bayesian network-based optimization. *Artificial Intelligence* 123 (1-2): 157–184, oct, 2000.

Kittler, J. Feature set search algorithms. *Pattern Recognition and Signal Processing*, 1978.

Liu, H. and Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Pub., USA, 1998.

Liu, H. and Yu, L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering* vol. 17, pp. 491–502, 2005.

Moscato, P. A gentle introduction to memetic algorithms. In *Handbook of Metaheuristics*. Kluwer Academic Pub., pp. 105–144, 2003.

Moshki, M., Kabiri, P., and Mohebalhojeh, A. Scalable feature selection in high-dimensional data based on GRASP. *Applied Artificial Intelligence* 29 (3): 283–296, mar, 2015.

Quinlan, J. R. *C4.5: programs for machine learning*. Morgan Kaufmann Pub. Inc., San Francisco, CA, USA, 1993.

Resende, M. and Ribeiro, C. GRASP: Greedy randomized adaptive search procedures. In *Search Methodologies*, E. K. Burke and G. Kendall (Eds.). Springer US, pp. 287–312, 2014.

Ruiz, R., Riquelme, J. C., and Aguilar-Ruiz, J. S. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition* 39 (12): 2383–2392, 2006.

Ruiz, R., Riquelme, J. C., and Aguilar-Ruiz, J. S. Best agglomerative ranked subset for feature selection. *JMLR Workshop Conference Proceedings* vol. 4, pp. 148–162, 2008.

Russell, S. J. and Norvig, P. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

Yang, J. and Honavar, V. G. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems* 13 (2): 44–49, mar, 1998.

Yusta, S. C. Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters* 30 (5): 525–534, apr, 2009.

Preparação de Dados Longitudinais: Estudo de Caso em Envelhecimento Humano

Caio Eduardo Ribeiro[1], Luis Enrique Zárate[1]

Pontifícia Universidade Católica de Minas Gerais, Brazil
caioedurib@gmail.com
zarate@pucminas.br

Abstract. O sucesso do processo de descoberta de conhecimento em bases de dados depende de uma preparação adequada da base de dados. Em estudos longitudinais, que acompanham um conjunto fixo de registros ao longo de um período de tempo, é recomendado atenção às características especiais acrescentadas às bases de dados pela adição do eixo do tempo, que provocam novas restrições e permitem abordagens diferentes na preparação da base de dados. Este trabalho mostra um estudo de caso com uma base de dados real, de um procedimento de preparação de bases de dados longitudinais, englobando desde a formação da base até o final de sua preparação.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Data Mining

Keywords: data mining, knowledge discovery, preprocessing

1. INTRODUÇÃO

O processo de descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases* - KDD) se inicia com a preparação dos dados, etapa importante para garantir que os resultados do processo sejam satisfatórios. Uma preparação adequada da base de dados reduz a distorção dos dados, auxilia no desempenho dos algoritmos de mineração de dados, e colabora para resultados mais valiosos e confiáveis dos processos de KDD [Pyle 1999].

O intuito de compreender fenômenos que ocorrem com o passar do tempo traz a necessidade de acrescentar às bases de dados, tipicamente bidimensionais (registros e atributos), um aspecto temporal. O paradigma de bases de dados temporais traz novos desafios para o processo de KDD, pois os padrões a serem descobertos se encontram, também, nas informações trazidas pelo eixo do tempo [Antunes and Oliveira 2001]. Um subconjunto dos estudos temporais são os estudos longitudinais, onde os mesmos registros são acompanhados ao longo do tempo, para caracterizar determinados aspectos de sua evolução [Diggle et al. 2002]. A preparação de uma base de dados longitudinal precisa considerar os aspectos temporais do estudo e os objetivos da pesquisa.

Um domínio no qual estudos longitudinais são frequentemente utilizados é o estudo do envelhecimento humano, considerado como estudo de caso neste trabalho. Com o aumento da população idosa no mundo, há um maior interesse na criação de políticas públicas, descoberta de hábitos saudáveis, e em programas sociais para aumentar o bem-estar da população idosa. Portanto, a demanda por conhecimento acerca do envelhecimento tem aumentado nos últimos anos [Malloy-Diniz et al. 2013]. Estudos longitudinais sobre o envelhecimento objetivam acompanhar uma conjunto fixo de pessoas ao longo de vários anos, e estabelecer relações entre as evoluções dos atributos da base de dados e as mudanças na vida dessas pessoas. O uso de técnicas de mineração de dados pode ajudar a extrair con-

hecimento importante das bases longitudinais, embora a grande maioria dos estudos utilizem apenas análises estatísticas para inferir e testar hipóteses, o que revela que há uma carência de fundamentos da área de mineração de dados para esse tipo de base de dados. Estudos de regressão são comuns para testar hipóteses do tipo causa e efeito, mas raramente são encontrados trabalhos que utilizam a Mineração de Dados para encontrar esses padrões, apesar da necessidade de análises mais compreensivas quando o objeto de estudo é complexo e tem atributos altamente dependentes, como o caso do estudo do envelhecimento humano [Ribeiro and Zárate 2014].

O objetivo deste trabalho é discutir procedimentos a serem aplicados na etapa de preparação de bases de dados longitudinais abordando, na perspectiva da mineração de dados, a montagem da base, seleção de atributos, limpeza dos dados, e a discretização e junção de atributos. Como estudo de caso, foi feita a preparação de uma base nominal do estudo longitudinal sobre o envelhecimento humano *English Longitudinal Study of Ageing*, do Reino Unido, com cerca de 12000 registros e 5000 características para cada onda do estudo. A base foi reduzida para um estudo de Mineração de dados longitudinais, tendo como dimensões após a aplicação da metodologia 5352 registros, 255 características, em 5 ondas.

Na seção seguinte, são apresentados os conceitos considerados neste trabalho. Em seguida, em uma única seção, são discutidas as etapas do procedimento sugerido para este tipo de base de dados, e a aplicação destas no estudo de caso acerca do envelhecimento humano, para facilitar a compreensão prática de como os métodos devem ser utilizados. Finalmente, são apresentadas as conclusões e revisadas as contribuições do trabalho.

2. REVISÃO TEÓRICA

2.1 Mineração de Dados Longitudinais

A mineração de dados (*Data Mining* - DM), como área da computação, surgiu da necessidade de extrair conhecimento de bases de dados extensas através de algoritmos que as analisam automaticamente, com ou sem a supervisão de profissionais, descobrindo padrões úteis e não-triviais nos dados [Kantardzic 2011]. Os padrões encontrados por algoritmos de DM representam informações que, ao serem interpretadas por especialistas, se tornam conhecimento útil para tomada de decisão. A mineração de dados pode ser definida como uma das etapas do processo de descoberta de conhecimento em bases de dados, sendo precedida por uma série de etapas de preparação, cruciais para o sucesso do processo [Fayyad et al. 1996].

Além dos desafios de explorar corretamente o espaço-problema e propor um modelo adequado, para estudos temporais, o processo de mineração possui outros aspectos a serem considerados. Acrescentar o aspecto temporal aumenta o volume e a complexidade dos dados, e o foco dos estudos geralmente está na reação causal entre um efeito observado e a evolução dos valores para o conjunto de atributos da base [Last et al. 2001].

2.2 Preparação de Bases de Dados

Neste trabalho, são abordados como etapas da preparação de dados todas aquelas realizadas entre a definição do objetivo do estudo e a aplicação dos algoritmos de Mineração de Dados, ou seja, a seleção de atributos, pré-processamento e transformações dos dados. Os procedimentos propostos abordam técnicas para preparação de bases com dados nominais e/ou contínuos, de alta dimensionalidade, a serem usadas em um estudo de mineração de bases longitudinais.

O processo de preparação dos dados visa garantir que estes sejam os mais relevantes possíveis para o conhecimento que se objetiva obter, representem corretamente a realidade, e estejam presentes em uma quantidade suficiente de registros, para garantir que os resultados sejam aplicáveis [Kotsiantis and et al. 2006]. O processo de seleção dos atributos relevantes, parte desse processo, é penoso devido

ao grande número de combinações possíveis, e existem várias estratégias para realizá-lo, como filtros e *wrappers* [Paes et al. 2013].

Como o objetivo deste trabalho é utilizar apenas dados nominais (categóricos), os atributos que possuem valores contínuos precisam passar por um processo de discretização. Esse processo se dá através de divisão dos valores contínuos em intervalos discretos, e na rotulagem desses intervalos, processo no qual o conhecimento representado pelo atributo deve ser mantido, na medida do possível [Garcia et al. 2013]. A discretização de atributos tem um custo implícito de perda de precisão, mas possibilita o uso de vários métodos de análise que exigem dados nominais como entrada.

3. PROCEDIMENTO E ESTUDO DE CASO

Bases de dados passam por uma série de filtragens e transformações, antes de serem usadas como dados de entrada em um algoritmo de mineração de dados, e existem procedimentos clássicos a serem seguidos durante o processo de KDD. A diferença para bases temporais é que elas possuem várias versões, possivelmente com configurações diferentes, para cada unidade de tempo considerada em sua construção, e isso deve ser levado em consideração durante o processo. Para que uma análise longitudinal seja possível, uma base deve ter os mesmos registros e atributos em todas as suas versões, e é desejável que apenas dados relevantes para o estudo sejam mantidos na base.

Com o objetivo de preparar uma base com características temporais para estudos longitudinais, este trabalho apresenta um procedimento replicável que aborda desde a formação da base até o final de sua preparação. O procedimento proposto pode ser dividido em cinco etapas, que serão descritas nesta seção, juntamente com a descrição do estudo de caso realizado para testá-lo. A Figura 1 ilustra essas etapas, que serão abordadas individualmente após a descrição da base de dados, discutindo-se as características especiais que essas técnicas recebem quando se prepara uma base de dados longitudinal.

Fig. 1. Etapas do procedimento proposto. Fonte: Elaborado pelo autor.

3.1 Descrição da Base de Dados

O UK Data Service é uma fundação especializada no arquivamento de dados, que abriga a maior coleção de dados digitais de estudos sociais e humanos do Reino Unido. O centro reúne dados de pesquisas de escala nacional, censos, empresariais, estudos qualitativos, entre outros, que são disponibilizados para pesquisadores associados. Um dos estudos arquivados no UK Data Service é o *English Longitudinal Study of Ageing* (ELSA), uma pesquisa feita em ondas com intervalos de dois anos, e o objetivo de monitorar diversos aspectos da vida dos participantes, através de entrevistas detalhadas realizadas por profissionais [Marmot 2013]. A base do ELSA foi a escolhida para este trabalho porque não existem bases brasileiras sobre o tema disponíveis para estudos acadêmicos.

O estudo teve início oficialmente em 2002, e as características abordadas incluem dados demográficos, econômicos, de saúde física, mental e psicológica, vida social e funções cognitivas. A maior parte das questões possui opções de resposta predeterminadas, tornando a base predominantemente nominal. O ELSA é voltado para pessoas com 50 anos de idade ou mais, com o intuito de acompanhar os participantes pelos anos que precedem sua aposentadoria e adiante, permitindo uma análise detalhada da evolução dos aspectos observados [Banks 2006]. Após uma solicitação formal e cadastro

no serviço, a base e sua documentação foram disponibilizadas para *download*, e adquiridas no formato de tabelas *MapInfo TAB File*, e documentos PDF e RTF. A base recebeu um tratamento antes de ser disponibilizada, de forma que não hajam dados ausentes ou fora dos padrões preestabelecidos, casos nos quais o atributo recebe um valor de erro cujo significado é informado na documentação do estudo.

3.2 Montagem da Base Longitudinal

Uma base de dados longitudinal é uma base temporal com a mesma identidade para todas as unidades de tempo. Define-se como longitudinal uma base que pode ser descrita como uma matriz M composta pelo produto cartesiano de três vetores: r (de registros), a (de atributos), e t (de unidades de tempo), como mostra a Equação 1. A representatividade de uma base de dados está relacionada à sua dimensionalidade, afetada pelo tamanho de cada um desses vetores. Dois aspectos devem ser observados na montagem de bases de dados longitudinais: o aspecto da evolução intrínseca e o aspecto da conformação da base de dados.

$$M_{rat} \quad = \quad [r] \times [a] \times [t] \tag{1}$$

(1) Observando o aspecto evolutivo da base de dados
Apesar de ser um estudo longitudinal, a base original do ELSA apresenta discrepâncias entre os registros e atributos, nas ondas do estudo. Isso ocorre porque o estudo evolui à medida que as ondas vão sendo estudadas, e modificações no questionário são recomendadas pelos especialistas envolvidos no projeto, e porque novos participantes são incluídos na pesquisa ao atingirem os requisitos para participarem do estudo. As modificações no questionário buscam adequar o escopo do estudo de acordo, principalmente, com hipóteses que são formadas através do estudo das ondas anteriores, que exigem um nível maior de detalhemento de algumas informações para poderem ser confirmadas ou refutadas. Quanto aos registros, o estudo é voltado para indivíduos com 50 anos ou mais, e a parcela mais jovem do estudo ficaria sem representantes em ondas futuras, se não houvesse uma revigoração da base de respondentes a cada onda. Para que um estudo longitudinal da base do ELSA fosse possível, a primeira etapa da preparação dos dados é um pré-processamento que tornará a base de dados utilizável para estudos longitudinais, retirando registros e atributos inconsistentes.

(2) Observando o aspecto de conformação da base de dados
Para a filtragem dos registros, foram selecionados apenas os integrantes das cinco ondas do ELSA que são abordadas neste estudo (2002-2010). Os registros mantidos na base de dados são, portanto, indivíduos que responderam aos questionários de todas as cinco ondas que constituem o estudo. A filtragem dos atributos não pôde ser feita da mesma forma, porque algumas questões sofreram pequenas alterações nas opções de resposta, que poderiam ser revertidas com uma recodificação dessas opções, e descartar essas questões por causa das diferenças ocasionaria grandes perdas de informação. Por exemplo, se uma questão tem cinco opções de resposta nas três primeiras ondas e seis nas duas últimas, e for possível mapear essas seis opções de reposta nas cinco iniciais, a questão pode ser utilizada no estudo depois das devidas alterações na base de dados. A filtragem inicial manteria todas as questões utilizáveis dos questionários das cinco ondas, realizando recodificações nas opções de resposta, quando considerado que a perda de informação seria aceitável.

Ao final da primeira etapa da metodologia, os dados brutos do ELSA foram transformados em uma base de dados utilizável para um trabalho longitudinal. Foram mantidos os atributos que se repetem (com as possíveis alterações no texto da questão, e/ou suas opções de resposta devidamente tratadas), e os registros dos participantes que responderam todas as ondas. Ao final dessa filtragem inicial, a base resultante tinha 5 ondas com 5352 registros e 1693 atributos cada.

3.3 Seleção Conceitual de Atributos

A etapa de seleção visa selecionar na base os dados atributos relevantes para a execução do estudo. A seleção implica na remoção de atributos menos relevantes, o que reduz a dimensionalidade da base, diminuindo a complexidade de execução dos algoritmos de mineração, e ajuda a garantir que o conhecimento gerado pelo processo seja compreensível e relevante. Conhecimento sobre a área do problema em estudo é crucial para esse tipo de definição, pois os atributos são julgados por um processo de pré-seleção de acordo com o entendimento do problema sendo tratado.

No estudo de caso, para que a escolha dos atributos relevantes fosse a mais precisa possível, primeiramente foi necessário definir o que compõe o ambiente no estudo do envelhecimento humano. A Revisão Sistemática da Literatura realizada em [Ribeiro and Zárate 2014] gerou um Modelo Conceitual, que foi usado como guia nessa etapa do trabalho. Através do conhecimento obtido no estudo e das definições do modelo, foi feita uma análise individual das questões da base utilizável, sendo descartados os atributos considerados pouco relevantes para os objetivos do trabalho, baseado no Modelo Conceitual, exibido na Figura 2. Com este processo pretende-se aliviar uma etapa posterior de seleção de atributos baseado em algoritmos de *Filter* e *Wrapper*.

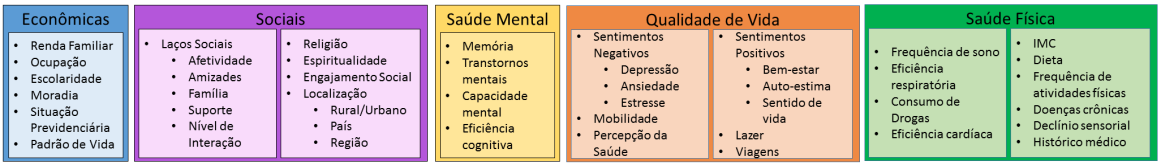


Fig. 2. Modelo Conceitual: Variáveis ambientais. Fonte: Adaptado de [Ribeiro and Zárate 2014]

3.4 Limpeza dos Dados

A base longitudinal gerada na seleção conceitual de atributos pode conter registros com valores que não caracterizam o domínio do problema sendo tratado, com excesso de dados ausentes (ou, no caso do ELSA, recusa a responder questões do estudo), ou mesmo erros de inserção e inconsistências, ou atributos que não possuem valores o suficiente para serem estudados. Esses registros e atributos poluem a base de dados e podem prejudicar os resultados do processo de KDD, sendo necessária uma limpeza prévia da base para detectar e tratar esses dados, retirando o máximo possível dessa poluição. Essa análise exige uma visão temporal, devendo considerar todas as versões da base, ou seja, os valores em cada unidade de tempo, e a forma de tratar eventuais inconsistências depende dos objetivos do estudo e características da base [Kantardzic 2011]. A etapa de limpeza dos dados é constituída de quatro análises, realizadas sequencialmente:

(1) Consistência:
Os atributos nominais possuem valores predeterminados, mas é possível que erros de inserção incluam na base valores diferentes dos previstos. Durante a análise de consistência, são realizadas as verificações cabíveis para garantir que os dados presentes na base estejam corretos. Se forem encontradas inconsistências, o registro ou o atributo podem ser eliminados, ou pode-se tentar recuperar o valor (correndo o risco de acrescentar imprecisões na base).

(2) Dados ausentes:
A análise de dados ausentes visa identificar dados com valores em falta, e lidar com estes como com as inconsistências (eliminando ou imputando um valor aproximado no lugar do dado). Diferentes técnicas para inferir o valor de um dado ausente podem ser aplicadas, como inserir a média dos valores, ou recuperar o valor através de atributos correlacionados.
No estudo de caso não houve número de dados ausentes relevantes em nenhum atributo para que ele fosse eliminado (o percentual de dados ausentes era inferior a 10% em todos os atributos, o

que foi considerável aceitável, para este estudo). Em muitos casos, as questões não se aplicavam a determinados respondentes, e a reposta para estas na base de dados é um código específico para estes casos, o que não é considerado dado ausente.

(3) *Análise de Outliers*:

Alguns registros possuem valores que não condizem com o conjunto do estudo, por serem muito discrepantes. Se ainda restarem *outliers* na base de dados após as duas primeiras análises, é recomendado, em estudos comuns, que esses registros sejam eliminados para que a caracterização dos dados não seja distorcida. Entretanto, em estudos longitudinais, a análise de *outliers* precisa levar em consideração a informação contida nesses registros, que pode ser relevante para predições de estados futuros da base.

Respondentes que tinham menos de 50 anos de idade na primeira onda foram eliminados sendo considerados *outliers* para este trabalho, por não caracterizarem o comportamento do público alvo. Esses registros existem na base porque indivíduos que moravam em domicílios com algum respondente eleito para o programa, com previsão de atingir os requisitos de idade durante o estudo, haviam sido incluídos como respondentes.

A Figura 3 exibe as possíveis evoluções de um caso de clusterização com *outliers* ao longo do tempo. A situação inicial pode se desenvolver das três formas mostradas, de acordo com a força de influência dos grupos e dos *outliers*, e com as características do estudo (por exemplo, se o estudo é de ordem social). A primeira hipótese é a de adaptação dos *outliers*, onde os registros modificam suas características ao longo do tempo para se encaixar nos padrões preestabelecidos de grupos existentes. Na segunda, observada principalmente em mudanças sociais, a influência dos *outliers* faz com que o comportamento do grupo se adapte gradualmente, modificando o comportamento característico daquele conjunto de registros. Na terceira, os *outliers* se unem para formar novos grupos, ou migram para outros grupos preexistentes, modificando o panorama do estudo.

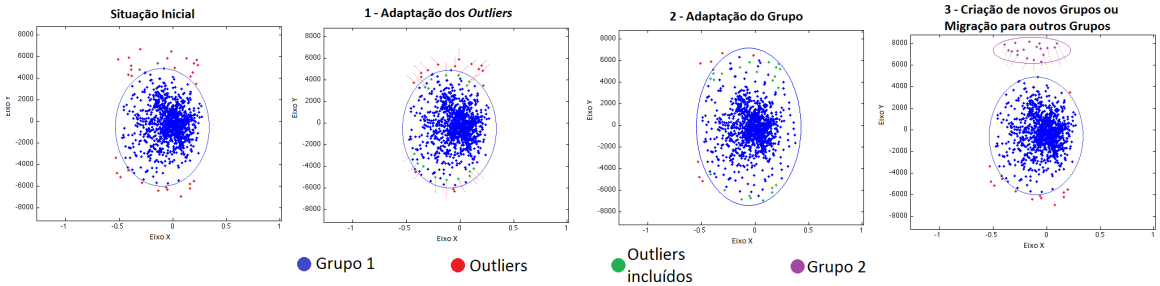


Fig. 3. Estudo temporal de *Outliers*. Fonte: Elaborado pelo autor.

(4) Quantidade de Informação:

Finalizando a limpeza, deve ser feita uma análise da quantidade de informação dos atributos da base, com uma análise de entropia nos atributos discretos e de variabilidade nos nominais. Se um atributo possuir uma variabilidade excessivamente baixa, incluí-lo no estudo afetará significativamente os resultados, tornando o conhecimento redundante e portanto diminuindo sua compreensão. Nesses casos, pode ser mais adequado reduzir a dimensionalidade da base retirando esses atributos com pouca informação. Na análise de quantidade de informação, foram eliminados atributos que tinham mais de 90% dos dados com um mesmo valor, o que possibilitou uma grande redução da base. Isto ocorre porque muitas questões podem não se aplicar à grande maioria dos respondentes.

3.5 Discretização de Atributos Numéricos

A quarta etapa da metodologia consiste em realizar as transformações necessárias nos dados para possibilitar o uso das ferramentas de mineração de dados. Algumas técnicas de DM exigem que os dados de entrada sejam nominais, criando a necessidade de se discretizar os dados numéricos presentes em grande parte das bases de dados. O problema da discretização não é trivial, por causa do número de combinações que podem ser realizadas em um intervalo contínuo de dados. Portanto, são utilizadas heurísticas, que se adequam a situações específicas e buscam uma aproximação da discretização ótima. A escolha da técnica mais indicada é feita de acordo com a distribuição dos valores, número de intervalos desejado, e a informação representada pelo atributo, considerando restrições de tempo e recursos [Garcia et al. 2013].

Como, no caso deste estudo, a equipe possui conhecimento sobre a informação que o atributo representa, foi possível utilizar um método empírico de discretização. O resultado dos métodos tradicionais, como discretização por frequência de valores ou intervalos equidistantes não foi capaz de representar adequadamente o conhecimento contido nos atributos, portanto foi feita a escolha de analisá-los caso a caso, e criar faixas e categorias condizentes com a informação representada. Conhecer o problema e os atributos a ponto de poder realizar uma discretização de forma empírica pode tornar a base mais robusta, pois as faixas foram criadas especificamente para a extração de conhecimento que se deseja fazer.

3.6 Junção de Atributos

Em bases nominais, a junção de questões altamente dependentes pode trazer uma redução significativa na dimensionalidade da base, sem grandes perdas de informação. As questões potencialmente relacionadas devem ser analisadas individualmente e, quando possível, julgar a forma de junção. A junção pode ser um simples produto cartesiano das opções de resposta das questões, ou uma recodificação deste, criando novas opções para as diferentes combinações de resposta possíveis, como ilustra a Figura 4. Ao analisar as questões para junção, deve ser considerado o grau de relação entre a informação que elas representam (apenas questões altamente relacionadas podem ser unidas), e o número de opções de respostas que a junção criada terá. Se esse número for muito alto, é recomendável discretizar a junção, ou deixar de realizá-la, porque atributos com excesso de opções de resposta dificultam a extração de conhecimento realizada pelos algoritmos de DM.

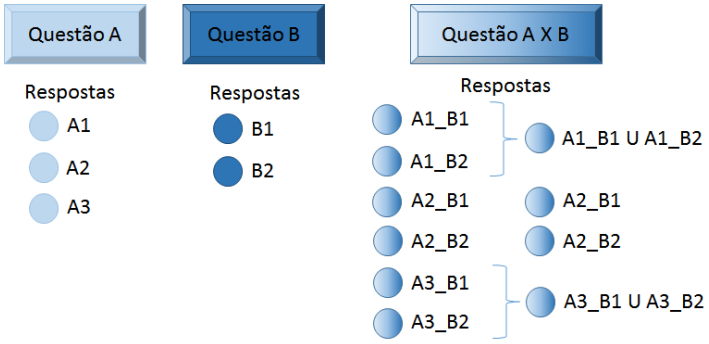


Fig. 4. Exemplo de junção de atributos. Fonte: Elaborado pelo autor.

No estudo de caso, dez questões do ELSA tinham um conjunto muito pequeno de opções de respostas, e estavam ligadas a outras questões sobre o mesmo tema (por exemplo, uma questão para determinar se o participante fuma e a seguinte para determinar com que frequência). A junção desses atributos com os relacionados o eles proporcionou uma última redução na dimensionalidade da base,

sem perda de informação, visto que foi possível realizá-la mantendo todas as possibilidades de conjuntos de resposta, sem tornar as questões excessivamente extensas. Ao final do procedimento, base de dados preparada ficou com 5352 registros, 255 atributos, em 5 ondas.

4. CONCLUSÕES

O processo de preparação dos dados está diretamente relacionado com o êxito do processo de KDD. Um amplo entendimento sobre o problema, exploração das possíveis fontes de dados, e modelagem sucinta da solução proposta afetam positivamente a capacidade dos envolvidos no projeto de tomar decisões acerca da relevância, confiabilidade e formato adequado dos dados e, consequentemente, o conhecimento gerado ao fim do processo.

Este trabalho apresentou um procedimento replicável de preparação de bases de dados voltada para bases longitudinais com atributos predominantemente nominais, realizando como estudo de caso a preparação da base ELSA, de um estudo sobre o envelhecimento humano. O estudo de caso partiu de uma base com características temporais e realizou uma série de filtragens e transformações nos dados para torná-la utilizável em um estudo longitudinal. Foram discutidas as diferenças inerentes a estudos longitudinais que devem ser observadas nas técnicas tradicionais de preparação de bases de dados, sendo estas as principais contribuições do trabalho.

REFERENCES

Antunes, C. M. and Oliveira, A. L. Temporal data mining: An overview. In *KDD workshop on temporal data mining*. pp. 1–13, 2001.

Banks, J. *Retirement, Health and Relationships of the Older Population in England: The 2004 English Longitudinal Study of Ageing (wave 2)*. Institute for Fiscal Studies (Great Britain), 2006.

Diggle, P., Heagerty, P., Liang, K.-Y., and Zeger, S. *Analysis of longitudinal data*. Oxford University Press, 2002.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. Advances in knowledge discovery and data mining. American Association for Artificial Intelligence, Menlo Park, CA, USA, From Data Mining to Knowledge Discovery: An Overview, pp. 1–34, 1996.

Garcia, S., Luengo, J., Sáez, J. A., López, V., and Herrera, F. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering* 25 (4): 734–750, 2013.

Kantardzic, M. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

Kotsiantis, S. B. and et al. Data preprocessing for supervised learning, 2006.

Last, M., Klein, Y., and Kandel, A. Knowledge discovery in time series databases. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 31 (1): 160–169, Feb, 2001.

Malloy-Diniz, L., Fuentes, D., and Cosenza, R. *Neuropsicologia do Envelhecimento: Uma Abordagem Multidimensional*. Vol. 1, 2013.

Marmot, M. English longitudinal study of ageing: Waves 0-5, 1998-2011. 20th edition, 2013.

Paes, B. C., Plastino, A., and Freitas, A. A. Seleção de atributos aplicada à classificação hierárquica. *Symposium on Knowledge Discovery, Mining and Learning - KDMiLe*, 2013.

Pyle, D. *Data preparation for data mining*. Vol. 1. Morgan Kaufmann, 1999.

Ribeiro, C. E. and Zárate, L. E. Uma revisão para identificar variáveis ambientais que influenciam o envelhecimento humano para estudos de mineração de dados. *Anais do XIV Congresso Brasileiro de Informática em Saúde*, 2014.

KDMiLe:05

Aprendendo a Ranquear com Boosting e Florestas Aleatórias: Um Modelo Híbrido

Clebson. C. A de Sá, Marcos. A. Gonçalves, Daniel. X. de Sousa, Thiago Salles

Universidade Federal de Minas Gerais, Brasil
{clebsonc, mgoncalv, danielxs, tsalles}@dcc.ufmg.br

Abstract. Aprendizado de Máquina tem sido aplicado com êxito em diversas tarefas de Recuperação de Informação, incluindo tarefas de ranqueamento conhecidas como *Learning to Rank (L2R)*. Nesse caso, o objetivo é recuperar os documentos mais relevantes para uma consulta, com base em funções aprendidas a partir de dados de treino associando pares (consulta, documento) a níveis pré-definidos de relevância. Neste artigo, apresentamos uma solução baseada em extensões do algoritmo BROOF, o atual estado-da-arte em classificação de texto. Na nossa abordagem adaptamos a ideia original desse algoritmo, que combina de forma única técnicas de *Boosting* e de Florestas Aleatórias (FAs), para considerar aspectos intrínsecos da tarefa de L2R tais como o contexto da consulta e a adaptação da noção de erro ao contexto de *rankings*. Nossos resultados experimentais demostram que é possível obter resultados de efetividade significativamente superiores ao estado-da-arte em L2R com uma redução substancial da necessidade de treinamento.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.2.6 [**Artificial Intelligence**]: Learning

Keywords: Boosting, Florestas Aleatórias, Aprendizado de Máquina, Ranqueamento, Recuperação de Informação

1. INTRODUÇÃO

Um dos principais desafios em buscas em grandes repositórios de informação como a Web (com trilhões de páginas) é a recuperação dos documentos "mais relevantes" no topo da lista de ranqueamento, de acordo com a informação fornecida pelo usuário para expressar sua necessidade de informação [Bartell et al. 1995], geralmente na forma de 2 ou 3 termos de pesquisa [Jansen et al. 2000]. Tal desafio é exacerbado por questões tais como a ambiguidade dos termos utilizados [Santos et al. 2015] ou pela falta de conhecimento ou experiência do usuário sobre o assunto pesquisado, o que forma uma barreira na construção de consultas com os "melhores termos" para um determinado mecanismo de busca.

Este problema se torna ainda mais desafiador nesta "nova era" de dispositivos móveis, visto que telas pequenas requerem um foco ainda maior em resultados mais relevantes no topo, dada a limitação de espaço para apresentação de resultados, sendo esta uma das características essenciais na determinação do sucesso de aplicativos. Outro fator de interesse é que nos dias de hoje, o conceito de relevância pode variar de acordo com muitos outros componentes: geolocalização, hora de acesso da busca, capacidade do dispositivo e muitas outras características possibilitadas pelos dispositivos móveis. Desta maneira, a principal pergunta a ser respondida com esses adventos é *"Como obter a melhor ordem, em um determinado contexto, entre todas as possíveis permutações de documentos recuperados contendo os documentos mais relevantes no topo e os não relevantes no inferior da lista de ranqueamento?"*

Para auxiliar na solução do problema acima mencionado, uma tendência recente é a utilização de algoritmos de *aprendizado de máquina* (AM) responsáveis por "combinar" um grande número de preditores contextuais que caracterizam pares (consulta, documentos) de treino em níveis de relevância

(e.g, relevante ou não-relevante). Essas combinações identificam padrões para geração das melhores ordenações das instâncias recuperadas para uma determinada consulta a partir de dados de treinamento. Essa área de pesquisa é conhecida em recuperação da informação como *"Learning to Rank (L2R)"*. Esses padrões aprendidos são utilizados para ordenar documentos para novas consultas de "teste" não observadas no treino.

Dentre as propostas para L2R encontradas na literatura, técnicas de combinação (*ensembles*) tais como RankBoost [Freund et al. 2003], AdaRank [Xu and Li 2007] e em especial Florestas Aleatórias (FAs) [Breiman 2001] têm sido consideradas como algumas das principais estratégias na predição correta de relevância das instâncias recuperadas. As duas primeiras técnicas de *ensemble* mencionadas acima utilizam uma estratégia conhecida como *Boosting* que melhora a acurácia final por meio da repesagem de amostras do conjunto de treino em regiões do espaço de entrada de difícil predição em iterações sequenciais [Schapire and Freund 2012]. A terceira abordagem funciona usando um comitê de modelos de predição conhecidos como árvores de decisão (DT) com utilização de *bagging* [Breiman 1996] e amostragem aleatória de preditores para aumentar a acurácia final [Mishina et al. 2014].

FAs e suas derivações possuem resultados considerados estado-da-arte em diversas tarefas de classificação e regressão [Fernández-Delgado et al. 2014]. Nesse ínterim, este artigo introduz uma nova abordagem para solução do problema de L2R com o desenvolvimento de um algoritmo original de FAs baseado na combinação de *bagging* com as propriedades fornecidas pelo procedimento de *Boosting*. Esta combinação, **única na literatura**, tem produzido os melhores resultados conhecidos em tarefas de classificação de texto [Salles et al. 2015]. Ela se baseia em duas ideias essenciais: (1) a estimativa dos erros do *boosting* é fornecida utilizando os documentos de treino não utilizados pelo *bagging* para o treinamento (o chamado *out-of-bag* ou *oob*) [Breiman 2001], invés de utilizar os dados de treinamento como o *boosting* original faz; (2) uma estratégia de *atualização de pesos ponderada* é introduzida na qual apenas documentos erroneamente classificados no *oob* são atualizados, em vez de todos os documentos do treino. Com (1) pretende-se obter estimativas mais confiáveis de erro já que existe uma tendência ao superajuste com o uso do treino para obter essas estimativas; e com (2) o objetivo é evitar alcançar mínimos locais muito rapidamente ao focar em poucas regiões do espaço de entrada que são difíceis de classificar.

Contudo, tarefas de L2R diferem consideravelmente daquelas de classificação em importantes aspectos, tais como: (i) foco na consulta (documentos com as mesmas características têm relevâncias diferentes para consultas distintas) e (ii) na interpretação do "erro" que envolve a posição relativa na ordenação do ranqueamento. Portanto, neste artigo estendemos as ideias de [Salles et al. 2015] para considerar as idiossincrasias específicas de tarefas de L2R. Em particular, utilizamos árvores de regressão no *ensemble* de FAs e ajustamos a estratégia ponderada dos estimadores *oob* gerados pelo *bagging* em FAs de acordo com os conceitos de L2R. Nesse caso, a atualização dos pesos das instâncias no conjunto de treino pelo mecanismo de *Boosting* são atualizados de acordo com o erro dado pela distância do valor predito para a relevância real dos documentos. Nossos resultados experimentais mostram ganhos estatisticamente significativos do nosso método (nomeado como BROOF-L2R) sobre o estado-da-arte em L2R na maioria das coleções *benchmark* utilizadas em nossas análises.

Esse artigo está organizado assim: trabalhos relacionados são descritos na Seção 2; a elaboração do problema como um de regressão é apresentada na Seção 3 conjuntamente com conceitos de *Boosting*, *Bagging* e Florestas Aleatórias; nosso método é detalhado na Seção 4; o projeto experimental é mostrado na Seção 5 com os resultados experimentais discutidos na Seção 6. A Seção 7 contém as conclusões e trabalhos futuros.

2. TRABALHOS RELACIONADOS

FAs são popularmente conhecidas devido a sua aplicação em diversas tarefas de regressão e classificação, como por exemplo, microarrays [Wu et al. 2012], segmentação de imagens [Yao et al. 2011],

reconhecimento de dígitos [Bernard et al. 2007] dentre outros. Outro fator importante é o fato de serem extremamente rápidas na construção do modelo de aprendizado de máquina e mais rápidas ainda na fase de predição [Biau et al. 2008]. Logo é de interesse geral dentre os pesquisadores tentar entender as características intrínsecas de FAs e melhorá-las, visto que suas derivações têm mostrado bons resultados em diversas áreas de pesquisa [Fernández-Delgado et al. 2014].

Uma das tentativas de melhorar a performance das FAs é proposta por [Geurts et al. 2006], denominada *Extremely Randomized Trees*, aplicada para L2R em [Geurts and Louppe 2011]. A ideia do algoritmo é remover a aleatoriedade das amostras dada pelo procedimento de *bagging* mantendo a aleatoriedade dos preditores. No entanto, diferentemente das FAs tradicionais, os preditores selecionados não são os que descrevem a melhor divisão dos dados, mas sim um limiar aleatório que é utilizado para definir a divisão entre os nós filhos de uma árvore de decisão. Desta maneira, o único parâmetro configurável durante a criação do modelo é esse limiar definido por $\sqrt{p}$ para classificação, ou p para regressão, onde p refere-se à dimensionalidade dos preditores. Estas alterações no método das FAs se mostraram competitivas com o método tradicional de FAs em termos de acurácia.

Em [Mishina et al. 2014], os autores apresentam uma versão de FAs melhorada com *Boosting*, com o objetivo de obter uma redução na quantidade de árvores de decisão no *ensemble* de FAs. Para isso eles ponderam todas as instâncias do conjunto de treino com pesos w_i, similares aos usados no *Boosting*, e em seguida modelam uma árvore de decisão sob esta ponderação, no qual as instâncias com maior peso são favorecidas durante a fase de divisão dos nós da árvore de decisão. As árvores geradas são avaliadas utilizando esse treino ponderado; caso haja uma predição incorreta, o peso desta amostra é aumentado de maneira que este seja melhor predito na próxima árvore a ser construída no *ensemble* de FAs. Caso a taxa de erro de classificação do conjunto de documentos avaliados em uma específica árvore de decisão ultrapasse o limiar $1 - \frac{1}{M}$ (sendo M a quantidade de classes) a árvore é descartada. Desta maneira, obtêm-se uma menor quantidade de árvores no *ensemble* com performance similar ao conjunto original, com a ajuda do mecanismo de *Boosting*.

Ao contrário do método anterior, a proposta em [Mohan et al. 2011] ataca o problema de L2R utilizando *Gradient Boosted Decision Trees – GBRT* e FAs como métodos independentes. Nesta abordagem os resíduos do algoritmo de FAs são utilizados para inicializar o modelo GBRT. Desta maneira, os resultados obtidos pelo modelo de FAs são refinados obtendo melhor predição quando comparado com o GBRT tradicional. De acordo com os autores, FAs são adequadas para inicializar o GBRT pelo fato de serem altamente resistentes ao superajuste do treino, sendo robustas a ruídos e resilientes ao ajuste de parâmetros.

Diferentemente dos trabalhos anteriores, em nosso método a criação do modelo é dada pela combinação *não-trivial* dos algoritmos de *Boosting* e FAs. No entanto o procedimento de *Boosting* é realizado em amostras do conjunto de treino erroneamente preditas que não foram utilizadas durante o treino de cada árvore de decisão do *ensemble* de FAs. Em suma, o nosso modelo obtém melhores resultados com a introdução de aleatoriedade dada por *bagging*, diminuindo o efeito de superajuste do treino ocorrido no algoritmo de *Boosting* em regiões de difícil predição do conjunto de treino. Os conceitos necessários para o entendimento do método proposto são introduzidos a seguir.

3. FUNDAMENTAÇÃO TEÓRICA

L2R pode ser considerado uma técnica de aprendizado de máquina supervisionada com a sútil introdução de consultas durante o estágio de criação do modelo de predição. Desta maneira, os dados de treino são constituídos por um conjunto de consultas q_i $(i = 1, \ldots, n)$, cada qual com um conjunto de documentos associados x_j^i (j refere-se ao j^{th} documento e i à i^{th} consulta). As características de cada documento são representadas por um vetor de preditores e o grau de relevância do documento. A relevância do documento indica a importância desse para a consulta [Liu 2007; 2011]. O objetivo principal é criar um modelo capaz de predizer o grau de relevância de um documento de teste não

presente no conjunto de treino. A construção do modelo de predição do método proposto por nós faz uso da combinação de dois mecanismos de aprendizado de máquina: *Boosting* e FAs.

Boosting é um arcabouço iterativo que proporciona que vários modelos foquem na predição de regiões distintas do espaço de entrada com o objetivo de obter uma melhor predição. Para atingir este objetivo, cada amostra do conjunto de treino possui um peso w_i que indica a importância desta amostra na construção do modelo de predição. Em cada iteração, as amostras são avaliadas se foram preditas corretamente ou erroneamente por um algoritmo base. Caso as predições sejam consideradas errôneas é feito uma repesagem dos pesos w_i com o intuito de induzir um novo modelo especialista com estas amostras de difícil predição. Ao final das t iterações, é obtido um comitê de modelos contendo t especialistas em diferentes amostras do conjunto original de treino [Schapire and Freund 2012]. O único requerimento para o funcionamento deste mecanismo é que o algoritmo base utilizado nas iterações tenham acurácia de predição melhor que $1/2$ *(random guessing)* e que a utilização dos pesos das amostras do treino sejam utilizadas no *resample* ou que o algoritmo base seja capaz de lidar com os pesos [Bauer and Kohavi 1999]. No nosso método, o algoritmo base utilizado são as FAs com árvores de decisão CART (Classification and Regression Trees) que utilizam os pesos para fazer a divisão dos dados nos nós das árvores de decisão.

O segundo mecanismo necessário no nosso método são as FAs, sendo estas consideradas o algoritmo base do procedimento interno do arcabouço de *Boosting*. FAs são algoritmos de *ensemble* criadas pela combinação de vários classificadores conhecidos como árvores de decisão. Árvores de decisão são extremamente rápidas na construção do modelo e na predição de amostras não vistas. No entanto, é sabido que árvores únicas quando criadas em seu tamanho máximo possuem alta variância, sendo responsáveis pelo superajuste do treino, obtendo baixo poder de generalização em amostras ainda não vistas durante o teste. Para contornar este problema, é introduzido um fator randômico no conjunto de treino com a utilização de ***b**ootstrap **agg**regati**o**n* sendo comumente chamado pelo acrônimo *bagging*. Em *bagging*, dado um conjunto de treino D, são construídos n subconjuntos das amostras aleatoriamente com repetição, desta maneira, estima-se seguindo a distribuição *Poison* que há $\approx 63\%$ das amostras em um subconjunto de D [Breiman 1996]. Cada subconjunto de D é utilizado para criar uma árvore de decisão independente, no qual os $\approx 37\%$ das amostras não utilizadas no processo de treino são conhecidos como *out-of-bag estimators* ou pela sigla *oob*. Estas estimativas *oob* possuem a vantagem de serem calculadas durante o processo de criação da árvore no modelo de *ensemble*, o que nos proporciona a vantagem de fazer a validação do modelo criado em momento de construção sem a necessidade de métodos de validação [Breiman 2001]. Outra característica acrescentada na abordagem de FAs são a criação de árvores de decisão com uma parcela aleatória dos preditores. Embora não seja regra, na proposta original [Breiman 2001] é utilizado $\sqrt{p}$ (p é a dimensionalidade dos preditores). Em nossos experimentos a proporção de 0.3 de p se mostrou suficiente.

Explanado os dois modelos utilizados na elaboração do nosso método, iremos demonstrar como a combinação aditiva desses dois mecanismos é utilizada para obter melhores resultados de ranqueamento, fazendo uso das estimativas oob em nosso método híbrido para L2R.

4. BROOF-L2R

Embasados no fato de que técnicas de *ensemble* como *Boosting* e FAs possuem os melhores resultados em diversas aplicações, combinamos as características dos dois arcabouços de maneira original seguindo os seguintes passos: (1) Primeiramente estimamos os erros para o algoritmo de *Boosting* utilizando as estimativas *oob* produzidas pelo procedimento de *bagging* ao invés de utilizar o erro no conjunto de treinamento como ocorrido na abordagem original de FAs; e (2) exploramos uma estratégia ponderada de atualização dos pesos w_i para as instâncias no conjunto de treino, no qual são atualizadas as estimativas *oob* em consideração a distância do grau de relevância. Com o passo (1) obtemos melhores estimativas de erro que são mais confiáveis do que aquelas mensuradas no conjunto de treino que é tendencioso ao superajuste. E, com o passo (2), evita-se que sejam feitas predições com enfoque

Algoritmo 1 BROOF-L2R

procedure FIT($Q_{train} = q_i(\{x_j^i\}^m, y_j^i)$, $iter = (1, 2, \ldots, t)$, $trees = n\_trees$)
  $w_j^i = 1 \big/ \sum_{i=1}^{n} m$;
  $L = \emptyset$;
  for t in $iter$ **do**
    $FA = FA_{Regressor}.fit(Q_{train})$;
    $D = max(|FA.oob\_pred_j - y_j|)$;
    $e_j^i = \left|FA.oob\_pred_j^i - y_j^i\right| \big/ D$;
    $\epsilon = \sum_{i=1}^{n}\sum_{j=1}^{m}(e_j^i * w_j^i)$;
    if $\epsilon >= 0.5$ **then**
      $iter = \emptyset$;
      $break$;
    end if
    $\beta = \epsilon/(1-\epsilon)$;
    $w_j^i = w_j^i * \beta^{1-e_j^i}$;
    $Z = \sum_{i=1}^{n}\sum_{j=1}^{m}(w_j^i)$;
    $w_j^i = w_j^i \big/ Z$;
    $L.add(FA, \beta)$;
  end for
  Retorne a lista L contendo os modelos criados com os devidos β de cada iteração
end procedure

procedure PREDICT($Q_{test} = q_i(\{x_j^i\}^m)$, L = (lista de FAs com as representações de importância de cada modelo no *ensemble* de *Boosting* data por β))
  $x_j = \dfrac{\sum_{\{t\}}(log\frac{1}{\beta^{\{t\}}}) * FAs^{\{t\}}.predict(x_j)}{\sum_{\{t\}} log\frac{1}{\beta^{\{t\}}}}$
  Retorne A predição dada pela combinação ponderada das iterações de *Boosting* para cada x_j.
end procedure

muito rápido em apenas algumas regiões do espaço de entrada. O pseudocódigo do nosso método é apresentado acima.

Conforme pode ser observado no Algoritmo 1, o processo de criação do modelo é dado por um procedimento com 3 parâmetros; 1) o conjunto de consultas q_i com seus respectivos documentos e julgamentos de relevância como base de treinamento do modelo; 2) a quantidade de iterações do mecanismo de *Boosting* e 3) a quantidade de árvores em cada FA criada por iteração. No início do algoritmo é criada uma distribuição de pesos dada por $w_j = 1 \big/ \sum_1^n m$ (m refere-se a quantidade de documentos x_j^i – no qual j identifica determinado documento de uma consulta i). Em sequência é construído em cada iteração um modelo de FA utilizando regressão. Após a criação do modelo, faz-se uso das estimativas preditivas dadas pelo *oob* para computar o erro absoluto de acordo com o grau de relevância do *oob*. Após obter o maior erro absoluto D os resíduos das instâncias analisadas e_j^i são computados com uma função linear, normalizados entre o intervalo [0, 1].

O erro do modelo ϵ é dado pela combinação aditiva dos resíduos com os pesos das instâncias do *oob*, sendo este utilizado como critério de convergência das iterações caso o erro ultrapasse $1/2$. Caso não haja convergência, é estimada a importância do modelo na região do espaço de entrada atual dado pelo fator β. Após o cálculo do fator de importância do modelo os pesos w_i são atualizados de acordo com o delineamento β do modelo na iteração t. Para que a atualização dos pesos se mantenha de acordo com uma determinada distribuição, fazemos divisão dos pesos pela constante Z. Ao final do procedimento de *fitting* do *ensemble*, obtêm-se uma lista L contendo os modelos de FAs e os graus de importância de cada modelo β na predição final do *ensemble*. A predição do conjunto de teste é dada pela combinação ponderada dos modelos obtidos de acordo com a importância do modelo β conforme a fórmula mostrada no procedimento PREDICT ao final do Algoritmo 1.

5. PROJETO EXPERIMENTAL

Os testes feitos com o nosso método foram executados em 8 coleções de dados conhecidas amplamente na comunidade científica na área de recuperação da informação. As coleções utilizadas estão disponibilizadas gratuitamente online, sendo elas: 1) WEB10K[1] contendo consultas da ferramenta de busca Bing disponibilizada pela Microsoft; 2) A base de dados WEBSCOPE^{TM} Yahoo! Learning to Rank

[1]http://research.microsoft.com/en-us/projects/mslr/

Challenge[2] versão 1 conjunto 2 disponível pela Yahoo; e 3) a coleção .Gov disponível na LETOR[3] contendo 6 bases de dados nomeados HP2003, HP2004, TD2003, TD2004, NP2003 e NP2004 coletados por um *crawler* com domínios .Gov. Com exceção da base de dados disponibilizada pela Yahoo, todas as outras estão divididas em 5 *folds* contendo partições do conjunto original dos dados para validação cruzada divididos em partições de treino, validação e teste em cada *fold*. Portanto, todos os experimentos foram executados utilizando um procedimento de validação cruzada com 5 *folds*.

Para efeito de comparação experimental do nosso método com as FAs tradicionais delimitamos o espaço de entrada em ambos os algoritmos de maneira que a primeira iteração do processo de *Boosting* no nosso algoritmo fosse construída com o mesmo *bagging* de dados da FA tradicional. Este passo é de extrema importância, visto que a perturbação dos dados de treino com a mínima discrepância do espaço de entrada pode gerar resultados distintos, como consequência favorecendo os resultados de um ou outro algoritmo. Para diminuir os efeitos dessa aleatoriedade, repetimos o procedimento experimental de validação cruzada com 5 *folds 30 vezes*, alterando-se o *bagging* inicial dos algoritmos com o intuito de obter uma amostra satisfatória dos resultados para comparação dos testes estatísticos analisados. Dessa forma, todos os resultados reportados correspondem à média dos 5 *folds* de teste em 30 repetições, totalizando 150 resultados.

Os resultados mostrados dos *baselines* AdaRank e RankBoost com a devida parametrização dos algoritmos são os mostrados na LETOR, no qual a quantidade de iterações do mecanismo de *Boosting* de 500 iterações foi mantida para o nosso método em todos os experimentos. No entanto, nos experimentos conduzidos em nosso método essa quantidade de iterações nunca foi atingida devido ao fator de convergência do erro da iteração não ultrapassar $\frac{1}{2}$. No caso do SVM^{rank} configuramos a margem de delineamento, dada pelo parâmetro C, com os valores 0.0001, 0.001, 0.01, 0.1, 1.0, 10.0, 100.0 e 1000.0 para as bases de dados WEB10K e Yahoo, no qual os resultados são mostrados em função do melhor parâmetro em cada *fold*. Nas bases da coleção LETOR usamos o parâmetro C ajustado conforme mostrado na página de *baselines*[4] disponíveis pela LETOR.

Após obter a permutação de ranqueamento dos métodos analisados, utilizamos as métricas de avaliação MAP e NDCG@10 como critério de avaliação. Os resultados obtidos são analisados em função da média por consulta da quantidade de experimentos executados. Com a média por consulta, efetuamos o cálculo do teste pareado de ambas as métricas considerando 0.95% do intervalo de confiança para computação do teste pareado Wilcoxon. Escolhemos utilizar o teste pareado Wilcoxon pelo fato deste ser não paramétrico, não sendo necessário assumir uma distribuição normal ou homogeneidade sobre o conjunto de dados analisados, no qual é considerado mais confiável em relação a testes estatísticos paramétricos [Demšar 2006].

6. RESULTADOS

Os resultados são mostrados na Tabela I para ambas as métricas de avaliação utilizadas. A Figura 1 mostra os resultados obtidos por meio de análise quantitativa a convergência da quantidade de árvores para o algoritmo de FAs tradicional e o nosso método BROOF-L2R na coleção TD2003. Nesta análise inicial observamos que nosso método sempre obtém os melhores resultados quando comparado com o algoritmo de FAs original. A partir dessa análise, conduzimos os demais experimentos considerando 300 árvores, que foi o ponto de convergência obtido no treino na maioria das coleções (resultado não apresentado por limitações de espaço).

Levando em consideração a métrica MAP, obtemos melhores resultados em 6 das 8 coleções avaliadas quando comparamos com o *baseline* AdaRank. Quando comparado com o algoritmo RankBoost temos ganhos em todas as coleções. Nos resultados obtidos com a comparação das FAs tradicionais podemos

[2]http://webscope.sandbox.yahoo.com/catalog.php?datatype=c
[3]http://research.microsoft.com/en-us/um/beijing/projects/letor/letor3dataset.aspx
[4]http://research.microsoft.com/en-us/um/beijing/projects/letor/letor3baseline.aspx

	Mean Average Precision – MAP					Normalized Discounted Cumulative Gain – NDCG@10				
Coleção	BROOF-L2R	FAs	AdaRank	RankBoost	RankSVM	BROOF-L2R	FAs	AdaRank	RankBoost	RankSVM
TD2003	**0.28804**	0.27864	0.22830	0.22740	0.26280	0.36080	**0.36346**	0.3069	0.31220	0.34610
TD2004	**0.26329***	0.25220	0.21890	0.26140	0.22370	**0.35815**	0.35058	0.3285	0.35040	0.30780
NP2003	**0.70905***	0.70344	0.67830	0.70740	0.69570	**0.81102***	0.79685	0.7641	0.80680	0.80030
NP2004	0.61560	0.59698	0.62200	0.56400	**0.65880**	0.74556	0.72465	0.7497	0.69140	**0.80620**
HP2003	**0.77581***	0.77139	0.77100	0.73300	0.74080	**0.83974***	0.83192	0.8384	0.81710	0.80770
HP2004	0.63874*	0.61054	**0.72190**	0.62510	0.66750	0.72400*	0.69873	**0.8328**	0.74280	0.72960
YAHOO-V1.S2	**0.56549***	0.56336	0.53249	0.54681	0.52220	**0.70633***	0.70314	0.65536	0.67730	0.64090
WEB10K	**0.34244***	0.33770	0.28209	0.31620	0.32456	**0.43496***	0.42450	0.34259	0.39707	0.39992

Table I: MAP à direita e NDCG@10 à esquerda. Melhores resul. em negrito. * são estatisticamente superiores às FAs.

vislumbrar que o nosso método obtém os melhores resultados (estatisticamente significativos) em 5 coleções e, similarmente ao caso do RankBoost, não perdemos em nenhuma coleção. Por fim, o nosso método obtém os melhores resultados na maioria das coleções quando comparado com o SVM^{rank}, possuindo ganhos favoráveis ao nosso método em 6 das 8 coleções avaliadas.

Resultados com NDCG@10 são bem similares àqueles obtidos com MAP: ganhamos nas mesmas coleções para os algoritmos AdaRank e RankBoost; comparado às FAs nosso método obtém os melhores resultados em 5 coleções com empate estatístico nas demais; e na comparação do SVM^{rank} constatamos também ganhos em 6 coleções, com algumas variações em coleções individuais. Em suma, podemos afirmar que o nosso método é um dos melhores preditores de ranqueamento em quase todas as coleções avaliadas, visto que, em comparação com a maioria dos *baselines* avaliados temos ganhos com significância estatística ou empate estatístico.

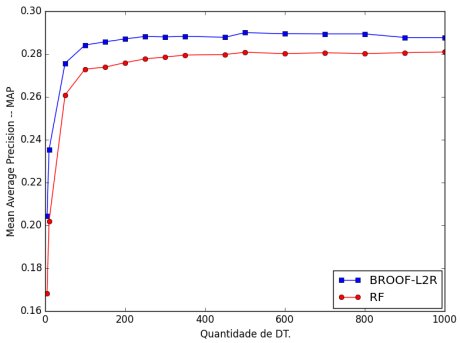


Fig. 1: Convergência no número de árvores (TD2003)

Dos resultados obtidos pelo nosso método, os mais surpreendentes estão relacionados à capacidade de generalização em relação ao tamanho da base de dados utilizadas como conjunto de treino. Nesses experimentos criamos subconjuntos do treino original com variações de intervalo de 10%. Após criar os subconjuntos, treinamos os algoritmos e os testamos nos *folds* de testes respectivos. Nesse cenário, o nosso método obteve os melhores resultados com uma quantidade substancialmente menor de treino quando comparado com as FAs tradicionais utilizando o conjunto de treino completo, em todas as bases avaliadas. Devido a restrições de limite de espaço mostramos apenas os 3 melhores resultados na Figura 2, no qual observa-se que com apenas 10% do conjunto de treino ultrapassamos o melhor resultado obtido com os 100% de treino das FAs tradicionais para a coleção WEB10K; em sequência temos que com 30% e 40% superamos os resultados das FAs para as coleções WEBSCOPE^{TM} Yahoo! e TD2004 respectivamente nesta ordem. Os ganhos obtidos com esse experimento para as demais coleções considerando parcelas do treino comparado com o conjunto completo de treino com as FAs tradicionais são de 50% para NP2004; 60% para a HP2004 e NP2003; 70% para a TD2003 e HP2003.

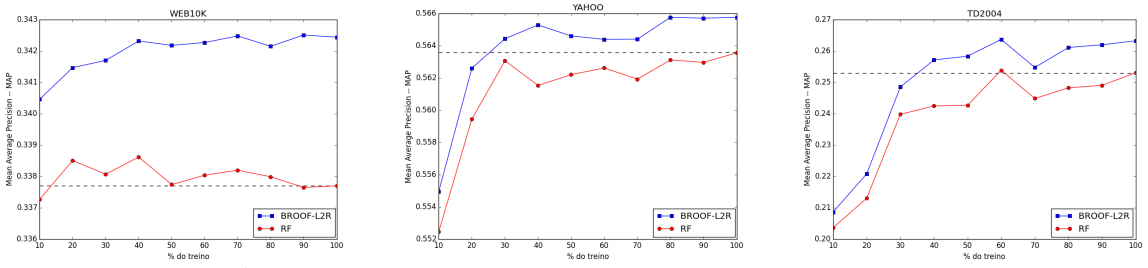


Fig. 2: MAP utilizando diferentes porcentagens do treino: WEB10K, Yahoo e TD2004.

7. CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho combinamos de forma original dois importantes algoritmos de *ensemble* com o intuito de melhorar a performance das FAs em tarefas de L2R. Nosso método explora o *Boosting* fazendo a perturbação de regiões do espaço de entrada que são consideradas de difícil predição por meio da escolha adaptativa destas regiões com estimativas *oob* geradas pelo procedimento de *bagging* das FAs. Esta aleatoriedade das amostras nas FAs em combinação com o *Boosting* em instâncias não utilizadas no treino (estimativas *oob*) são capazes de reduzir a variância e o superajuste e, como consequência produzir melhores modelos de ranqueamento. Mostramos que utilizar o cálculo do erro como a diferença entre o valor absoluto da função de regressão e a relevância real dos documentos é uma estratégia eficaz. Em nossos experimentos, constatamos que o nosso método possui os melhores resultados dentre todos os *baselines* avaliados sendo capaz de generalizar as predições com proporções bem menores do conjunto de treino, o que mostra o potencial de utilização do nosso método em aplicações reais de L2R. Futuramente pretendemos: (i) avaliar o comportamento do modelo sob os aspectos de variância e de superajuste do modelo que acreditamos explicar nosso bons resultados; (ii) considerar a ponderação das consultas; e (iii) melhorar o esquema de pesos para que preditores fortes tenham maior probabilidade de escolha na hora de divisão do conjunto de dados nas FAs.

REFERENCES

Bartell, B., Britannica, E., Belew, R., Cottrell, G., and Belew, R. Learning to retrieve information. In *Proceedings of the Swedish Conference on Connectionism*, 1995.

Bauer, E. and Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach. Learn.* 36 (1-2): 105–139, July, 1999.

Bernard, S., Adam, S., and Heutte, L. Using random forests for handwritten digit recognition. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*. Vol. 2. pp. 1043–1047, 2007.

Biau, G., Devroye, L., and Lugosi, G. Consistency of random forests and other averaging classifiers. *J. Mach. Learn. Res.* vol. 9, pp. 2015–2033, June, 2008.

Breiman, L. Bagging predictors. *Machine Learning* 24 (2): 123–140, 1996.

Breiman, L. Random forests. *Machine Learning* 45 (1): 5–32, 2001.

Demšar, J. Statistical comparisons of classifiers over multiple data sets. *JMLR* vol. 7, pp. 1–30, Dec., 2006.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* 15 (1): 3133–3181, Jan., 2014.

Freund, Y., Iyer, R., Schapire, R. E., and Singer, Y. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.* vol. 4, pp. 933–969, 2003.

Geurts, P., Ernst, D., and Wehenkel, L. Extremely randomized trees. *Machine Learning* 63 (1): 3–42, 2006.

Geurts, P. and Louppe, G. Learning to rank with extremely randomized trees. In *Proc. of the Yahoo! L2R Challenge, held at ICML 2010, Haifa, Israel, June 25, 2010*. pp. 49–61, 2011.

Jansen, B. J., Spink, A., and Saracevic, T. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management* vol. 36, pp. 207–227, 2000.

Liu, T.-Y. Learning to Rank for Information Retrieval. *FTIR* 3 (3): 225–331, 2007.

Liu, T.-Y. *Learning to Rank for Information Retrieval.* Springer, 2011.

Mishina, Y., Tsuchiya, M., and Fujiyoshi, H. Boosted random forest. In *VISAPP 2014*. pp. 594–598, 2014.

Mohan, A., Chen, Z., and Weinberger, K. Q. Web-search ranking with initialized gradient boosted regression trees. In *Proc. of the Yahoo! L2R Challenge, held at ICML 2010, Haifa, Israel, June 25, 2010*. pp. 77–89, 2011.

Salles, T., Gonçalves, M., Rodrigues, V., and Rocha, L. Broof: Exploiting out-of-bag errors, boosting and random forests for effective automated classification. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '15. ACM, New York, NY, USA, pp. 353–362, 2015.

Santos, R. L. T., Macdonald, C., and Ounis, I. Search result diversification. *FTIR* 9 (1): 1–90, 2015.

Schapire, R. E. and Freund, Y. *Boosting: Foundations and Algorithms.* The MIT Press, 2012.

Wu, X., Zang, W., Cui, S., and Wang, M. Bioinformatics analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *Eur. Rev. for Med. and Pharm. Sci.* 16 (11): 1582–1587, 2012.

Xu, J. and Li, H. Adarank: A boosting algorithm for information retrieval. In *SIGIR'07*. pp. 391–398, 2007.

Yao, B., Khosla, A., and Li, F. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*. pp. 1577–1584, 2011.

KDMiLe:06

Padrões de Alta Utilidade em Relações n-árias *Fuzzy*

Loïc Cerf

Universidade Federal de Minas Gerais, Brazil
lcerf@dcc.ufmg.br

Abstract. Dada uma relação binária na qual cada tupla é associada a um número positivo chamado utilidade, um *itemset de alta utilidade* envolve tuplas cujas utilidades se somam a um valor suficientemente alto. Este artigo trata da mineração desse tipo de padrão em relações n-árias *fuzzy*, isto é, um contexto mais geral. "Ter uma utilidade acima de um limiar" é visto como uma restrição. Ela permite uma poda da busca dos padrões realizada pelo algoritmo `multidupehack`. O desempenho obtido no contexto clássico da relação binária permanece competitivo. Padrões de alta utilidade descobertos em uma verdadeira relação ternária *fuzzy* mostram, pela relevância deles, que a nossa generalização do contexto de aplicação é valiosa.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords: itemsets de alta utilidade,mineração de dados,poda,relações n-árias *fuzzy*,restrição

1. INTRODUÇÃO

Os *itemsets* frequentes são padrões bastante populares. Por exemplo, observando no caixa de um supermercado os produtos comprados pelos clientes, um conjunto de produtos todos comprados por muitos clientes (o *suporte*) é um *itemset* frequente. Esse padrão indica uma correlação entre os produtos e pode ajudar na elaboração de estratégias de posicionamento dos produtos, de criação de pacotes, etc. A Tabela Ia representa uma possível amostra dos dados de venda de um supermercado. A mineração imaginada acima somente considera as colunas `cliente` e `produto`. Ela trata a relação binária composta pelos pares (`cliente`,`produto`). Considerar as outras informações pode ser valioso. Os *itemsets de alta utilidade* podem levar em conta a informação de preço. Esses padrões são os *itemsets* que "cobrem" pares (`cliente`,`produto`) tal que o dinheiro total gasto nas compras correspondentes ultrapassa um limiar escolhido pelo analista que, desta maneira, pode ignorar os *itemsets* com pouco peso no faturamento.

"Ter um peso significativo no faturamento" pode ser visto como uma restrição. Uma restrição filtra os melhores *itemsets*, assim facilitando o trabalho do analista. Além disso, existem restrições que podem ser usadas ao longo da busca dos padrões para podar o espaço de busca, ou seja, o espaço dos *itemsets*. O tempo de mineração é assim (drasticamente) reduzido e maiores conjuntos de dados se tornam tratáveis.

Este artigo considera a mineração dos *itemsets* de alta utilidade como uma mineração sob restrição. O trabalho consiste em estudar a restrição e implementá-la no algoritmo `multidupehack` [Cerf and Meira Jr. 2014]. Esta abordagem se diferencia das soluções existentes na literatura, que são específicas à mineração dos *itemsets* de alta utilidade ([Yao et al. 2004; Liu and Qu 2012; Fournier-Viger et al. 2014] entre muitas outras referências). A maior vantagem do nosso trabalho é a sua generalidade. `multidupehack` minera relações n-árias *fuzzy* e relações binárias são somente um caso particular. Em nosso exemplo, essa generalidade permite levar em consideração os dias das compras e as quantidades

Este trabalho foi financiado pelo CNPq através do edital Universal – MCTI/CNPq №14/2013.

dia	cliente	produto	quantidade	preço
1 maio	alice	iogurte	6	R$ 7
1 maio	alice	ovo	6	R$ 2
1 maio	bob	ovo	12	R$ 4
1 maio	bob	vinho	1	R$ 20
1 maio	bob	água	1	R$ 1
2 maio	alice	vinho	2	R$ 40
2 maio	dave	iogurte	6	R$ 5
⋮	⋮	⋮	⋮	⋮

(a) Dados de venda de um supermercado.

dia	cliente	produto	↦	grau de pertinência
1 maio	alice	iogurte	↦	0,6
1 maio	alice	ovo	↦	0,5
1 maio	bob	ovo	↦	0,9
1 maio	bob	vinho	↦	0,8
1 maio	bob	água	↦	0,3
2 maio	alice	vinho	↦	1
2 maio	dave	iogurte	↦	0,6
⋮	⋮	⋮	⋮	⋮

(b) Relação ternária *fuzzy*.

dia	cliente	produto	↦	utilidade
1 maio	alice	iogurte	↦	R$ 7
1 maio	alice	ovo	↦	R$ 2
1 maio	bob	ovo	↦	R$ 4
1 maio	bob	vinho	↦	R$ 20
1 maio	bob	água	↦	R$ 1
2 maio	alice	vinho	↦	R$ 40
2 maio	dave	iogurte	↦	R$ 5
⋮	⋮	⋮	⋮	⋮

(c) Função de utilidade ($I = \{1, 2, 3\}$).

Table I: Dados brutos, relação ternária *fuzzy* e função de utilidade correspondentes.

compradas. Os dias formam uma terceira dimensão da relação e um padrão é um conjunto de produtos todos comprados por um conjunto de clientes durante um conjunto de dias. A descoberta de padrões sazonais se torna possível. O grau de pertinência de uma tupla à relação pode ser derivado da quantidade comprada. Por exemplo, na Tabela Ib, o analista considerou que comprar seis ovos é moderadamente significativo (o grau de pertinência da tupla correspondente é 0,5), menos do que comprar uma dúzia (o grau de pertinência é 0,9). `multidupehack` oferta outras novas possibilidades como a imposição de uma noção de maximalidade dos padrões e o uso de restrições adicionais.

A Seção 2 lista as definições necessárias à formalização do problema. Na Seção 3, esse problema é relacionado àqueles tratados na literatura. A Seção 4 é dedicada ao estudo da restrição de utilidade mínima e ao seu uso no `multidupehack` para podar o espaço de busca. A Seção 5 mostra que o nosso trabalho é competitivo com o estado da arte no caso da mineração de relações binárias e que o seu aspecto geral permite a descoberta de padrões relevantes em uma grande relação ternária *fuzzy*. Finalmente, a Seção 6 conclui o artigo.

2. DEFINIÇÕES

2.1 Relação n-ária *fuzzy*

Neste artigo, $\times$ e $\prod$ denotam o produto Cartesiano. Dados $n \in \mathbb{N}$ conjuntos finitos $(D_1, \ldots, D_n)$, chamados de *dimensões*, uma *relação n-ária fuzzy* $\mathcal{R} \in [0,1]^{\prod_{i=1}^{n} D_i}$ associa a cada tupla $t \in \prod_{i=1}^{n} D_i$ um valor $\mathcal{R}_t$ entre 0 e 1, o *grau de pertinência* da tupla t à relação $\mathcal{R}$ (ver a Tabela Ib para um exemplo). O conceito de relação n-ária *fuzzy* generaliza o conceito de relação n-ária *crisp* na qual cada tupla ou pertence a relação (o grau de pertinência é 1) ou não (o grau de pertinência é 0).

2.2 Padrões

Dadas as n dimensões $(D_1, \ldots, D_n)$ da relação, chamamos de *padrão* n subconjuntos de cada uma das n dimensões. Matematicamente, $(X_1, \ldots, X_n)$ é um padrão se e somente se $\forall i \in \{1, \ldots, n\}$, $X_i \subseteq D_i$. Dados n limiares de tolerância a ruído $(\epsilon_1, \ldots, \epsilon_n) \in \mathbb{R}_+^n$, um padrão $(X_1, \ldots, X_n)$ é

um *ET-n-conjunto*[1] se e somente se $\forall i \in \{1,\ldots,n\}$, $\forall x \in X_i$, $\sum_{t \in X_1 \times \cdots \times \{x\} \times \cdots \times X_n} 1 - \mathcal{R}_t \leq \epsilon_i$. Nessa definição, $1 - \mathcal{R}_t$ é a quantidade de ruído a ser tolerado para aceitar a tupla t no ET-n-conjunto. Como tais quantidades são simplesmente somadas, a tolerância a ruído é dita "absoluta". Como cada somatório é indexado pelas tuplas no padrão que envolvem um elemento x particular, a tolerância é também dita "por elemento". [Cerf and Meira Jr. 2014] justifica as duas escolhas. Com $(\epsilon_1,\ldots,\epsilon_n) = (0,\ldots,0)$ (nenhuma tolerância a ruído), os ET-n-conjuntos em uma relação *crisp* são os padrões que envolvem somente tuplas que pertencem à relação.

Um padrão é *fechado na i-ésima dimensão* quando adicionar elementos ao seu i-ésimo conjunto sempre leva a padrões que não são ET-n-conjuntos. Matematicamente, dado $i \in \{1,\ldots,n\}$, um padrão $(X_1,\ldots,X_i,\ldots,X_n)$ é fechado na i-ésima dimensão se e somente se $\forall X_i' \supset X_i$, $(X_1,\ldots,X_i',\ldots,X_n)$ não é um ET-n-conjunto. Um ET-n-conjunto fechado é um ET-n-conjunto que é fechado nas n dimensões. No caso de uma relação *crisp* e nenhuma tolerância a ruído, os ET-n-conjuntos fechados são padrões que envolvem somente tuplas que pertencem à relação e que não podem ser estendidos sem introduzir no padrão uma tupla ausente da relação. Quando $n = 2$, são os *itemsets* fechados com os suportes deles. Quando os ET-n-conjuntos são somente fechados na dimensão do suporte, temos os *itemsets* (ditos "frequentes" se a cardinalidade do suporte é suficientemente grande).

2.3 Utilidade

Dado um subconjunto $I \subseteq \{1,\ldots,n\}$, uma *função de utilidade* $u \in \mathbb{R}_+^{\prod_{i\in I} D_i}$ associa um valor positivo a cada tupla no espaço $\prod_{i\in I} D_i$, ou seja, a cada tupla reduzida às componentes com índices em I (ver a Tabela Ic para um exemplo). Esse valor é chamado *utilidade* da tupla. A utilidade de um padrão $(X_1,\ldots,X_n)$ é simplesmente a soma das utilidades das projeções únicas das tuplas contidas nele:

$$\sum_{t \in \prod_{i\in I} X_i} u(t) \ .$$

2.4 Definição do problema

Podemos agora definir o problema principal de que este artigo trata. Dados uma relação n-ária *fuzzy* $\mathcal{R} \in [0,1]^{\prod_{i=1}^n D_i}$, limiares de tolerância a ruído $(\epsilon_1,\ldots,\epsilon_n) \in \mathbb{R}_+^n$, um subconjunto $I \subseteq \{1,\ldots,n\}$, uma função de utilidade $u \in \mathbb{R}_+^{\prod_{i\in I} D_i}$ e um limiar de utilidade mínima $\alpha \in \mathbb{R}_+$, queremos listar os ET-n-conjuntos fechados com utilidades acimas ou iguais a α.

3. TRABALHOS RELACIONADOS

O problema foi tratado na literatura no caso de uma relação binária *crisp* e nenhuma tolerância a ruído. Se o fechamento é somente imposto na dimensão do suporte e I contém o índice da dimensão dos itens, esse problema é a mineração dos *itemsets* sob a restrição de soma mínima [Ng et al. 1998]. Se I contém ambos índices, temos o problema da mineração dos *itemsets* de alta utilidade [Yao et al. 2004]. [Tseng et al. 2015] introduziu a mineração dos *itemsets* fechados de alta utilidade. Em nossa implementação, o fechamento pode ser imposto em somente algumas dimensões. Porém, será demostrado na Seção 4.3 que estender um padrão sempre leva a um padrão de utilidade maior ou igual. Logo, um ET-n-conjunto fechado pode ser considerado mais relevante que qualquer subpadrão e o analista provavelmente prefere não ver estes subpadrões.

Muitos algoritmos específicos à mineração dos *itemsets* de alta utilidade foram propostos. Diferentemente do presente trabalho, todos eles geram novos padrões de forma clássica: adicionando ao último padrão considerado elementos que sempre pertencem à dimensão dos *itens*. Porém a utilidade não é

[1]Tradução do inglês *ET-n-set* que significa *Error-Tolerant-n-set*.

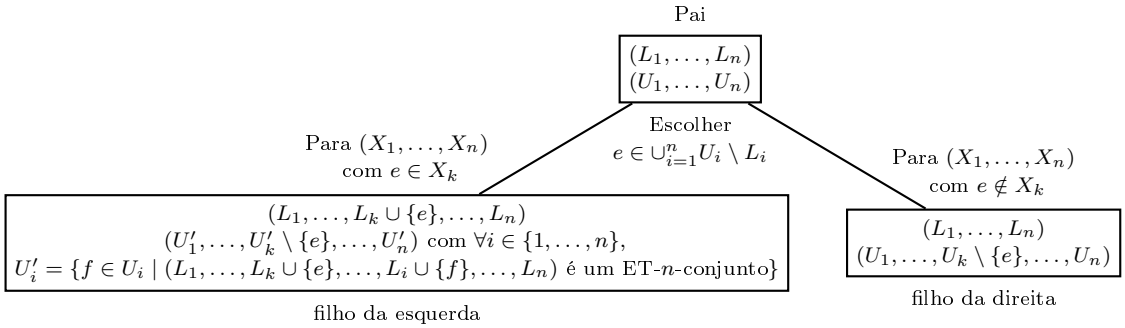


Fig. 1: Travessia do espaço dos padrões por `multidupehack`.

anti-monotônica (como definido em [Ng et al. 1998]) e um novo padrão pode ter uma utilidade maior que aquela do padrão anterior. Os trabalhos passados definem várias cotas superiores da utilidade que possibilitam uma poda do espaço de busca. HUI-Miner [Liu and Qu 2012] e FHM [Fournier-Viger et al. 2014] são os algoritmos estado da arte para a mineração dos *itemsets* de alta utilidade.

4. PODAR OS SUBESPAÇOS DE BUSCA SEM PADRÃO DE ALTA UTILIDADE

4.1 `multidupehack` e restrições

[Cerf and Meira Jr. 2014] já tratou o problema da mineração dos ET-n-conjuntos fechados. O algoritmo descrito nesse artigo, `multidupehack`, aceita restrições adicionais especificadas pelo analista. Cada ET-n-conjunto fechado deve satisfazê-las para pertencer a saída. Dessa forma, o analista especifica características dos ET-n-conjuntos fechados desejados, um processo análogo ao desenho de uma consulta SQL para listar as informações relevantes em um banco de dados.

Além disso, uma grande gama de restrições permite reduções (frequentemente drásticas) do tempo de mineração pois possibilitam uma poda do espaço de busca, ou seja, do espaço dos padrões $\prod_{i=1}^{n} 2^{D_i}$. Para isso, `multidupehack` identifica, ao longo da busca dos ET-n-conjuntos fechados, subespaços de $\prod_{i=1}^{n} 2^{D_i}$ nos quais nenhum padrão satisfaz uma restrição dada. Os subespaços de busca que são considerados derivam da estratégia de travessia. `multidupehack` explora $\prod_{i=1}^{n} 2^{D_i}$ recursivamente. Cada chamada recursiva leva à consideração de um subespaço de busca definido por um limite inferior e um limite superior. O limite inferior de um subespaço de busca é o menor padrão $(L_1, \ldots, L_n) \in \prod_{i=1}^{n} 2^{D_i}$ nele. O limite superior é o maior padrão $(U_1, \ldots, U_n) \in \prod_{i=1}^{n} 2^{D_i}$ nele e sempre temos $\forall i \in \{1, \ldots, n\}$, $L_i \subseteq U_i$. Os padrões no subespaço de busca são os $(X_1, \ldots, X_n) \in \prod_{i=1}^{n} 2^{D_i}$ tal que $\forall i \in \{1, \ldots, n\}$, $L_i \subseteq X_i \subseteq U_i$. Se $(L_1, \ldots, L_n) = (U_1, \ldots, U_n)$, então este padrão é um ET-n-conjunto e `multidupehack` não é chamado recursivamente. Caso contrário, um elemento $e \in \cup_{i=1}^{n} U_i \setminus L_i$ é escolhido e duas chamadas recursivas são realizadas. Elas correspondem a dois novos subespaços de busca que são uma partição do subespaço pai: o subespaço da esquerda no qual todos os padrões contêm e e o espaço da direita no qual e está ausente. A Figura 1 mostra os cálculos dos limites inferiores e superiores desses dois subespaços de busca. Inicialmente, $(L_1, \ldots, L_n) = (\emptyset, \ldots, \emptyset)$ e $(U_1, \ldots, U_n) = (D_1, \ldots, D_n)$. [Cerf and Meira Jr. 2014] descreve `multidupehack` em detalhes.

4.2 Monotonicidade

Estamos, neste artigo, somente interessados no uso dos limites do subespaço de busca para testar a possível existência, no subespaço, de um padrão que satisfaz uma restrição dada. Se não existe essa possibilidade, o subespaço é podado, ou seja, `multidupehack` não é chamado recursivamente (sem perda de padrões que satisfazem a restrição). Entre outras, as restrições monotônicas permitem esse teste. Uma restrição é *monotônica* se e somente se a satisfação da restrição por um padrão implica a sua satisfação por um superpadrão. Matematicamente, escrevendo $\mathcal{C}(X_1, \ldots, X_n)$ para indicar que

o padrão $(X_1, \ldots, X_n)$ satisfaz a restrição $\mathcal{C}$ (vista como um predicado), uma restrição monotônica é tal que:

$$\forall (X_1, \ldots, X_n, Y_1, \ldots, Y_n) \in \left(\prod_{i=1}^{n} 2^{D_i} \right)^2, \mathcal{C}(X_1, \ldots, X_n) \Rightarrow \mathcal{C}(X_1 \cup Y_1, \ldots, X_n \cup Y_n) \ .$$

A contraposição dessa definição é:

$$\forall (X_1, \ldots, X_n, Y_1, \ldots, Y_n) \in \left(\prod_{i=1}^{n} 2^{D_i} \right)^2, \neg\mathcal{C}(X_1 \cup Y_1, \ldots, X_n \cup Y_n) \Rightarrow \neg\mathcal{C}(X_1, \ldots, X_n) \ .$$

Escolhendo $(Y_1, \ldots, Y_n) = (U_1, \ldots, U_n)$ e $(X_1, \ldots, X_n)$ tal que $\forall i \in \{1, \ldots, n\}$, $X_i \subseteq U_i$:

$$\neg\mathcal{C}(U_1, \ldots, U_n) \Rightarrow \neg\mathcal{C}(X_1, \ldots, X_n) \ .$$

Traduzindo em português, se o limite superior do subespaço de busca não satisfaz uma restrição monotônica, nenhum subpadrão satisfaz esta mesma restrição. Vale em particular para os padrões entre os limites inferior e superior, ou seja, o teste que provoca (ou não) a poda do subespaço de busca é simplesmente $\neg\mathcal{C}(U_1, \ldots, U_n)$.

4.3 Monotonicidade da restrição de utilidade mínima

Provamos agora que a restrição "ter uma utilidade acima de α" é monotônica. Por definição (na Seção 2.3), a utilidade de um padrão $(X_1 \cup Y_1, \ldots, X_n \cup Y_n)$ é a soma das utilidades das tuplas em $\prod_{i \in I} X_i \cup Y_i$. Esse conjunto de tuplas pode ser particionado em $\prod_{i \in I} X_i$ e $\Delta = \left(\prod_{i \in I} X_i \cup Y_i\right) \setminus \left(\prod_{i \in I} X_i\right)$. Logo, a utilidade de $(X_1 \cup Y_1, \ldots, X_n \cup Y_n)$ pode ser decomposta da seguinte forma:

$$\forall (X_1, \ldots, X_n, Y_1, \ldots, Y_n) \in \left(\prod_{i=1}^{n} 2^{D_i} \right)^2, \sum_{t \in \prod_{i \in I} X_i \cup Y_i} u(t) = \sum_{t \in \prod_{i \in I} X_i} u(t) + \sum_{t \in \Delta} u(t) \ .$$

A função de utilidade u é, por definição (na Seção 2.3), positiva. Logo, $\sum_{t \in \Delta} u(t) \geq 0$ e:

$$\forall (X_1, \ldots, X_n, Y_1, \ldots, Y_n) \in \left(\prod_{i=1}^{n} 2^{D_i} \right)^2, \sum_{t \in \prod_{i \in I} X_i \cup Y_i} u(t) \geq \sum_{t \in \prod_{i \in I} X_i} u(t) \ .$$

A monotonicidade da restrição segue:

$$\forall \alpha \in \mathbb{R}_+, \forall (X_1, \ldots, X_n, Y_1, \ldots, Y_n) \in \left(\prod_{i=1}^{n} 2^{D_i} \right)^2, \sum_{t \in \prod_{i \in I} X_i} u(t) \geq \alpha \Rightarrow \sum_{t \in \prod_{i \in I} X_i \cup Y_i} u(t) \geq \alpha \ .$$

Como "ter uma utilidade acima de α" é uma restrição monotônica, a poda do espaço de busca explicada na Seção 4.3 se aplica: dado o limite superior $(U_1, \ldots, U_n)$ do subespaço de busca atual, este subespaço é deixado inexplorado se $\sum_{t \in \prod_{i \in I} U_i} u(t) < \alpha$.

4.4 Implementação e complexidade

Após o cálculo dos limites de um novo subespaço de busca, se realiza o teste $\sum_{t \in \prod_{i \in I} U_i} u(t) < \alpha$ que provoca (ou não) a poda. Uma implementação ingênua desse teste leria as utilidades de todas as tuplas em $\prod_{i \in I} U_i$. Porém, se observa na Figura 1 que o limite superior $(U_1^{\text{pai}}, \ldots, U_n^{\text{pai}})$ do subespaço de busca pai é um superpadrão do limite superior $(U_1^{\text{filho}}, \ldots, U_n^{\text{filho}})$ de um subespaço de busca filho. Armazenamos $\sum_{t \in \prod_{i \in I} U_i^{\text{pai}}} u(t)$, a utilidade de $(U_1^{\text{pai}}, \ldots, U_n^{\text{pai}})$, e calculamos por substração $\sum_{t \in \prod_{i \in I} U_i^{\text{filho}}} u(t)$, a utilidade de $(U_1^{\text{filho}}, \ldots, U_n^{\text{filho}})$. Basta subtrair as utilidades das tuplas

em $(\prod_{i\in I} U_i^{\text{pai}}) \setminus (\prod_{i\in I} U_i^{\text{filho}})$. Dessa forma, a utilidade de cada tupla é lida, no máximo, uma vez ao longo da travessia de um ramo qualquer da árvore construída de acordo com a Figura 1 (menos a raiz na qual $\sum_{t\in\prod_{i\in I} D_i} u(t)$ é calculado). A complexidade temporal associada é $O\left(|\prod_{i\in I} D_i|\right)$. Ela pode ser bem menor que $O\left(|\prod_{i\in I} D_i| \times |\cup_{i=1}^{n} D_i|\right)$, o custo das avaliações ingênuas do teste ao longo do ramo (de tamanho $O\left(|\cup_{i=1}^{n} D_i|\right)$).

5. EXPERIMENTOS

A restrição de utilidade mínima foi implementada no `multidupehack`[2], escrito em C++. A compilação é realizada com G++ 5.1 no nível O3 de otimização. Java 7 executa HUI-Miner [Liu and Qu 2012] e FHM [Fournier-Viger et al. 2014], o estado da arte para a mineração dos *itemsets* de alta utilidade. Os autores respectivos forneceram as implementações. Todos os experimentos foram realizados em um sistema operacional GNU/Linux™ rodando por cima de um CPU Intel® Core™ i5-4440 funcionando com frequência de 3,1 GHz. O experimento mais exigente requer menos de 2,6 GB de memória.

5.1 Mineração dos *itemsets* de alta utilidade

A Figura 2 mostra os desempenhos de `multidupehack` (com e sem o fechamento na dimensão dos itens), HUI-Miner e FHM para a mineração dos *itemsets* de alta utilidade em três relações binárias *crisp*: `chess`, `connect` e `foodmart`. Elas foram usadas nos artigos que descrevem HUI-Miner e FHM e os arquivos de entrada correspondentes foram disponibilizados pelos autores. Nos casos de `chess` e `connect`, as utilidades das tuplas foram tiradas aleatoriamente segundo um procedimento descrito em [Liu and Qu 2012]. As utilidades das tuplas em `foodmart` são verdadeiras.

HUI-Miner e FHM têm desempenhos similares. Apesar da sua generalidade, em particular da sua tolerância a ruído (aqui inútil), `multidupehack` lista mais rapidamente que os competidores os *itemsets* de alta utilidade em `chess` e `connect`. Com a menor utilidade mínima usada para `connect`, `multidupehack` é mais de 6,3 vezes mais rápido. Ele é mais lento quando se trata de `foodmart`. Porém, essa relação é de fácil mineração. O tempo de execução nem atinge cinco minutos e quase não varia com a utilidade mínima. Poucos *itemsets* de alta utilidade em `foodmart` são fechados. Menos de 3% quando a utilidade mínima é 500. Como explicado na Seção 3, o analista provavelmente prefere não ver as centenas de milhares de *itemsets* não fechados que são menos informativos e sempre têm utilidades menores que aquelas dos seus superpadrões fechados.

5.2 Mineração sob restrições adicionais dos ET-n-conjuntos fechados

Uma relação ternária *fuzzy* é construída a partir dos dados de conexões e de desconexão de espectadores que assistiram a vídeos em *streaming* no site `Twitch.tv`, especializado na transmissão de *video games*. 1.198.282 espectadores (primeira dimensão) assistiram pelo menos a um dos 94 canais coletados (segunda dimensão) durante as 19 semanas (terceira dimensão) de coleta, do dia 7 de outubro de 2013 ao dia 16 de fevereiro de 2014. Escolhemos os canais que focam no jogo StarCraft II e atingiram um pico de audiência de mais de mil espectadores nas semanas que precederam a coleta. A utilidade de uma tupla é o tempo total em segundos gasto por um espectador assistindo a um canal durante uma semana ($I = \{1, 2, 3\}$). A mesma informação é transformada em um grau de pertinência da tupla à relação através de uma função logística de declividade 0,002 e de ponto médio 3600 (uma hora gasta é associada ao grau de pertinência 0,5). Com 7.388.095 utilidades não nulas (0,35% do total possível), a relação é grande mas esparsa.

`multidupehack` permite o uso de restrições adicionais à utilidade mínima. Elas possibilitam mais podas. Mineramos ET-3-conjuntos fechados com pelo menos três espectadores, três canais e três

[2] `multidupehack` é distribuído segundo os termos da licença GNU GPLv3 na página `http://dcc.ufmg.br/~lcerf/pt/prototipos.html#multidupehack`.

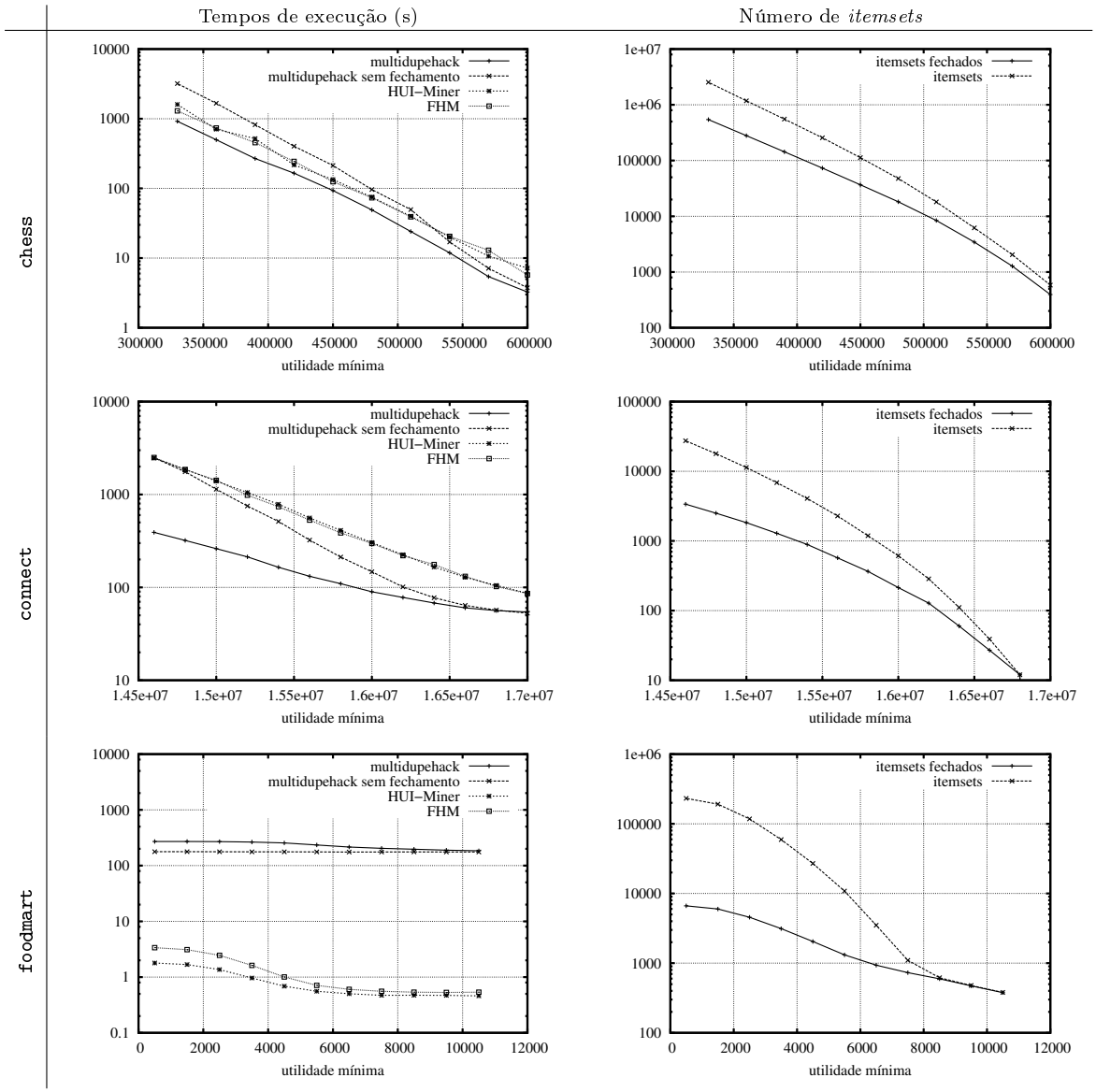


Fig. 2: Mineração dos *itemsets* de alta utilidade em três relações binárias *crisp*.

semanas. Também especificamos (ou não) uma quase-contiguidade na terceira dimensão: as semanas em um ET-3-conjunto fechado têm que ser percorríveis com passos de duas semanas no máximo (ver [Cerf and Meira Jr. 2014] para uma definição matemática). Escolhemos $(\epsilon_1, \epsilon_2, \epsilon_3) = (1, 1, 1)$ como limiares de tolerância a ruído.

A Figura 3 mostra que a restrição de quase-contiguidade não descarta nenhum ET-3-conjunto fechado de alta utilidade, ou seja, eles são naturalmente quase-contíguos. Porém, ela abaixa os tempos de execução por um fator 2,8. O ET-3-conjunto fechado de maior utilidade envolve 83 espectadores que gastaram cerca de 28 milhões de segundos (uma média de 3,9 dias por espectador) assistindo aos canais `mlgsc2`, `wcs_america` e `wcs_europe2` durante as três primeiras semanas de coleta. Esse ET-3-conjunto fechado é relevante. Segundo `http://wiki.teamliquid.net/starcraft2/2013_StarCraft_II_World_Championship_Series`, vários canais transmitiram a edição 2013 dos *StarCraft II World Championship Series*, a maior competição de StarCraft II. `mlgsc2`, `wcs_america` e

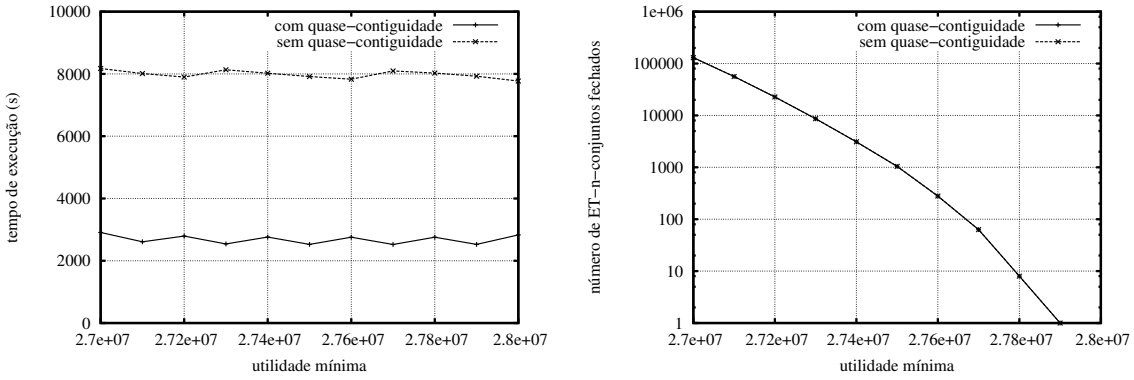


Fig. 3: Mineração dos ET-3-conjuntos fechados de alta utilidade nos dados de `Twitch.tv`.

`wcs_europe2` transmitiram os jogos americanos e europeus que aconteceram durante as três semanas no ET-3-conjunto fechado. Um outro canal, `wcs_gsl`, transmitiu os jogos coreanos mas eles aconteceram antes da coleta. A competição acabou no fim da terceira semana.

6. CONCLUSÃO

Diferentemente da literatura existente, este artigo não propõe um algoritmo específico à mineração dos *itemsets* de alta utilidade. Aqui, "ter uma utilidade acima de um limiar" é visto como uma restrição que possibilita uma poda da busca dos padrões realizada pelo algoritmo `multidupehack`. No contexto de uma relação binária *crisp*, o desempenho obtido é competitivo com o estado da arte. Além disso, a generalidade de `multidupehack` é aproveitada. Os ET-n-conjuntos de alta utilidade são definidos em relações n-árias *fuzzy*. Eles podem ser fechados e minerados sob restrições adicionais que também podam a busca dos padrões. Graças a isso, o analista pode levar em conta mais informação para especificar características dos ET-n-conjuntos desejados e obtê-los rapidamente. `multidupehack` é visto aqui como um sistema genérico que possibilita a consulta de padrões em relações n-árias *fuzzy*, um processo análogo à consulta de tuplas com um sistema de gerenciamento de banco de dados. Nessa perspectiva, este artigo traz a condição `WHERE SUM` $\geq \alpha$ da SQL para a mineração de padrões.

ACKNOWLEDGMENT

Dedico este artigo a Vitor Hugo Pereira que contribuiu para o trabalho. Ele não está mais conosco.

REFERENCES

Cerf, L. and Meira Jr., W. Complete discovery of high-quality patterns in large numerical tensors. In *Proceedings of the International Conference on Data Engineering*. Chicago, USA, pp. 448–459, 2014.

Fournier-Viger, P., Wu, C.-W., Zida, S., and Tseng, V. S. FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In *Proceedings of the International Symposium on Methodologies for Intelligent Systems*. Roskilde, Denmark, pp. 83–92, 2014.

Liu, M. and Qu, J. Mining high utility itemsets without candidate generation. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. Maui, USA, pp. 55–64, 2012.

Ng, R. T., Lakshmanan, L. V. S., Han, J., and Pang, A. Exploratory mining and pruning optimizations of constrained association rules. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. Seattle, USA, pp. 13–24, 1998.

Tseng, V. S., Wu, C.-W., Fournier-Viger, P., and Yu, P. S. Efficient algorithms for mining the concise and lossless representation of high utility itemsets. *TKDE* 27 (3): 726–739, 2015.

Yao, H., Hamilton, H. J., and Butz, C. J. A foundational approach to mining itemset utilities from databases. In *Proceedings of the SIAM International Conference on Data Mining*. pp. 482–486, 2004.

Initialization Heuristics for Greedy Bayesian Network Structure Learning

Walter Perez Urcia
and
Denis Deratani Mauá

Instituto de Matemática e Estatística, Universidade de São Paulo, Brazil
wperez@ime.usp.br,denis.maua@usp.br

Abstract. A popular and effective approach for learning Bayesian network structures is to perform a greedy search on the space of variable orderings followed by an exhaustive search over the restricted space of compatible parent sets. Usually, the greedy search is initialized with a randomly sampled order. In this article we develop heuristics for producing informed initial solutions to order-based search motivated by the Feedback Arc Set Problem on data sets without missing values.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning

Keywords: Bayesian networks, machine learning, local search

1. INTRODUCTION

Bayesian Networks are space-efficient representations of complex multivariate probability distributions [Jensen 2001]. They are defined by two components: (i) a directed acyclic graph (DAG) encoding the (in)dependence relationships among the variables in the model; and (ii) a collection of local conditional probability distributions of each variable given its parents.

Manually specifying a Bayesian network is a difficult task, and practitioners often resort to "learning" the model from data. A common approach to learning a Bayesian network consists of associating every DAG with a polynomial-time computable score value and searching for structures with high score values [Cooper and Dietterich 1992; Lam and Bacchus 1994; Margaritis 2003; Tessyer and Koller 2005]. The score value of a structure usually rewards structures that assign high probability of observing the data set (i.e., the data likelihood) and penalizes the complexity of the model (i.e., the number of parameters). Some examples are the Bayesian Information Criterion (BIC) [Cover and Thomas 1991], the Minimum Description Length (MDL) [Lam and Bacchus 1994] and the Bayesian Dirichlet score (BD) [Heckerman et al. 1995]. An alternative approach is to learn the DAG by multiple conditional independence hypothesis testing [Spirtes and Meek 1995; Cheng et al. 2002]. Although both approaches can recover the true DAG (if one exists) given infinite data and computational resources, testing for independence introduces a lot of false positives and it is often followed by a score-based approach [Tsamardinos et al. 2006].

Score-based Bayesian network learning from data is a NP-hard problem [Chickering et al. 2004], even when the in-degree (i.e., maximum number of parents) of the graph is bounded. For this reason, the most common approach is to resort local search methods that find an approximate solution [H. Friedman and Peér 1999; Chickering 2002]. A popular and very effective method for learning

Bayesian networks is to perform a local search on the space of topological orderings [Tessyer and Koller 2005]. The search is usually initialized with an ordering sampled uniformly at random from the space of orderings. This can make the search converge to a poor local optima unless more sophisticate techniques are employed [Elidan et al. 2002], which can add significant computational overhead. An alternative solution is to initialize the search in high-scoring regions.

In this work we design two new heuristics for generating good initial solutions to order-based Bayesian network structure learning. The first heuristic follows the observation that only orderings consistent with a relaxed version of the problem (in which cycles are permitted) can lead to an optimal structure. Although this heuristic biases the search away from regions which are *guaranteed* to be sub-optimal, it generates orderings with equal probability in any other region. Our second heuristic refines the first one by selecting high scoring orderings among the ones that are consistent with the relaxed version solution. We do this by reducing the problem to a variant of the Feedback Arc Set Problem (FASP), which is the problem of transforming a cyclic direct graph into a DAG. Our experiments show that using these new methods improves the quality of order-based local search.

The rest of this paper is structured as follows: we begin in Section 2 explaining greedy search approaches to learning Bayesian networks. Then in Section 3 we describe the new algorithms for generating initial solutions. Section 4 shows the experiments using both approaches and comparing them (in scoring and number of iterations needed) with multiple data sets. Finally, in Section 5 we give some conclusions about the new methods.

2. LEARNING BAYESIAN NETWORKS

In this section, we formally define the score-based approach learning of Bayesian networks, and review some of the most popular techniques for solving the problem.

2.1 Definition of the problem

A Bayesian network specification contains a DAG $G = (V, E)$, where $V = \{X_1, X_2, \ldots, X_n\}$ is the set of (discrete) variables, and a collection of conditional probability distributions $P(X_i \mid Pa_G(X_i))$, $i = 1, \ldots, n$, where $Pa_G(X_i)$ is the set of variables that are parents of X_i in G. This definition shows that the number of numerical parameters (i.e., local conditional probability values) grows exponentially with the number of parents (in-degree) of a node (assuming the values are organized in tables). A Bayesian network induces a joint probability distribution over all the variables through the equation $P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Pa_G(X_i))$. Hence, Bayesian networks with sparse DAGs succinctly represent joint probability distributions over many variables.

A *scoring function* $sc(G)$ assigns a real-value to any DAG indicating its goodness in representing a given data set.[1] Most scoring functions can be written in the form $sc(G) = F(G) - \varphi(N) \times P(G)$, where N is the number of records in the data set D, $F(G)$ is a data fitness function (i.e., how well the model represents the observed data), $\varphi(N)$ is a non-decreasing function of data size and $P(G)$ measures the model complexity of G. For example, he Bayesian information criterion (BIC) is defined as $BIC(G) = LL(G) - \frac{\log N}{2} size(G)$, where $LL(G) = \sum_{i=1}^{n} \sum_k \sum_j N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$ is the data loglikelihood, $size(G) = \sum_{i=1}^{n} (|\Omega_i| - 1) \prod_{X_j \in Pa(X_i)} |\Omega_j|$ is the "size" of a model with structure G, n is the number of attributes on D, N_{ijk} the number of instances where attribute X_i takes its kth value, and its parents take the jth configuration (for some arbitrary fixed ordering of the configurations of the parents' values), and similarly for N_{ij}, and Ω_i is the set of possible values for the attribute X_i. Most commonly used scoring functions, BIC included, are *decomposable*, meaning that they can be written as a sum of local scoring functions: $sc(G) = \sum_i sc(X_i, Pa(X_i))$. Another property often

[1]The dependence of the scoring function on the data set is usually left implicitly, as for most of this explanation we can assume a fixed data set. We assume here that the dataset contains no missing values.

satisfied by scoring functions is *likelihood equivalence*, which asserts that two structures with same loglikelihood also have the same score [Chickering and Meek 2004]. Likelihood equivalence is justified as a desirable property, since two structures that assign the same loglikelihood to data cannot be distinguished by the data alone. The BIC scoring function satisfies likelihood equivalence.

Given scoring function $sc(G)$, the score-based Bayesian network structure learning problem is to compute the DAG

$$G^* = \arg \max_{G:G \text{ is a DAG}} sc(G)\,. \tag{1}$$

Provided the scoring function is decomposable, we can obtain an upper bound on the value of $sc(G^*)$ by computing $sc(\overline{G})$, where

$$\overline{G} = \arg \sum_i \max_{Pa(X_i)} sc(X_i, Pa(X_i)) \tag{2}$$

is the directed graph where the parents $Pa(X_i)$ of each node X_i are selected so as to maximize the local score $sc(X_i, Pa(X_i))$. We call the parents of a variable in $\overline{G}$ the *best parent set* (for X_i). Note that $\overline{G}$ usually contains cycles, and it is thus not a solution to equation 1.

2.2 Greedy Search Approaches

Greedy Search is a popular approach used to finding an approximate solution to equation (1). The method relies on the definition of a neighborhood space among solutions, and on local moves that search for an improving solution in the neighborhood of an incumbent solution. Different neighborhoods and local moves give rise to different methods such as Equivalence-based, Structure-based, and Order-based methods. Algorithm 1 shows a general pseudocode for this approach.

Algorithm 1: Greedy Search

```
1   GreedySearch( Dataset D ) : return a BN G
2       G = Initial_Solution(X_1, ..., X_n)
3       For a number of iterations K
4           best_neighbor = find_best_neighbor(G)
5           if score(best_neighbor) > score(G) then
6               G = best_neighbor
7       Return G
```

The main idea of the approach is to start with an initial solution (e.g., a randomly generated one), and for a number of iterations K, explore the search space by selecting the best neighbor of the incumbent solution. Additionally, an early stop condition can be added to verify whether the algorithm has reached a local optimum (i.e., if no local move can improve the lower bound). Several methods can be obtained by varying the implementation of lines 2, 4 and 5, which specify how to generate an initial solution, what the search space is and what the scoring function is, respectively.

2.2.1 *Structure-based.* One of earliest approaches to learning Bayesian networks was to perform a greedy search over the space of DAGs, with local moves being the operations of adding, removing or reverting an edge, followed by the verification of acyclicity in the case of edge addition [Cooper and Dietterich 1992; Grzegorczyk and Husmeier 2008]. The initial solution is usually obtained by randomly generating a DAG, using one of the many methods available in the literature [Ide and Cozman 2002; Melançon and Philippe 2004].

2.2.2 *Equivalence-based.* An alternative approach is to search within the class of score-equivalent DAGs. This can be efficiently achieved when the scoring function is likelihood equivalent by using pDAGs, which are graphs that contain both undirected and directed edges (but no directed cycles) with the property that all orientations of a pDAG have the same score. In this case, greedy search

operates on the space of pDAGs, and the neighborhood is defined by addition, removal and reversal of edges, just as in structure-based search [Chickering 1996; 2002].

2.2.3 *Order-based.* Order-Based Greedy Search is a popular and effective approach, which is based on the observation that the problem of learning a Bayesian network can be written as

$$G^* = \arg\max_{<} \max_{G \text{ consistent with } <} \sum_{i=1}^{n} sc(X_i, Pa(X_i)) = \arg\max_{<} \sum_{i=1}^{n} \max_{P \subseteq \{X_j < X_i\}} sc(X_i, P), \qquad (3)$$

which means that if an optimal ordering over the variables is known, an optimal DAG can be found by maximizing the local scores independently [Heckerman et al. 1995; H. Friedman and Peér 1999; Tessyer and Koller 2005]. This can be made efficiently if we assume G^* is sparse, which is true for many scoring functions [de Campos and Ji 2011].

Order-Based Search starts with a topological ordering L, and greedily moves to an improving ordering by swapping two adjacent attributes in L if any exists. Algorithm 2 shows a pseudocode for the method. The function *swap* in line 6 swaps the values $L[i]$ and $L[i+1]$ in the order L to obtain a neighbor of the incumbent solution.

Algorithm 2: Order-Based Greedy Search

```
1   OrderBasedGreedySearch( Dataset D ) : return a BN
2       L = Get_Order(X_1, ..., X_n)
3       For a number of iterations K
4           current_sol = L
5           For each i = 1 to n-1 do
6               L_i = swap(L, i, i+1)
7               if score(L_i) > score(current_sol)
8                   current_sol = L_i
9           if score(current_sol) > score(L) then
10              L = current_sol
11      Return network(L)
```

The standard approach to generate initial solutions is to sample a permutation of the attributes uniformly at random by some efficient procedure such as the Fisher-Yates algorithm [Knuth 1998]. While this guarantees a good coverage of the search space when many restarts are performed, it can lead to poor local optima. In the next section, we propose new strategies to informed generation of topological orderings to be used as initial solutions in Order-Based search.

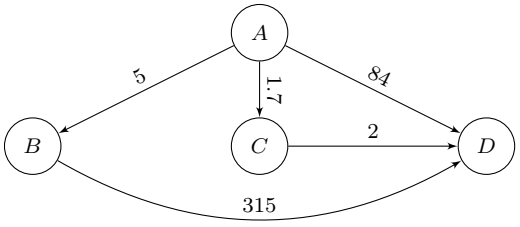
3. GENERATING INFORMED INITIAL SOLUTIONS

As with most local search approaches, the selection of a good initial solution is crucial for avoiding convergence to poor local maxima in Order-Based Learning. Traditionally, this is attempted by randomly generating initial solutions (i.e., a node ordering) in order to cover as much as possible of the search space. In this section, we devise methods that take advantage of the structure of the problem to produce better initial solutions.

3.1 DFS-based approach

We can exploit the information provided by the graph $\overline{G}$ (defined in equation 2) to reduce the space of topological orderings and avoid generating orderings which are guaranteed sub-optimal. Assume the best parent sets are unique, and consider a pair of nodes X_i, X_j in $\overline{G}$ such that X_j is parent of X_i but there is not arc from X_i into X_j. Then, no optimal ordering can have X_i preceding X_j (this can easily be shown by contradiction). Hence, only topological orderings consistent with $\overline{G}$ are potential candidates for optimality, and this number can be much smaller than the full space of orderings. To

see this clearly, consider Figure 1 which shows a possible graph $\overline{G}$ and the corresponding consistent orderings. As can be noticed we have 14 consistent orderings out of $4! = 24$ possible topological orders. This difference is likely to increase as the number of variables increases.



ABDC	CDBA
ACDB	DABC
ADBC	DACB
ADCB	DBAC
BDAC	DBCA
BDCA	DCAB
CDAB	DCBA

Fig. 1: A an example of a fraph $\overline{G}$ and its consistent topological orderings

Taking into consideration the previous analysis, we propose the following algorithm to generate initial solutions. Take as input the graph $\overline{G}$ and mark all nodes as unvisited. While there is an unvisited node, select an unvisited node X_i uniformly at random and add to the list the nodes visited by a depth-first search (DFS) tree rooted at X_i. Finally, return L, an ordering of the nodes.

3.2 FAS-based approach

The DFS approach can be seen as removing edges from $\overline{G}$ such as to make it a DAG (more specifically, a tree), and then extracting a consistent topological ordering. That approach hence considers that all edges are equally relevant in terms of avoiding poor local maxima. We can estimate the arguably relevance of an edge $X_j \to X_i$ by

$$W_{ji} = sc(X_i, Pa^*(X_i)) - sc(X_i, Pa^*(X_i) \setminus \{X_j\}), \tag{4}$$

where $Pa^*(X_i)$ denotes the best parent set for X_i (i.e., its parents in $\overline{G}$). The weight W_{ji} represents the cost of removing X_j from the set $Pa^*(X_i)$ and it is always a positive number because $Pa(X_i)$ maximizes the score for X_i. A small value means that the parent X_j is not very relevant to X_i (in that sense), while a large value denotes the opposite. For instance, in the weighted graph $\overline{G}$ in Figure 1, the edge $C \to D$ is less relevant than the edges $A \to D$, which in turn is less relevant than the edge $B \to D$.

The main idea of our second heuristic is to penalize orderings which violate an edge $X_i \to X_j$ in $\overline{G}$ by their associated cost W_{ij}. We then wish to find a topological ordering of $\overline{G}$ that violates the least cost of edges. Given a directed graph $G = (V, E)$, a set $F \subseteq E$ is called a Feedback Arc Set (FAS) if every (directed) cycle of G contains at least one edge in F. In other words, F is an edge set that if removed makes the graph G acyclic [Demetrescu and Finocchi 2003]. If we assume that the cost of an ordering of $\overline{G}$ is the sum of the weights of the violated (or removed) edges, we can formulate the problem of finding a minimum cost ordering of $\overline{G}$ as a Minimum Cost Feedback Arc Set Problem (min-cost FAS): given the weighted directed graph $\overline{G}$ with weights W_{ij} given by equation (4), find a FAS F such that

$$F = \min_{G-F \text{ is a DAG}} \sum_{X_i \to X_j \in E} W_{ij}. \tag{5}$$

Even though the problem is NP-hard, there are efficient and effective approximation algorithms like the one described in Algorithm 3 [Demetrescu and Finocchi 2003].

Algorithm 3: FAS approximation

```
1   MinimumCostFAS( Graph G ) : Return FAS F
2       F = empty set
```

```
3       While there is a cycle C on G do
4           W_min = lowest weight of all edges in C
5           For each edge (u,v) ∈ C do
6               W_uv = W_uv − W_min
7               If W_uv = 0 add to F
8       For each edge in F, add it to G if does not build a cycle
9       Return F
```

We can now describe our second heuristic for generating initial solutions, based on the minimum cost FAS problem: take the weighted graph $\overline{G}$ with weights W_{ij} as input, and find a min-cost FAS F; remove the edges in F from $\overline{G}$ and return a topological order of the obtained graph $\overline{G} - F$ (this can be done by performing a DFS starting with root nodes).

4. EXPERIMENTS, RESULTS AND DISCUSSION

In order to evaluate the quality of our approaches, we learned Bayesian networks using Order-based greedy search and different initialization strategies from several data sets commonly used for benchmarking. The names and relevant characteristics of the data sets[2] used are shown in Table I, where the density of a graph is defined as the ratio of the number of edges and the number of nodes. For

Dataset	n (#attributes)	N (#instances)	Density of $\overline{G}$
Census	15	30168	2.85
Letter	17	20000	2.41
Image	20	2310	2.45
Mushroom	23	8124	2.91
Sensors	25	5456	3.00
SteelPlates	28	1941	2.18
Epigenetics	30	72228	1.87
Alarm	37	1000	1.98
Spectf	45	267	1.76
LungCancer	57	27	1.44

Table I: Data sets characteristics

each dataset we performed 1000 runs of Order-Based Greedy Search with a limit of 3 parents ($d = 3$) and 100 iterations ($K = 100$), except for the LungCancer dataset where only 100 runs were performed due to the limited computational resources. We used the BIC score and found the best parent sets for a given ordering by exhaustive search.

We compared our proposed initialization strategies, which we call DFS- and FAS-based, against the standard approach of randomly generating an order (called Random). For each strategy, we compared the best score obtained over all runs (Best score), the average initial score (i.e., the score of the best DAG consistent with the initial ordering), the average best score (i.e., the average of the scores of the local searches) and the average number of iterations that local search took to converge. The results are shown in Table II. The results show that in most of the datasets with less than 25 attributes, the Random strategy finds the highest-scoring networks over all runs, even though it finds worse networks on average. The best initial solutions are found by the FAS-based strategy followed by the DFS-based strategy. For datasets with more than 25 variables, Random is less effective in finding high-scoring networks, except for the LungCancer (which has very little data). These results suggest that more informed approaches to generating initial orderings might be more effective in high dimensionality domains, or when the number of restarts is limited e.g. for computational reasons. The proposed

[2]These datasets were extracted from http://urlearning.org/datasets.html

Dataset	Approach	Best Score	Avg. Initial Score	Avg. Best Score	Avg. It.
Census	Random	**-212186.79**	-213074.18 ± 558.43	-212342.26 ± 174.21	7.26 ± 2.90
	DFS-based	-212190.05	-212736.80 ± 379.96	-212339.83 ± 152.26	5.90 ± 2.61
	FAS-based	-212191.64	**-212287.99 ± 92.54**	**-212222.12 ± 70.99**	**3.28 ± 1.67**
Letter	Random	-138652.66	-139774.54 ± 413.74	-139107.13 ± 329.15	6.07 ± 2.50
	DFS-based	-138652.66	-139521.38 ± 396.61	**-138999.84 ± 310.06**	5.75 ± 2.35
	FAS-based	-138652.66	**-139050.43 ± 70.55**	-139039.26 ± 87.97	**2.24 ± 0.96**
Image	Random	**-12826.08**	-13017.13 ± 44.35	-12924.24 ± 41.39	7.59 ± 2.71
	DFS-based	-12829.10	-12999.09 ± 38.56	-12921.13 ± 37.88	7.10 ± 2.47
	FAS-based	-12829.10	**-12930.63 ± 20.83**	**-12882.30 ± 26.43**	**5.05 ± 1.72**
Mushroom	Random	**-55513.38**	-58450.72 ± 1016.54	-56563.84 ± 616.59	7.59 ± 2.76
	DFS-based	**-55513.38**	-58367.11 ± 871.25	-56472.72 ± 546.19	7.75 ± 2.58
	FAS-based	-55574.71	**-56450.49 ± 154.54**	**-56198.66 ± 174.64**	**4.65 ± 1.63**
Sensors	Random	**-62062.13**	-63476.33 ± 265.46	-62726.60 ± 251.26	9.22 ± 2.94
	DFS-based	-62083.21	-63392.60 ± 255.90	-62711.50 ± 257.79	9.65 ± 3.12
	FAS-based	-62074.88	**-62530.26 ± 133.44**	**-62330.94 ± 121.82**	**5.17 ± 2.24**
SteelPlates	Random	-13336.14	-13566.50 ± 65.80	-13429.13 ± 52.14	8.96 ± 3.43
	DFS-based	**-13332.91**	-13572.77 ± 81.12	-13432.30 ± 57.57	9.30 ± 3.38
	FAS-based	-13341.73	**-13485.26 ± 38.27**	**-13397.08 ± 29.53**	**7.77 ± 2.24**
Epigenetics	Random	-56873.76	-57722.30 ± 228.44	-57357.60 ± 222.12	5.89 ± 2.67
	DFS-based	**-56868.87**	**-57615.36 ± 189.17**	**-57308.93 ± 165.18**	6.42 ± 2.47
	FAS-based	**-56868.87**	-57660.09 ± 146.45	-57379.59 ± 148.42	**5.33 ± 2.28**
Alarm	Random	-13218.22	-13324.52 ± 30.49	-13245.43 ± 15.63	10.92 ± 3.24
	DFS-based	**-13217.97**	-13250.72 ± 17.70	-13236.71 ± 12.02	**4.32 ± 2.32**
	FAS-based	-13220.55	**-13249.77 ± 2.57**	**-13233.98 ± 6.19**	6.34 ± 1.74
Spectf	Random	-8176.81	-8202.03 ± 5.23	-8189.69 ± 4.65	7.20 ± 2.17
	DFS-based	**-8172.37**	-8200.04 ± 4.08	-8187.29 ± 4.91	7.86 ± 2.49
	FAS-based	-8172.51	**-8176.98 ± 2.01**	**-8176.07 ± 2.05**	**2.27 ± 1.11**
LungCancer	Random	**-711.23**	-723.79 ± 2.69	-718.03 ± 2.84	5.46 ± 1.78
	DFS-based	-711.36	-720.47 ± 2.51	**-715.29 ± 1.86**	5.02 ± 1.50
	FAS-based	-711.39	**-716.13 ± 0.89**	-715.67 ± 1.19	**2.73 ± 1.79**

Table II: Best score obtained, Average initial score generated, Average best score obtained, Average number of iterations (Avg. It.) using each approach (best values in bold)

strategies are also more robust, which can be seen by the smaller variance of the average initial and best scores.

The results also suggest that the proposed strategies are more effective than Random in datasets for which the graph $\overline{G}$ is sparser (smaller density), showing that pruning the space of orderings can be effective in those cases. The initial orderings provided by the proposed strategies speed up convergence of the local search, as can be seen by the smaller number of average iterations for those strategies in the table.

Overall, the new heuristics are able to improve the accuracy of Order-Based Greedy Search with only a small overhead. Although the differences observed in our experiments were small, we expect greater differences in domains of higher dimensionality.

5. CONCLUSIONS AND FUTURE WORK

Learning Bayesian networks from data is a notably difficult problem, and practitioners often resort to approximate solutions such as greedy search. The quality of the solutions produced by greedy approaches strongly depends on the initial solution. In this work, we proposed two new heuristics for producing topological orderings to be fed into Order-Based Greedy Bayesian network Structure Search methods. One is based on a Depth-First Search traversal of the (cyclic) graph obtained by greedily selecting the best parents for each variable; the other is based on finding an acyclic subgraph

of that same graph by solving a related minimum cost Feedback-Arc Set problem. Experiments with real-world datasets containing from 15 to 57 variables demonstrate that compared to the commonly used strategy of generating initial ordering uniformly at random the proposed heuristics lead to better solutions on average, and increase the convergence of the search with only a small overhead . Although the gains observed in our experiments are small, we expect larger differences for datasets with more variables. A follow-up work should verify this hypothesis.

Our proposed techniques could be adapted to generate initial solutions also for Structure- and Equivalence-based local search methods by returning directed acyclic graphs instead of node orderings. Another extension of this work is to employ the proposed heuristics in branch-and-bound solvers such as [de Campos and Ji 2011] for finding optimal solutions. These ideas are left as future work.

REFERENCES

Cheng, J., Greiner, R., Kelly, J., Bell, D., and Liu, W. Learning bayesian networks from data: An information-theory based approach. *Artificial Intelligence* vol. 137, pp. 43–90, 2002.

Chickering, D. M. Learning equivalence classes of Bayesian-network structures. *Conference on Uncertainty in Artificial Intelligence*, 1996.

Chickering, D. M. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2002.

Chickering, D. M., Heckerman, D., and Meek, C. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research* 5 (1): 1287–1330, 2004.

Chickering, D. M. and Meek, C. Finding optimal Bayesian networks. *Journal of Machine Learning Research*, 2004.

Cooper, G. F. and Dietterich, T. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 1992.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley-Interscience, 1991.

de Campos, C. P. and Ji, Q. Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research* vol. 12, pp. 663–689, 2011.

Demetrescu, C. and Finocchi, I. Combinatorial algorithms for feedback problems in directed graphs. *Information Processing Letters*, 2003.

Elidan, G., Ninio, M., and Schuurmans, N. F. D. Data perturbation for escaping local maxima in learning. *Proceedings of the National Conference on Artificial Intelligence*, 2002.

Grzegorczyk, M. and Husmeier, D. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, 2008.

H. Friedman, I. N. and Peér, D. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. *Conference on Uncertainty in Artificial Intelligence* (15, 1999.

Heckerman, D., Geiger, D., and Chickering, D. Learning Bayesian networks: The combination of knowledge and statistical data. *Journal of Machine Learning Research* 20 (MSR-TR-94-09): 197–243, 1995.

Ide, J. S. and Cozman, F. G. Random generation of Bayesian networks. vol. 2507, pp. 366–376, 2002.

Jensen, F. V. *Bayesian Networks and Decision Graphs*. Springer Science and Business Media, 2001.

Knuth. *The Art of Computer Programming 2*. Boston: Adison-Wesley, 1998.

Lam, W. and Bacchus, F. Learning Bayesian belief networks. an approach based on the MDL principle. *Computational Intelligence* 10 (4): 31, 1994.

Margaritis, D. Learning Bayesian network model structure from data, 2003.

Melançon, G. and Philippe, F. Generating connected acyclic digraphs uniformly at random. *Information Processing Letters* 90 (4): 209–213, May, 2004.

Spirtes, P. and Meek, C. Learning Bayesian networks with discrete variables from data. *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, 1995.

Tessyer, M. and Koller, D. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. *Proceedings of the Conference in Uncertainty in Artificial Intelligence*, 2005.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning* vol. 65, pp. 31–78, 2006.

Social PrefRec framework: leveraging recommender systems based on social information

Crícia Z. Felício[1,2], Klérisson V. R. Paixão[2], Guilherme Alves[2], Sandra de Amo[2]

[1] Federal Institute of Triâgulo Mineiro, Brazil
[2] Federal University of Uberlândia, Brazil
cricia@iftm.edu.br, klerisson@doutorado.ufu.br, guilhermealves@mestrado.ufu.br, deamo@ufu.br

Abstract. Social recommender systems assume a social network among users and make recommendations based on ratings of users that hold a relationship with a given user. However, explicit user's ratings suffer from loss of information. One way to deal with such problem is mining preferences from user's ratings. Even though, for a new user, a preference recommender system also needs techniques to provide accurate recommendations. In this paper, we present Social PrefRec, a social pairwise preference recommender system based on Preference Mining techniques. We focus on leveraging social information on pairwise preference recommender system, corroborating with the idea that matching new people with existing similar people help on providing accurate recommendations. Remark that our approach makes use of social information only on recommendation phase to select among existent recommendation models the most appropriate for a new user. In addition, this is the first step towards a general framework to incorporate social information in traditional approaches, improving upon the state-of-art in this context. We test this idea against two real data sets from Facebook and Flixster. We contribute to this line of work in three ways: (1) Social PrefRec, a social framework for pairwise preference recommender system; (2) a strategy for recommending items based on social metrics; (3) Two publicly available data set of item ratings with social information. For cold start users, the empirical analisys demonstrates that Social PrefRec reaches nDCG@10 equals to 0.9869.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Clustering-Information filtering; J.4 [**Computer Applications**]: Social and behavioral sciences

Keywords: Pairwise preferences, Social Recommender System, Social Network

1. INTRODUCTION

Social recommender systems are becoming increasingly important to help users to find relevant content. This is in part because of social media contents now account for the majority of content published on web. Typical social recommender systems assume a social network among users and makes recommendations based on the ratings of the users that have direct or indirect social relations with the target user [Jamali and Ester 2010]. However, explicit user's ratings suffer from two known drawbacks: (i) The problems of calibration (consistency), which consists in incompatible users ratings on same scale, for example, on 1 to 5 star ratings scale, a rating of 4 for user X might be comparable to a rating of 5 for user Y. (ii) Resolution (granularity), this problem states that any numeric scale for ratings, say 1 to 5 stars, may be insufficient to capture all the users interests without loss of information [Balakrishnan and Chopra 2012] [de Amo and Ramos 2014]. Thus, we advance previous work, PrefRec [de Amo and Oliveira 2014], proposing Social PrefRec a social recommender that applies user preference mining and clustering techniques to incorporate social information on the pairwise preference recommender system.

One of the most significant discussions in recommender system field is the user cold start problem. This

We would like to thank all volunteers who took time to participate in our survey. C. Z. Felício would like to thank Federal Institute of Triângulo Mineiro for granting her study leave. We also thank the Brazilian Research Agencies CAPES, CNPq and FAPEMIG for supporting this work.

problem appears when users do not receive any recommendation, because they had not previously rated any item (user cold start or new user problem). Furthermore, the recommendation process worsens when it faces data sparsity problem. This latter problem is characterized by a system with too many items to be rated and few ratings per user, and also when the number of items rated in common among users is small or zero. Researches related to social recommendation argue that social information can easily deal with new user problem and data sparsity, because instead of relying on user's preferences, which are not available, they use available ratings from users whose hold a relationship with the target user [Ma et al. 2011] [Wang et al. 2014]. In this work, we propose an approach to incorporate social rating network to provide recommendations. To leverage social influence in our model, we exploit several well know social network metrics (Section 3.2).

In addition, model-based social recommender systems in general make use of social information to build recommendation models. Thus, for each new user a new model must be built for each of them. In comparison, our approach harnessing pre-existent models. Instead of building a new model from scratch for each new user, we cluster existent users and generate recommendation models for each group. Through social information we select among existent models the most appropriated for a new user.

The main hypothesis of this paper is that *matching people through their similarities can help on providing accurate recommendations* in a pairwise preference recommender. It is addressed by investigating two research questions:

RQ 1: How accurately social information help on pairwise preference recommendation?
RQ 2: How relevant are the recommendations made by a social pairwise preference recommender?

Main Contributions. The main contributions of this paper can be summarized as follows: (1) The introduction of Social PrefRec, a social recommender system which incorporates social information in pairwise preference approach. (2) Strategies for recommending items based on social metrics. Social PrefRec achieves significantly highly correctness of ranking, calculated using the normalized Discounted Cumulative Gain (nDCG), in particular for cold start users. (3) Two publicly available real life datasets from facebook.com and flixter.com have been used to validate our proposal. The former we crawled and the existing latter we enriched with movie information from imdb.com.

Organization of the Paper. This paper reads as follows. Section 2 presents the background knowledge undertaking in this work. Section 3 describes our proposed framework the Social PrefRec, as well as the applied social metrics and recommender model selection strategies. Section 4 describes our experimental settings and results. Then, Section 5 discusses related work and, finally, Section 6 concludes the paper.

2. BACKGROUND

In this section we briefly introduce the main concepts underlying this work. Due to the lack of space, please refer to de Amo and Oliveira [2014] for more details on pairwise preference recommender systems.

A *preference relation* on a finite set of objects $A = \{a_1, a_2, \ldots, a_n\}$ is a strict partial order over A, that is a binary relation $R \subseteq A \times A$ satisfying the irreflexibility and transitivity properties. We denote by $a_1 > a_2$ the fact that a_1 is preferred to a_2. A *contextual preference model* is modeled as a *Bayesian Preference Network* (BPN) over a relational schema $R(A_1, ..., A_n)$. A BPN is a pair (G, θ) where G is a directed acyclic graph whose nodes are attributes and edges stand attribute dependency, and θ is a mapping that associates to each node of G a set of probability's rules of the form: $A_1 = a_1 \wedge \ldots \wedge A_m = a_m \rightarrow X = x_1 > X = x_2$ where $A_1, \ldots, A_m$, X are item attributes. The left side of the rule is called the context and the right side is the preference on the values of the attribute X. This rule reads: if the values of the attributes $A_1, \ldots, A_m$ are respectively $a_1, \ldots, a_m$ then I prefer x_1 to x_2 for the attribute X. Remark that the preferences on X depends on the values of the context attributes. A contextual preference model is capable to compare items: given two items i_1 and i_2, the model is capable to predict which one is the preferred.

A *recommendation model* is constituted by a set $M = \{(\theta_1, P_1), \ldots, (\theta_k, P_k)\}$, where k is the number of groups in user-item matrix, computed by profile similarities, and for each $i = 1, \ldots, k, \theta_i$ is the consensual preference

vector (preferences' group vector expressed by average of group items rates) and P_i is the preference model extracted from θ_i. The output is a ranking $< i_1, i_2, \ldots, i_n >$ where an item i_k is preferred or indifferent to an item i_m, for $k < m$ and $k, m \in \{1, ..., n\}$.

Considering the relational schema of movies attributes in Table I, we build, from C_1 consensus ratings (Table II), the pairwise preference relation (Table III). Thus, we are able to define the BPN depicted in Fig. 1 and then compare a set of items pair. For more details see [de Amo et al. 2013].

Table I: Movie dataset.

	Title	Decade	Director	Star	Genre
i_1	Gangs of New York	2000	Scorsese	Di Caprio	Drama
i_2	Catch me If You Can	2000	Spielberg	Di Caprio	Drama
i_3	The Terminal	2000	Spielberg	Tom Hanks	Drama
i_4	The Departed	2000	Scorsese	Di Caprio	Thriller
i_5	Shutter Island	2010	Scorsese	Di Caprio	Thriller
i_6	Saving Private Ryan	1990	Spielberg	Tom Hanks	Drama
i_7	Artificial Intelligence	2000	Spielberg	Haley J. Osment	Drama

Table II: Users ratings over movie dataset.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
Ted	5	2	4	1		2	1
Zoe	5	2	4	1	5	1	1
Fred	4		5	1	5		1
C_1	**4.7**	**2.0**	**4.3**	**1.0**	**5.0**	**1.5**	**1.0**
Mary	2		3	5	1		
Rose	1		2	4	2		
Paul	1		3	4	1		
John	2		2	5	2		
C_2	**1.5**	*	**2.5**	**4.5**	**1.5**	*	*

Table III: C_1 pairwise preference relation

$(i_1 > i_2)$
$(i_1 > i_3)$
$(i_3 > i_6)$
$(i_5 > i_6)$
$(i_2 > i_6)$
$(i_5 > i_3)$
$(i_2 > i_4)$
$(i_6 > i_7)$

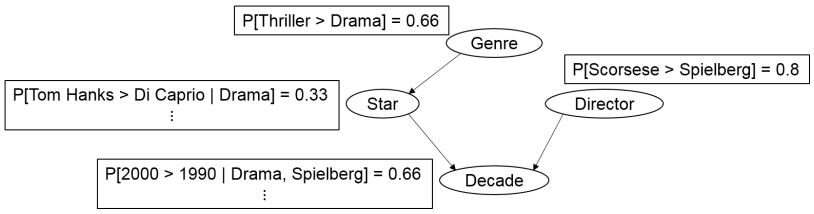


Fig. 1: Bayesian Preference Network over C_1 preferences.

3. SOCIAL PREFREC

Social PrefRec proposes a new approach to address new user problem through social information. It is a PrefRec framework extension, incorporating social information at recommendation module. There were no modifications on how models are built, but at recommendation phase we propose an alternative based on social information to recommend items for new users.

Recommendation process for a new user using social information, in a simple way, could recommend items well rated by his direct friends. Another option is to leverage the tie strength among friends to provide better recommendations. The challenge here is to determine how much influence or similarities exists among user's relationship. Tie strength among users can be computed through similarities on profiles (profession, age bracket, location, etc), interaction between users (messaging, photos, etc) and degree of influence.

To support this features, PrefRec was extended considering the **Social PrefRec** structure: Let U be a user set and I be an item set. The user set U is composed by user identifier and others attributes related to users, where $A_u = \{a_1, ..., a_r\}$ is an attribute set for users. Item set I is composed by item identifier and others attributes related to items, where $A_i = \{a_1, ..., a_t\}$ is an attribute set for items. A *friendship set* over U is defined as $F = \{(u_j, u_k) \mid u_j, u_k \in U\}$, where $(u_j, u_k) = (u_k, u_j)$. We denote by F_j the set that contains all friends of u_j user. The weight function $w : U \times I \to \mathbb{R}$ computes an *user preference degree* for an item and a function $l : F \to \mathbb{R}$ defines *tie strength* between u_j and u_k. Social PrefRec structure, shown on Fig. 2, consists of: one graph $G = (U, I, F, w, l)$. A social network in G is represented by sub-graph $S_N = \{U, F, l\}$. An illustrative example of Social PrefRec is shown on Fig. 3. Nodes represent users and edges are friendships relations. Labels on edges indicate computed tie strength. Dashed groups are computed clusters of users. Each cluster is associated with

a recommendation model. Suppose that Paty is a new user, therefore there is no item previous rated for her. The system already knows some Paty's friends and had previously clustered them. As soon as Paty shows up the tie strength is computed and a suitable recommendation model is selected.

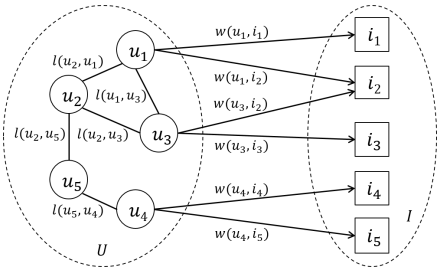


Fig. 2: Social PrefRec structure

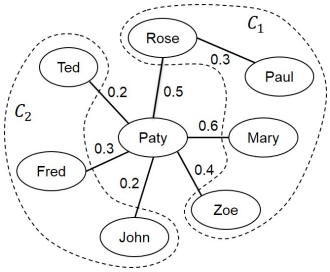


Fig. 3: Social network example

3.1 Social PrefRec Framework

Social PrefRec is an extension of PrefRec, a model-based hybrid recommender system framework using pairwise preferences mining and preferences aggregation techniques [de Amo et al. 2013]. The general Social PrefRec architecture, the interactions among the five modules, as well as their respective input and output is presented at Fig. 4. Note, that modules from 1 to 4 are from PrefRec, however we improved the later system where instead of representing user and consensus preferences in a matrix, now they are represented in a vector. This reduces the algorithm complexity, execution time and allows a better clustering step.

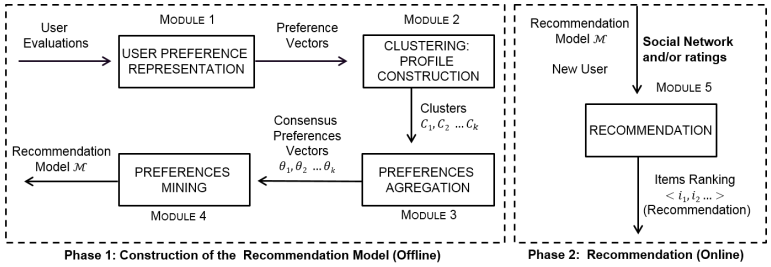


Fig. 4: Social PrefRec Framework

Next, we describe how recommendation module works. Recommendation model is given as input for module 5. This input is constituted by a set $\mathcal{M} = \{(\theta_1, P_1), ..., (\theta_k, P_k)\}$, where for each $i = 1, ..., k$, θ_i is the consensual preference vector associated to cluster C_i and P_i is the preference model extracted from θ_i.

Recommendation Module. The aim of this module is to use the recommendation model $\mathcal{M}$ to recommend items for new users. It is executed online, differently from the previous modules which are executed offline. Recommendation process could be executed using one out of the two strategies:

A) PrefRec. (1) Given a target user u and a (small) set R_u of ratings provided by u on items in I, the first task of Module 5 consists in obtaining the preference vector σ_u corresponding to R_u; (2) the similarity between σ_u and each consensual preference vector θ_i is calculated. Let θ_u be the consensual vector most similar to σ_u; (3) consider the preference model P_u corresponding to θ_u; (4) P_u is used to infer the preference between pairs of items in I which have not been rated by the user u in the past. From this set of pairs of items (i, j) indicating that user u prefers item i to item j, a ranking can be built by applying one ranking algorithm adapted from the algorithm Order By Preferences introduced in [Cohen et al. 1999].

B) Social PrefRec metrics. (1) Given a target user u and a its social network SN_u, the first task of Module 5 consists in applying one of the social strategies (described in Section 3.2) to compute the tie strength between u and it contacts; (2) obtaining the consensual vector θ_u corresponding to the cluster C_k where $u_k \in SN_u$ are u

direct contacts using one of the chosen model methods (Average or threshold, described in Section 3.2) ; (3) and (4) are identical to PrefRec strategy. Note that by using this strategy is possible to recommend to an user without taking in consideration any previous ratings, but considering user's relations in the cluster set.

3.2 Tie strength calculus and Recommendation model selection

We compute tie strength between users through the following metrics: (1) *friendship* considers that $l(u_j, u_k) = 1((u_j, u_k) \in F)$, where $1(\cdot)$ is the characteristic function (1 if argument is true, 0 otherwise); (2) *interaction level* is calculated as $\frac{a(u_j,u_k)}{\widehat{a}(u_j)}$, where $a(u_j, u_k)$ is the number of times where user u_k appears at u_j's timeline and $\widehat{a}(u_j)$ is the number of all occurrences of users u_k at u_j's timeline; (3) *mutual friends* considers that $l(u_j, u_k) = \mathcal{J}(u_j, u_k)$, where $\mathcal{J}$ is the Jaccard similarity coefficient; (4) in *similarity score*, function $l(u_j, u_k) = sims(u_j, u_k)$ is the average of $similarity(u_j, u_k, A_i)$ binary values for all attributes A_i, where $similarity(u_j, u_k, A_i)$ represents user u_j compatibility with an user u_k considering the demographic user attributes A_i (1 if similar, 0 otherwise) like Relationship Status, Age Bracket, Sex, Religion, Location, etc; (5) *centrality* as tie strength is calculated by average of closeness, betweenness and eigenvector centrality measures.

Social PrefRec recommender uses two metrics for model selection based on tie strength value:

- *Minimum threshold*: Let $\varepsilon \in [0, 1]$ be a tie strength minimum threshold. The strategy C_m will select the preference model P_i (associated with model $\mathcal{M}_i \in \mathcal{M}$) with more users who have a tie strength with the target user u_j above a minimum threshold according to Eq. 1.

$$C_m(F_j, \mathcal{M}, u_j) = \arg \max_{\mathcal{M}_i \in \mathcal{M}} \left| \{u_k : (u_j, u_k) \in F_j \wedge l(u_j, u_k) \geq \varepsilon\} \right| \quad (1)$$

- *Average*: The strategy C_a will select the preference model P_i with users who have the highest average tie strength with the target user u_j according Eq.2.

$$C_a(F_j, \mathcal{M}, u_j) = \arg \max_{\mathcal{M}_i \in \mathcal{M}} \frac{1}{|F_j|} \sum_{(u_j,u_k) \in F_j} l(u_j, u_k) \quad (2)$$

4. EXPERIMENTS

4.1 Datasets

Facebook Dataset. We surveyed this data set through a developed Facebook web application. With volunteers permission we crawled relationship status, age bracket, gender, born-in, lives-in, religion, study-in, last 25 posts in user's timeline, posts shared and posts' likes, as well as movies previous rated on Facebook platform. In addition, we asked each volunteer to rate 169 Oscar nominated movies in 1 to 5 star scale. We obtained data from 720 users and 1,454 movies, resulting in 56,903 ratings.

Flixster Dataset. Jamali and Ester [2010] published this dataset. However, movie information was restricted to its title, then we improved it by adding genres, directors, actors, year, languages and countries information retrieved from IMDB.com public data.

In our experiments we considered only the 169 movies surveyed, because there are more common movies rated among users. We split Facebook data into two datasets, FB50 and FB100, to represent the set of users that rated at least 50 and 100 movies, respectively. This was done to evaluate the overall system performance under data sets with different sparsity and level of social information. In Table IV we summarize our datasets. The movies attributes considered were genres, directors, actors, year, languages and countries. In FB50 and FB100, user similarity metric was computed using the attributes: relationship status, age bracket, gender, born-in, lives-in, religion and study-in. The *interaction_level* was computed considering the last 25 posts in user timeline, posts shared and likes. Flixster social information includes friends relationships, mutual friends, friends centrality and users similarities. Similarity between users is computed only through three attributes: gender, age bracket and location. Interaction information is not available on Flixster dataset.

4.2 Experimental Protocol and Evaluation methods

Each experiment was performed against the datasets split into two parts: training and test sets. Fig. 5 shows a comparative scheme of our protocols. PrefRec and Social PrefRec make use of training data to build clusters (K-Means clustering) of similar users. For each cluster they associate a correspondent recommendation model. Then, to recommend items for a given user u, is necessary to select the most similar model (cluster) that fits u. This process is done during test phase. However, those approaches take different directions. Since PrefRec is not able to deal with social information, it relies only on previous ratings of u to select its best recommendation model, whereas Social PrefRec needs social information for this choice. To better validate our tests we apply one adaptation of traditional validation protocols: for each iteration one user is taken for test purposes and the remaining users assemble the training set. In this case we have n iterations, where n is the number of users.

Table IV: Movies Datasets

Features	FB100	FB50	Flixster
# of users	230	361	357
# of items	169	169	625
# of ratings	35,458	44,925	175,523
Sparsity	8.77%	26.36%	21.33%
Friends relationships	1,330	2,926	706
Avg friends per user	6.4	8.6	2.8
Avg rates per user	154.16	124.44	491
Users without friends	9.56%	5.54%	29.97%

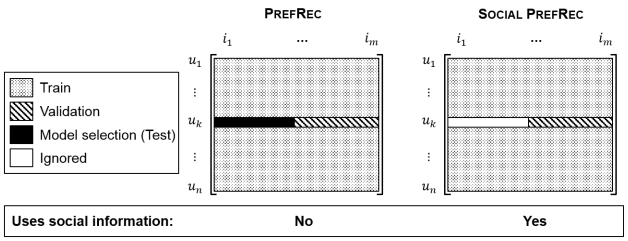


Fig. 5: Scheme of experimental protocols

PrefRec protocol. The PrefRec recommendation model is built offline. For the test phase m random ratings of current test user u_k were considered for the choice of the most similar cluster C_i. Then, calculating similarity between u_k and c_i is a matter of calculating the *Euclidian distance* between their respective ratings arrays $p = (p_1, \ldots, p_n)$ and $q = (q_1, \ldots, q_n)$. Remark that this similarity distance was used for models clustering (training) and selection models (test) phases. Finally, for validation purpose, the remaining ratings of current test user u_k were used.

Social PrefRec protocol. There is no difference regarding recommendation model building on Social PrefRec from PrefRec. However, during test phase, we do not take in account any rating. It relies on social information to find the most similar cluster, C_i, according to a given social metric and a model selection strategy.

Regarding our evaluation methods we present results from two metrics: (1) ***nDCG*** is a standard ranking quality metric to evaluate the ability of the recommender to rank the list of top-k items [Shani and Gunawardana 2011]. (2) We also compute the standard ***F1 score***, based on precision and recall, to evaluate the predictions quality of a pairwise preference items [de Amo and Oliveira 2014].

4.3 Results

RQ1: Quality of recommendation. Comparative F_1 scores can be seen in Fig. 6, for minimum threshold ($\epsilon = 0.4$) and tie strength average selection model strategies. In all datasets with a profile length of 30-ratings scenario for PrefRec versus 0-ratings for Social PrefRec, social metrics achieve better results using Minimum threshold strategy. Rate-15-items baseline is widely used to bootstrap traditional recommender systems [Chang et al. 2015]. Thus, to make a fair comparison we provide 30-ratings for PrefRec, which means that all runs have a good safe margin and should not harm its performance.

A Kruskal-Wallis test was performed to check statistical significance among social metrics performance and PrefRec. Regarding Mutual Friends, Interaction, Similarity there are no significant differences. Also Friendship and Centrality results are not significant different from PrefRec (profile length = 30-ratings) result. Thus, the test shows, with 95% confidence, that with the three first metrics we can better accurately recommend in social 0-rating profile scenario than 30-rating profile in a traditional recommender approach. The others social metrics achieved the same result as the traditional approach, but they do not need any previous rate from a given user.

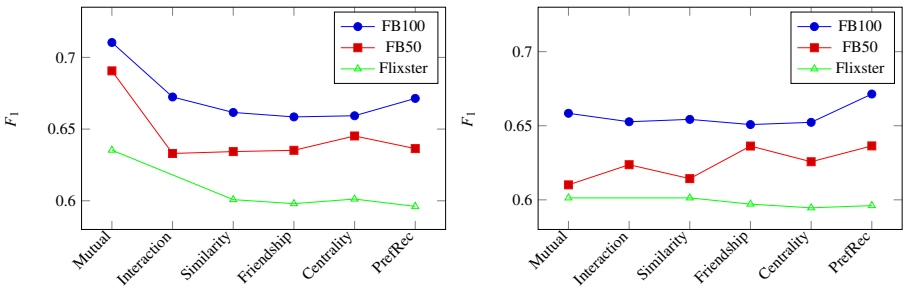
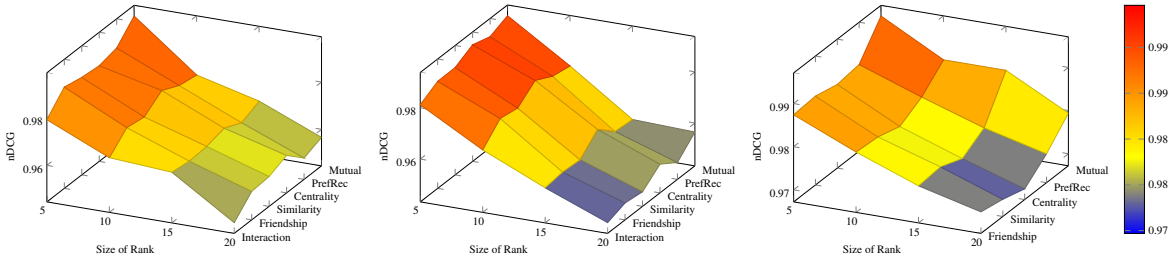


Fig. 6: (a) Minimum threshold ($\epsilon = 0,4$), (b) Tie strength average

RQ2: Relevance of recommendation. Fig. 7 shows the *nDCG* results for rank size 5, 10, 15 and 20, considering Minimum threshold ($\epsilon = 0.4$) strategy. Rank quality is better in Flixster dataset, because the number of items is greater than Facebook data, generating a richer preference model. Test of statistical significance shows, with 95% confidence, that Mutual Friends metric is better than others. The performance with Centrality achieves equivalent score as PrefRec. Finally, Similarity, Friendship and Interaction results are not significant different.



(a) FB100 (b) FB50 (c) Flixster

Fig. 7: nDCG@5,@10,@15 and @20 for PrefRec versus Social PrefRec Metrics

5. RELATED WORK

Pairwise Preference Recommendation. Balakrishnan and Chopra [2012] have proposed an adaptive scheme in which users are explicitly asked for their relative preference between a pair of items. Though it may provide an accurate measure of a user's preference, explicitly asking users for their preference may not be feasible for large numbers of users or items, or desirable as a design strategy in certain cases. Park *et al.* [2009] proposed a pairwise preference regression model to deal with cold start user problem. We corroborate with their idea. They argue that ranking of pairwise users preferences minimize the distance between real rank of items and then could lead to better recommendation for a new user. On the same direction Sharma and Yan [2013] propose a probabilistic latent semantic indexing model for pairwise learning, which assumes a set of users' latent preferences between pairs of items. We build on previous work [de Amo and Ramos 2014] by adapting a pairwise preference recommender to leverage a graph of information, social network.

Social Recommender. This research field especially started because social media content and recommender systems can mutually benefit from one another. Many social-enhanced recommendation algorithms are proposed to improve recommendation quality of traditional approaches [Canamares and Castells 2014] [Alexandridis et al. 2013]. Moreover, the works of Ma *et. al.* [2008] [2011] [2011] are the most related to this one. No matter what techniques are developed, the basic assumption employed in these works is that users' social

relations can positively reflect users' interests similarities. Although we also explore users' relation in our approach, we do it in different way. Instead of embedding social information in the recommendation models, we built a loosely coupled approach based on clustering techniques to incorporate social relation into our system.

6. CONCLUSION

In this paper, we have devised and evaluated Social PrefRec, an approach whose ultimate goal is to help pairwise preferences recommender systems to deal with cold start problem. We built on the shoulders of others, and expand previous work by: (1) Working in a way to incorporate social information in pairwise preference recommender approach; (2) presenting strategies for recommending items based on several social metrics; and (3) evaluating the resulting approach on two real life data sets from facebook.com and flixter.com, which we made publicly available[1]. This work opens several avenues for future research. First, it is worth exploring the use of others networks (graphs) where we can compute a tie from similarities scores among nodes, such as scientific networks. Furthermore, we ought to empirically compare Social PrefRec performance against benchmark social recommenders. From the application point of view, we believe that Social PrefRec framework could be generalized to others hybrid model-based recommenders, allowing traditional approaches to incorporate contextual social information.

REFERENCES

Alexandridis, G., Siolas, G., and Stafylopatis, A. Improving social recommendations by applying a personalized item clustering policy. In *5th RecSys Workshop on Recommender Systems and the Social Web co-located with the 7th ACM Conference on Recommender Systems*. Hong Kong, China, 2013.

Balakrishnan, S. and Chopra, S. Two of a kind or the ratings game? adaptive pairwise preferences and latent factor models. *Frontiers of Computer Science* 6 (2): 197–208, 2012.

Canamares, R. and Castells, P. Exploring social network effects on popularity biases in recommender systems. In *6th Workshop on Recommender Systems and the Social Web (RSWeb 2014) at the 8th ACM Conference on Recommender Systems (RecSys 2014)*. Foster City, CA, USA, 2014.

Chang, S., Harper, F. M., and Terveen, L. Using groups of items for preference elicitation in recommender systems. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. CSCW '15. ACM, New York, NY, USA, pp. 1258–1269, 2015.

Cohen, W. W., Schapire, R. E., and Singer, Y. Learning to order things. *J. Artif. Int. Res.* 10 (1): 243–270, May, 1999.

de Amo, S., Bueno, M. L. P., Alves, G., and da Silva, N. F. F. Mining user contextual preferences. *JIDM* 4 (1): 37–46, 2013.

de Amo, S. and Oliveira, C. Towards a tunable framework for recommendation systems based on pairwise preference mining algorithms. In *Advances in Artificial Intelligence*, M. Sokolova and P. van Beek (Eds.). Lecture Notes in Computer Science, vol. 8436. Springer International Publishing, pp. 282–288, 2014.

de Amo, S. and Ramos, J. Improving pairwise preference mining algorithms using preference degrees. In *29th Brazilian Symposium on Databases*. SBBD '14. Curitiba, Brazil, pp. 107–116, 2014.

Jamali, M. and Ester, M. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10. ACM, New York, NY, USA, pp. 135–142, 2010.

Ma, H., Yang, H., Lyu, M. R., and King, I. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. CIKM '08. ACM, New York, NY, USA, pp. 931–940, 2008.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. WSDM '11. ACM, New York, NY, USA, pp. 287–296, 2011.

Ma, H., Zhou, T. C., Lyu, M. R., and King, I. Improving recommender systems by incorporating social contextual information. *ACM Trans. Inf. Syst.* 29 (2): 9:1–9:23, Apr., 2011.

Park, S.-T. and Chu, W. Pairwise preference regression for cold-start recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*. RecSys '09. ACM, New York, NY, USA, pp. 21–28, 2009.

Shani, G. and Gunawardana, A. Evaluating recommendation systems. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor (Eds.). Springer US, pp. 257–297, 2011.

Sharma, A. and Yan, B. Pairwise learning in recommendation: Experiments with community recommendation on linkedin. In *Proceedings of the 7th ACM Conference on Recommender Systems*. RecSys '13. ACM, New York, NY, USA, pp. 193–200, 2013.

Wang, D., Ma, J., Lian, T., and Guo, L. Recommendation based on weighted social trusts and item relationships. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. SAC '14. ACM, New York, NY, USA, pp. 254–259, 2014.

[1] `http://www.lsi.facom.ufu.br/~cricia/`

From the sensor data streams to linked streaming data. A survey of main approaches

K. R. Llanes[1], M. A. Casanova[1], N. M. Lemus[2]

[1] Pontificia Universidade Católica do Rio de Janeiro, Brazil
kllanes@inf.puc-rio.br, casanova@inf.puc-rio.br
[2] Laboratório Nacional de Computação Científica, Brazil
noelml@lncc.br

Abstract. Nowadays, large amounts of data are produced by sensor networks. They are continuously producing information about real world phenomena in the form of data streams. However, these data are generated in raw and different formats, lacking of semantic to describe their meanings, which imposes barriers in accessing and using them. To tackle this problem several solutions using Linked Data Principles have been proposed. In this paper we provide a survey about the main solutions developed by the research communities for publishing stream data in the web of data, identifying its strengths and limitations. Over that basis, the main steps that someone should follow to plublish its data streams in a manner that anyone can use them with a minimal understanding of data details, are defined; which represets the main contribution of this work. We also highlight the main challenges that emerge from this survey, concluding with a list of research tasks for future work.

Categories and Subject Descriptors: H.2.5 [**Database Management**]: Heterogeneous Databases; C.2.3 [**Computer-Communication Networks**]: Network Operations

Keywords: data streams, linked data, semantic web, sensor data publishing

1. INTRODUCTION

In recent years, data sensors networks have been deployed in various domains (medical sciences for patient care using biometric sensors, wildfire detection, meteorology for weather forecasting, satellite imaging for earth and space observation, agricultural lands, etc). The sensors are distributed across the globe, capturing and continuously producing an enormous amount of information about a number of real world phenomena in the form of data streams.

However, commonly, the data produced by sensors networks is in raw and different formats, lacking of semantic to describe their meanings. This failure intensifies the current traditional problem "too much data and not enough knowledge" [Sheth et al., 2008] and imposes barriers in accessing and using sensor data in applications and linking them with other related data sources.

To tackle this problem several solutions using Linked Data Principles [Berners-lee, 2006] have been proposed. They allow integrate sensor technologies with semantic web technologies in order to publish sensor data streams in enriched and standardized way, so they can be accessed and consumed by external applications. The publication process consists of transforming the data streams in linked streaming data following the Linked Data Principles.

The process of publishing data streams in the Linked Open Data (LOD) cloud is relatively similar to publishing static data. Nevertheless, during data streams publishing the time component must be taken into account, which substantially changes the way of data processing.

The publishing of static data in LOD cloud is composed of several activities: specification, modeling, generation, publication and exploitation. Specification refers to a preliminary set of tasks to identify and analyze the data to be published. Then we need to select the ontology or ontologies to be used for modeling and semantically describing data. After that, they are transformed to standard representation in RDF format [http://www.w3.org/RDF] and linking with external data sources through generation activity. This activity ends with a meaningful and enriched triple set. During the publication activity, this enriched triple set is stored and published in a triple store to be consumed later. Once data are published in the Web of Data, they would be queried and consumed. The activity that takes care of these tasks is the exploitation, which is the goal of the publication process as whole.

In multiple domains, the time component is critical to make right decisions quickly. In terms of data streams processing, it implies that it is done in real-time. That means data from sensors observations should be processed on-the-fly with a minimum delay. To fulfill this requirement, significant modifications to the traditional static data publishing process should be made, such as incorporate data compression, data streams abstraction, continuous queries and creating real-time links among others.

Several efforts have been developed for publishing data streams on the Web of Data. From them, some take into account the real-time and others do not consider it. In this paper we provide a survey about the main works available in the research literature, identifying their weaknesses and strengths. Based on shortcomings and lacks of them we propose a set of next research tasks to facet in the near future.

The remainder of the paper is organized as follow: section 2 describes in detail the main steps that someone should follow to publish its data streams in a way that anyone can use them with the minimal understanding of data details. Section 3 shows the most relevant approaches proposed to publish data streams on the Semantic Web following the Linked Data Principles. Section 4 discusses lessons learned and open challenges emerged from this survey. Section 5 concludes the paper and presents our future research directions.

2. SENSOR DATA PUBLISHING ON THE SEMANTIC WEB

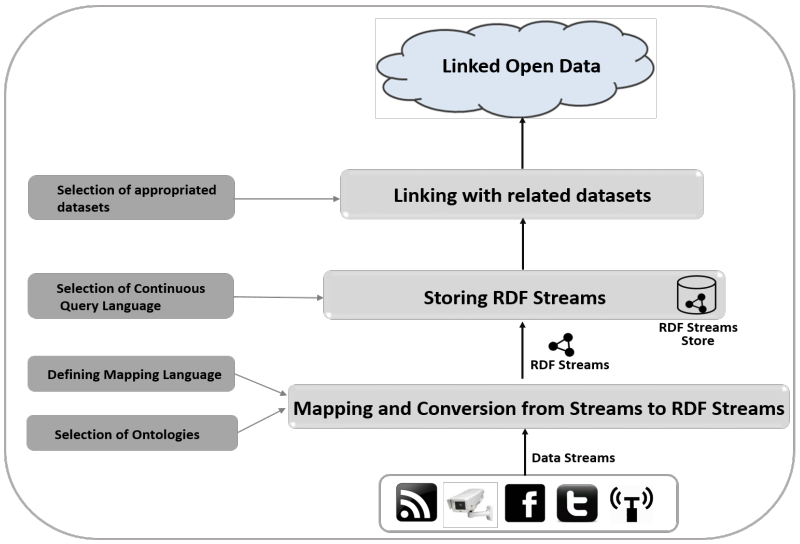


Fig. 1 Data Stream Publication Process

Sensor networks employ various types of hardware and software components to observe and measure

physical phenomena and make the obtained data available through different networking services. Applications and users are typically interested in querying various events and requesting measurement and observation data from the physical world.

Through the process of sensor data publishing in the semantic web are generated semantic streams (RDF streams) that satisfy this high-level information need, which include the data streams captured by sensors and their semantics, see Figure 1. This process encompasses three main stages: mapping and conversion from data streams to RDF streams, storing RDF streams and linking them with related data sources existing in the LOD cloud. To carry out this process a set of important tasks are required such as: ***(i)selection of ontologies to semantically describe data streams***, ***(ii) defining the mapping language to do the conversion*** , ***(iii) selection of continuous queries languages*** and ***(iv) choosing the appropriated datasets from LOD cloud to create the links***. To support the complete process a stream publishing framework is being developed.[1]

2.1 Selection of ontologies

With the development of semantic sensor networks a number of ontologies describing the sensor networks domain have been brought forth in the past years. A detailed survey was performed by Michael Compton et al. in [Compton et al., 2009], where eleven sensor network ontologies were analyzed. Therefore, considering the need of standardization regarding sensor networks ontologies, was formed the Semantic Sensor Network Incubator Group from W3C, with the purpose of developing ontologies for sensor networks and search for appropriated methods for enhancing available standards with semantic technologies. Due to the efforts of this group arises the Semantic Sensor Network (SSN) ontology [Compton et al., 2012], which can describe capabilities, measurements and resultant observations from sensors.

On the other hand the W3C Semantic Sensor Network Incubator Group developed a methodology to perform semantic annotations over data generated by sensors following the standards defined by Open Geospatial Consortium (OGC). These standards help to describe observed phenomena such as space, time and theme.

Spatial metadata provide information regarding the sensor location and data, in terms of either a geographical reference system, local reference, or named location. Temporal metadata provide information regarding the time instant or interval when the sensor data is captured. Thematic metadata describe a real world state from sensor observations, such as objects or events. All these metadata are very important, because they play an essential role in managing sensor data and provide more meaningful description and enhanced access to sensor data.

Both projects developed by W3C Semantic Sensor Network Incubator Group: the SSN ontology (SSNO) and the proposed methodology, facilitate the stream data semantic fusion applications and the integration of stream data with linked data sets, because the fact does not only publish the streaming data, but also integrate them with other related datasets. Sometimes, sensor ontologies are not able to provide all the semantics needed by a scientific system and additional ontologies are often required.

2.2 Defining the mapping language

Several languages have been proposed by the semantic web research communities for expressing customized mappings from relational databases to RDF datasets. Such mappings provide the ability to view existing relational data in RDF data model, expressed in a structure and target vocabulary of the mapping author choice. D2R [Bizer, 2003], R2RML [Consortium, 2012] and R2O [Barrasa et al., 2004] are some of them. They are fruitfully on transforming static relational data to RDF, but present some disadvantages to face the challenge of convert data streams to RDF streams.

[1] https://github.com/nmlemus/streams2LSD

Despite the existence of this gap, solutions for streaming data mapping and querying using ontology-based approaches have been little explored. Calbimonte et al. [Calbimonte et al., 2010] presented an extension of R2O called S2O for data stream to RDF mapping. Also, Harth et al. [Harth et al., 2013] developed an extension for R2RML with the same purpose. These last extensions are most suitable during publishing stream data process.

2.3 Selection of continuous queries languages

Languages such as SPARQL are designed to execute queries over RDF triples, but they do not have functionalities to query RDF streams. To face this challenge some continuous RDF query languages have been proposed.

Barbieri et al. [Barbieri et al., 2009] introduced Continuous SPARQL (CSPARQL) as the extension of SPARQL for querying RDF streams. It supports continuous queries, registered and continuously executed over RDF data streams, considering windows of such streams. C-SPARQL is currently not designed to handle large volumes of data, which constitutes their main weakness.

SPARQLstream [Calbimonte et al., 2011] is an extension to SPARQL for RDF streams. It has been inspired by previous proposals C-SPARQL and SNEEQL [Brenninkmeijer and Galpin, 2008], but with some improvements that can be summarized as: it only supports windows defined in time; the result of a window operation is a window of triples, not a stream, over which traditional operators can be applied. It uses S2O and R2RML for the definition of stream-to-ontology mappings. Its main disadvantage is that currently does not support querying on both stream and RDF dataset.

Ainic et al. [Anicic and Fodor, 2011] developed Event Processing SPARQL (EPSPARQL). It is a continuous query language that uses a black box approach backed by a logic engine. It translates queries into logic programs which are then executed by a Prolog engine. EP-SPARQL provides a unified execution mechanism for event processing and stream reasoning which is grounded in logic programming. The main deficiency of EPSPARQL is that its performance drops significantly for complex queries.

Le Phuoc [Phuoc, 2013] presented Continuous Query Execution over Linked Stream (CQELS), an adaptive execution framework for Linked Stream Data and Linked Data. CQELS provides a flexible architecture for implementing efficient continuous query processing engines over Linked Data Stream and Linked Data.

From the best of our knowledge the more complete approach to do continuous queries over the RDF streams is CQELS presented by Le Phuoc, because despite it has scalability troubles with respect to multiple concurrent queries, the CQELS engine can achieve better performance than other black box systems by order of magnitude. It represents a solution for RDF stream processing built on top of the notion of linked stream data. The solution offers a native way to interpret and implement common stream processing futures (time windows operator, relational database like join and union operators, and stream generation operator) in RDF data stream processing environment.

2.4 Choosing LOD datasets to create the links

Other important task in the process of data streams publishing on the Semantic Web is the selection of the most suitable triple sets with which RDF streams may be interlinked. It will allow users to take advantage of existing knowledge. Once the most suitable triple sets are found, the next step is to link them with a local sensor triple set, thus completing the process of publishing. However, interlinking is a laborious task. Thus, users interlink their triple sets mostly with data hubs, such as DBpedia and Freebase, ignoring the more specific yet often even more promising triple sets. To alleviate this problem, some triple sets interlinking recommendation tools have been implemented.

Lopes et al. [Lopes et al., 2013] presented a tripleset recommendation approach using strategies

borrowed from social networks. To generate the ranked list, the procedure uses a recommendation function adapted from link prediction measures used in social networks. The tool obtains high levels of recall and reduces in up to 90% the number of triplesets to be further inspected for establishing appropriate links.

Caraballo et al. [Caraballo et al., 2014] presented a web-based application, called TRTML, that explores metadata available in Linked Data catalogs to provide data publishers with recommendations of related triplesets. TRTML combines supervised learning algorithms and link prediction measures to provide recommendations. It high precision and recall results demonstrate the usefulness of TRTML.

Lopes et al. [Lopes et al., 2014] developed RecLAK. It is a Web application developed for the LAK Challenge 2014 focused on the analysis of the LAK dataset metadata and provides recommendations of potential candidate datasets to be interlinked with LAK dataset. RecLAK follows an approach to generate recommendations based on Bayesian classifiers and on Social Networks Analysis measures. Furthermore, RecLAK generates graph visualizations that explore the LAK dataset over other datasets in the Linked Open Data cloud.

The main disadvantage of these tripleset recommendation tools is that they are not able to do the recommendation process on-the-fly, since it is not designed to act in real-time, which represents a gap in the process of sensor data publishing. For this reason, the current solution is to choose the LOD datasets related with each new sensor that will be incorporated to sensor network, using the tools described above, before sensor start to capture observations and do not add the sensors to the sensor network ad-hoc.

3. MAIN APPROACHES ABOUT SENSOR DATA PUBLISHING ON THE SEMANTIC WEB

Although the main goal of Linked Stream Data is to make available the sensor data in the LOD cloud in real-time, quite few projects have achieved it. In this section the most recently efforts of research communities to publish sensor data on the Semantic Web will be analyzed. Some of them do not publish sensor data in real-time, which is its main weakness, but may serve as starting point for future work.

3.1 Non real-time approaches

Le-Phuoc et al. [Phuoc and Hauswirth, 2009] presented an approach and an infrastructure which makes sensor data available following the linked open data principles and enables the seamless integration of such data into mashups. This project publishes sensor data as web data sources which can then easily be integrated with other linked data sources and sensor data. Also, it allows users to describe and annotate semantically raw sensor readings and sensor. These descriptions can then be exploited in mashups and in linked open data scenarios and enable the discovery and integration of sensors and sensor data at large scale. The user generated mashups of sensor data and linked open data can in turn be published as linked open data sources and be used by other users.

Patni et al. [Patni et al., 2010] presented a framework to make this sensor data openly accessible by publishing it on the LOD cloud. This is accomplished by converting raw sensor observations to a standard representation in Resource Description Framework (RDF) and linking with other datasets on LOD. With such a framework, organizations can make large amounts of sensor data openly accessible, thus allowing greater opportunity for utilization and analysis. They were the first to add to the LOD cloud a large dataset of sensor descriptions and measurement, by first representing it in Observation and Measurements (O&M) standard.

Barnaghi and Presser [Barnaghi et al., 2010] proposed a platform called Sense2Web for publishing sensor data description defined by spatial, temporal and thematic attributes. The platform offers an interface for publishing linked sensor data without requiring from the users a semantic technological

background. The sensor observation and measurement data can also be published following similar principals. However, publishing observation and measurement data raises other concerns such as time-dependency, scalability, freshness and latency.

Moraru et al. [Moraru et al., 2011] proposed a system for publishing sensor data following the linked data principles and providing hereby integration with the Semantic Web. In their proposal they focused on a single sensor source and for storing sensor data they used a relational database, which represents its main deficiency; because a relational database is not prepared to support the continuous arriving of data streams.

3.2 Real-time approaches

Barbieri et al. [Barbieri and Valle, 2010] proposed an approach to publish data as linked data streams. The approach uses C-SPARQL to register and run continuous queries over streams of RDF and C-SPARQL engine to publish the retrieved data as Linked data streams. To represent RDF in RDF streams, they proposed the use of two named graphs: stream graph (S-graph) and instantaneous graph (I-graph). An RDF stream can be represented using one s-graph and several i-graphs, one for each timestamp. The main limitation of this approach is that is only a prototype, and it does not have a finished application that supports it.

Le Phuoc [Le-Phuoc et al., 2011] proposed a Linked Stream Middleware, a platform to facilitate publishing Linked Data Stream and making it available to other applications. It provides the following functionalities: wrappers to access stream data sources and transform the raw data into Linked Stream Data, data annotation and visualization through web interface and life querying over unified Linked Stream Data and data from the LOD cloud. Besides processing real-time data, it is also necessary to store the data generated, either for queries defined over a time period or for archiving purposes. It is here where it appears the main limitations of this approach: the triple storage cannot efficiently handle high update rates; the materializing sensor reading into triples is also inefficient, especially numeric readings and also, it runs into performance issue with complicated queries.

Hasemann et al. [Hasemann et al., 2012] proposed Platform-independent Wiselib RDF Provider for embedded Internet of Thing (loT) devices such as sensor nodes. It enables the devices to act as semantic data providers. They can describe themselves, including their services, sensors, and capabilities, by means of RDF documents. The greatest contribution of this proposal is the introduction of Streaming HDT, a lightweight serialization format for RDF documents that allows for transmitting compressed documents with minimal effort for the encoding. Also a platform allows to publish and share sensors data with reduce level of cost, less complexity of sensors data integration, and easy to access the integrated sensors data.

Harth et al. [Harth et al., 2013] developed a web architecture that enables (near) real-time access to data sources in a variety of formats and access modalities. Also, it enables rapid integration of new live sources by modeling them with respect to domain ontology and automatically transforming all the arrived data streams from their format (CSV, TSV, JSON) to RDF and publishing them following the Linked Data Principles. This approach is a very good approximation to solve the problem related to the integration of new sensor devices into the LOD cloud, but it is still immature project.

As we mentioned in section 2, we are developing a stream publishing framework using the Linked Data Principles that tries to solve the gaps presented in the proposals described above.

4. LESSON LEARNED AND OPEN CHALLENGES

4.1 Lesson Learned

During our study we have realized there are valuable lessons to be taken into account for publishing data streams in the LOD cloud in real-time:

(1) There are several ontologies designed to semantically describe sensor data that help us during annotation process. However, sometimes sensor ontologies are not able to provide all the semantics and additional ontologies are often required.
(2) Before starting the transformation process from data streams to RDF, it is extremely important to make an abstraction of streams to select the most significant data and not spend time to process those less relevant streams.
(3) An efficient and lightweigh serialization format for RDF should be used, in order to transmit compressed documents with minimal effort for encoding.
(4) Integrate information from heterogeneus sources (sensor networks and social networks) in order to support decision making in real-time. Integrating sensor data with data from social networks, allows you to capture human perception, implying that better decisions are made.

4.2 Open Challenges

In order to integrate sensor technologies with semantic web technologies and publish them as Linked Streaming Data in real-time, some efforts have been made. Nevertheless, some challenges are still being faced:

(1) To publish and consume data from sensors data streams in real-time it is necessary a lighter mapping language, capable of guaranteeing on demand mapping and an efficient conversion from sensors data streams to RDF streams.
(2) The conversion of the data streams to RDF streams must be on-the-fly. This restriction captures the idea that the data must be continuously triplified, albeit with limited delay. To fulfill this requirement, techniques for efficient triplification should be developed.
(3) The interlinking process of RDF streams with data sources of the LOD cloud must be on-the-fly. To address the restriction of minimum delay, interconnection techniques should be based on a strategy of preprocessing or caching data to accelerate the creation of links.
(4) The lack of an efficient RDF storage that supports the real-time stream processing. Although the classical RDF storages are efficient to store RDF that will persist over time, they are not efficient to handle the RDF streams, because the RDF streams need to be stored, accessed and processed on-the-fly.

5. CONCLUSIONS AND FUTURE WORK

Real-time publishing of sensor data streams based on semantic technologies is indeed not only possible, but also find actual applicability in many areas. In this paper we present a study that covers the main approaches proposed to publish the sensor data in the LOD cloud from 2009 to present, identifying its main contributions and limitations. We describes in detail the main steps that someone should follow to publish its data streams in a maner that anyone can use them with the minimal understanding of data details and the must suitable tool for use at every step. Based on the limitations of current approaches, we are developing a stream publishing framework to cover the gaps. Also we discuss the ongoing challenges existing and with the aim of cope some of them we propose the following directions of future work:

(1) Conclude the implementation of framework that has being developed.
(2) Develop a NoSQL systems as compelling alternative to distributed and native RDF stores for simple workloads. Considering its strengths, the very large user base that it has, and the fact that there is still ample room for query optimization techniques, we are confident that NoSQL databases will present an ever growing opportunity to store and manage RDF data in the LOD cloud.

REFERENCES

Anicic, D. and Fodor, P. (2011). EP-SPARQL: a unified language for event processing and stream reasoning. *Proceedings of the 20th conference on World wide web*.

Barbieri, D., Braga, D., and Ceri, S. (2009). C-SPARQL: SPARQL for continuous querying. *Proceedings of the 18th international conference on World wide web*, (c).

Barbieri, D. and Valle, E. D. (2010). A proposal for publishing data streams as linked data-a position paper.

Barnaghi, P., Presser, M., and Moessner, K. (2010). Publishing linked sensor data. In *CEUR Workshop Proceedings: Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN), Organised in conjunction with the International Semantic Web Conference*, volume 668.

Barrasa, J., Corcho, O., and Gómez-pérez, A. (2004). R2O, an Extensible and Semantically based Database-to-Ontology Mapping Language. In *in In Proceedings of the 2nd Workshop on Semantic Web and Databases(SWDB2004*, pages 1069–1070. Springer.

Berners-lee, T. (2006). Linked Data - Design Issues.

Bizer, C. (2003). D2R Map: A Database to RDF Mapping Language. In *12th World Wide Web Conference*, pages 2–3, Budapest, Hungary.

Brenninkmeijer, C. and Galpin, I. (2008). A semantics for a query language over sensors, streams and relations. *Sharing Data, Information and Knowledge*, pages 87–99.

Calbimonte, J., Corcho, O., and Gray, A. (2010). Enabling ontology-based access to streaming data sources. *The Semantic Web ISWC*, (September):1–16.

Calbimonte, J., Jeung, H., Corcho, O., and Aberer, K. (2011). Semantic sensor data search in a large-scale federated sensor network. In *4th International Workshop on Semantic Sensor Networks 2011. 23 October, 2011.*, Bonn, Germany.

Caraballo, A., Júnior, N., and Nunes, B. (2014). TRTML-A Tripleset Recommendation Tool based on Supervised Learning Algorithms. In *11th Extended Semantic Web Conference*, Anissaras, Crete, Greece.

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., and Taylor, K. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:25–32.

Compton, M., Henson, C., and Neuhaus, H. (2009). A Survey of the Semantic Specification of Sensors. *SSN*.

Consortium, W. W. W. (2012). R2RML: RDB to RDF mapping language.

Harth, A., Knoblock, C., and Stadtmüller, S. (2013). On-the-fly Integration of Static and Dynamic Linked Data. In *12th International Semantic Web Conference*, number 257641, Sydney, Australia.

Hasemann, H., Kremer, A., Pagel, M., Group, A., and Braunschweig, T. U. (2012). RDF Provisioning for the Internet of Things.

Le-Phuoc, D., Quoc, H. N. M., Parreira, J. X., and Hauswirth, M. (2011). The linked sensor middleware–connecting the real world and the semantic web. *Proceedings of the Semantic Web Challenge*, 152.

Lopes, G., Leme, L., Nunes, B., and Casanova, M. (2014). RecLAK: Analysis and Recommendation of Interlinking Datasets. In *4th Int. Conf. on Learning Analytics and Knowledge*, Indianapolis, USA.

Lopes, G. R., Andr, L., Leme, P. P., Nunes, B. P., Casanova, M. A., and Dietze, S. (2013). Recommending Tripleset Interlinking. In *14th International Conference on Web Information System Engineering*, number i, pages 149–161, Nanjing, China.

Moraru, A., Fortuna, C., and Mladenic, D. (2011). A System for Publishing Sensor Data on the Semantic Web. *CIT. Journal of Computing and Information Technology*, pages 239–245.

Patni, H., Henson, C., and Sheth, A. (2010). Linked sensor data. In *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, pages 362–370. IEEE.

Phuoc, D. and Hauswirth, M. (2009). Linked open data in sensor data mashups. *In proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09), in conjunction with ISWC*, pages 1–16.

Phuoc, D. L. (2013). *A Native and Adaptive Approach for Linked Stream Data Processing*. PhD thesis, National University of Ireland.

Sheth, A., Henson, C., and Sahoo, S. (2008). Semantic sensor web. *Internet Computing, IEEE*, 12(4):78–83.

Analyzing the Correlation Among Traffic Loop Sensors to Detect Anomalies in Traffic Loop Data Streams

Gustavo Souto, Thomas Liebig

Dortmund University, Germany
gustavo.souto@tu-dortmund.de, thomas.liebig@tu-dortmund.de

Abstract. This work aims to analyze whether traffic loop data sensors hold any correlation among them which could support the process to detect anomalies in traffic data stream. In order to find out such a correlation among them we apply a Statistical Baseline Method along with a Sensor Correlation Analysis (SCA) approach. The statistical model analyzes in an unsupervised manner the data distribution in order to detect the events that are three times standard deviation or greater than a threshold ($3 \times \sigma^2 + \mu$) and then passes them to the SCA which in turn analyzes whether an event in a sensor S_k also affected its nearest sensor in time period ΔT after the statistical model detects it. We evaluate our approach by comparing the detected anomalies against traffic alerts which are emitted by Traffic Agents on Twitter.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.2.6 [**Artificial Intelligence**]: Learning

Keywords: anomaly detection, data stream, spatio-temporal correlation, traffic loop sensors

1. INTRODUCTION

Anomaly detection is the process of finding patterns which deviate much from the normal behavior of the data. As result, this process might find one of the following types of anomalies: point anomaly, contextual anomaly, and collective anomaly [Chandola et al. 2009]. The literature also refers anomaly as outliers, abnormalities, discordant or deviants [Aggarwal 2013] and an Event can be described as an occurrence of an anomaly in a certain place during a particular interval of time, Equation 1 [Artikis et al. 2014; Souto and Liebig 2015]. Anomaly detection has applications in Stocks Exchange, Health Care, Network Security as well as in other fields of the industry and science.

$$E =< timestamp, location < lat, long >, cause > \tag{1}$$

In literature, a data stream is defined as a continuous, high-speed and unbounded source of data in which the data arrives as an uncontrollable sequence. This paradigm has recently emerged due to the continuous data problem [Bifet et al. 2011], and therewith this process holds important challenges, specially in the field of anomaly detection. Data stream analysis process imposes some constraints such as processing of the data in a limited amount of memory and in a limited quantity of time, be able to process at any point, and receive a data point at a time and inspect it in at most only once. An approach for anomaly detection in data stream depends also on some particular factors about the data domain. For instance, an approach which desires to detect anomalies in *spatio-temporal* data should take into account the autocorrelation between spatial and temporal features. The vehicle traffic data is an example of spatio-temporal data which has gained more attention in recent years

Fig. 1. Locations of SCATS sensors (marked by red dots) within Dublin, Ireland. Best viewed in color.

due to its importance in city traffic planning. By analyzing traffic data is possible to detect some events such traffic jams and accidents. The Figure 1 depicts the SCATS[1] sensors within Dublin, Ireland. See Section 4.1 for more details. Unfortunately, the SCATS data emitter and Dublinked[2] do not provide training dataset or ground truth which could provide us insights about what is normal and/or what is an anomaly in Dublin traffic data. Therefore, building a classification model to detect traffic anomalies is not possible since we do not have such a training dataset directly. It is known supervised methods are more reliable than unsupervised ones, but the task to label data could be very time-consuming depending on the size of data as well as, in the most of the cases, a domain expert must manually label the data. Therefore, our aim is to analyze whether the traffic loop data sensors hold any correlation among them which could indicate low-level anomalous events in traffic loop data stream. This work applies a basic statistical model ($3 \cdot \sigma^2$) which is baseline method along with *Sensor Correlation Analysis* (SCA) approach to detect low-level anomalous events in traffic loop data stream through the spatio-temporal correlation among traffic loop sensors. This statiscal model is applicable to SCATS data, because it is modeled by a Gaussian distribution. The statistical model analyzes the data distribution in an unsupervised manner in order to detect the events that are three times standard deviation or greater than this threshold and then passes them to the *SCA* which in turn analyzes whether an event in a sensor S_k also affects its nearest sensor in the time period $\Delta T1$ after the statistical model detects it. Some important questions arise from this approach and we aim to answer them in this work: "*Does an event at a sensor S_k affect its nearest sensor S_w within a time-period t?*","*How often is the nearest Sensor affected by an event which takes place at another Sensor?*", and "*Does the correlation among traffic loop sensors help the detection of traffic anomalies?*"

This work is structured as follows: Section 2 discusses the related works about anomaly detection in traffic data streams, Section 3 describes our approach to analyze the correlation among traffic loop data sensors, Section 4 presents our experiments, and finally, the conclusion in the Section 5.

2. RELATED WORK

Stolpe et. al. propose [Stolpe et al. 2013] a Vertically Distributed Core Vector Machines (VDCVM) algorithm for anomaly detection which is based on Core Vector Machine (CVM) algorithm [Bădoiu and Clarkson 2002]. The VDCVM has two components, the Central Node P_0 which coordinates the entire system and the Data Node $P_1 ... P_k$ which detects the anomalies in a distributed manner. The Data Node has two more sub-components, the Worker and Data Repository. The anomaly is

[1]Sydney Coordinated Adaptive Traffic System (SCATS)

[2]Dublinked (http://www.dublinked.com/) is a data sharing network which provides different datasets from Dublin, Ireland.

detected locally by each Worker through a local model and sent to the Central Node along with a small sample of all observations. Then, the Central Node trains a global model on such a sample and used to define whether the sent observation is an anomaly or not. The advantage of this work is the good communication cost between Workers and the Central Node in the training phase, but this approach cannot detect anomalies which are global due to a combination of features, and that is its disadvantage.

In [Yang et al. 2014], Yang et. al present a non-parametric Bayesian method, or Bayesian Robust Principal Component Analysis (RPCA) - BRPCA, to detect traffic events on road. This method takes the traffic observations as one dimension data (1-D) and converts it into a matrix format which in turn decomposes it into a superposition of low-rank, sparse, and noise matrices. The idea of BRPCA is to improve the traffic detection by sharing a sparsity structure among multiple data streams affected by the same events. Such an approach uses multiple homogeneous data streams and a static weather data source in the detection process. The advantage of this work is the generation of a ground truth by 3 expertises in the traffic domain which reviewed different plots. However, the approach is limited to detect only 3 types of traffic events which are Slow down, Unexpected high traffic volume and Traffic jam.

Guo et al. [Guo et al. 2014] propose a traffic flow outlier detection approach which focuses on the pattern changing detection problem to detect anomalies in traffic conditional data streams. The traffic data comes from inductive loop sensors of four regions in United State and United Kingdom, as well as this works makes use of a short-term traffic condition forecasting system to evaluate the proposed approach. This approach performs the analysis of the incoming data point after the data point be processed by Integrated Moving Average filter (IMA) which captures the seasonal effect on the level of traffic conditional series, and then Kalman filter picks up the local effect flow levels after IMA, and GARCH filter models and predict time-varying conditional variance of the traffic flow process. These filters constitute together the integrated forecast system aforementioned. Although the results present good performance about the detection of outliers. This work does not apply another procedure to verify the uncertainty of the detection (e.g. check a different source such as traffic alerts on social networks), that is, whether that event is a real anomaly, or not.

Trilles et al. [Trilles et al. 2015] propose a variation of CUmulative SUM (CUSUM) algorithm in Storm Framework[3] to detect anomalies in data streams near to Real-Time. This approach is only applied when the observations are in-control, that is, the data is normally distributed. In the anomaly detection process the CUSUM is obtained by computing $Y_i = Y_{i-1} z_i$, where z_i is the standard normal variable which is computed as follows $z_i = \frac{x_i - \bar{x}}{s}$, where the s is the Standard Deviation of time series. The events are detected by the Equation 2, if Y_{H_i} exceeds the threshold (CUSUM control charts) $\hat{A} \pm h\sigma_x$ ($h = 5$ and σ_x is the Standard Deviation), then it is an *Up-Event* due its increasing and if Y_{L_i} is greater than threshold (CUSUM control charts) $\hat{A} \pm h\sigma_x$ ($h = 5$ and σ_x is the Standard Deviation), then it is an *Down-Event* due its decreasing. The k variable ("Slack") is the reference value which is usually set to be one half of the mean. The advantages of this work are the application of a simple approach for Real-Time anomaly detection and the dashboard application to visualize the detected events. However, the work does not present experiments with a data source wich has high refresh rate such as SCATS data stream.

$$Y_{H_i} = MAX[0, (z_i - k) + Y_{H_i} - 1] \qquad Y_{L_i} = MIN[0, (z_i - k) + Y_{L_i} - 1] \tag{2}$$

Other works also propose solutions to detect anomaly traffic events such as [Yang and Liu 2011], [Liu et al. 2011], [Pang et al. 2013], [Pan et al. 2013], [Yang et al. 2014], [Liu et al. 2014], [Liu et al. 2014]. However, these solutions make use of moving sensors such as GPS, and we have been focusing

[3]Storm Framework: https://storm.apache.org/

on Static sensors (e.g., SCATS sensors) since our work deals with such a kind of sensors as well as the literature present fewer works using Static sensors than Moving sensors.

Although these works present some substantial advances in the field of anomaly detection in data streams, the field is still in its early stage, and therewith it is possible to see that such works hold some drawbacks which were already discussed as well as open tasks such as incorporate expert knowledge in anomaly detection in traffic of vehicles. Incorporation of expert knowledge data is an interesting research direction which should receive more attention in future, because expert knowledge on the relationship between events may improve detection of anomalous event patterns. None of presented related works approached expert knowledge, but [Schnitzler et al. 2014] and [Liebig et al. 2013] are good references. These works use *Street Network* from OpenStreetMap[4] (OSM) that is a kind of expert knowledge in the process to detect traffic anomalies.

3. TRAFFIC LOOP SENSOR ANALYSIS

In order to find out whether the traffic loop sensors hold some spatio-temporal correlation among them which might support in the anomaly detection process. We apply a statical baseline method along with a SCA approach. The statistical model analyzes the SCATS data stream in order to find (vehicle) flow values which are above some threshold. The detected events are sent to SCA process which analyzes the spatio-temporal correlation of anomalous events over a close sensor, at this process we make use of Street Network data from OpenStreetMap which is a kind of expert knowledge to find close sensors. Our approach to find the spatio-temporal correlation among sensors in the anomaly detection process consists of the following components: Feature Selection, Data Segmentation, Data Summarization, Anomaly Detection and Sensor Correlation Analysis (SCA). These components are implemented on the Storm Framework which was designed to process data streams. The idea to analyze the spatio-temporal correlation among anomalies is possible since the position of all sensors are static and a sensor holds its nearest sensor at close range as seen in Figure 1.

The *Feature Selection (Input) Component* makes the connection to the data source which receives the data stream in a JSON format. It also selects the set of features for the next processes, see more about the SCATS data stream in 4.1.

In order to check a fixed time period of the vehicle traffic the *Data Segmentation Component* performs a segmentation of traffic flow of each traffic sensor according to a specific traffic time period $\Delta T2$ (e.g. 15, 30, 45 or 60 Minutes of traffic). A Fixed Sliding Window approach is applied and the segmentation process adds the most recent data point and discard the oldest one in the segment.

The *Data Summarization Component* summarizes the segment of a time period $\Delta T2$ by computing statistical measures, the mean (μ) and standard deviation (σ^2) (Equation 3), and Upper Bound Limit (Equation 4).

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \qquad \sigma^2 = \frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N} \tag{3}$$

The *Anomaly Detection Component* analyzes the traffic flow of each sensor and whether this component detects a value above the (upper bound limit) threshold, Equation 4, (i.e., the statistical model in this work considers solely the *upper bound limit* since there is not negative traffic flow), it considers that the sensor holds an anomalous event and send the event for further analysis to *SCA* component, otherwise the component discards the event, because our aim is to analyze the correlation among the sensors and their influence on the detection of traffic anomalies. The event is sent in the form of Equation 1; *cause* of the anomaly is the trigger condition of the anomaly detection component:

[4] openstreetmap.org

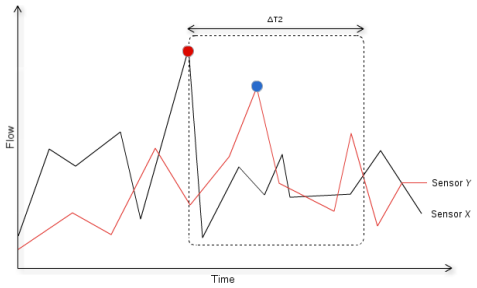


Fig. 2. SCA Approach.

'unexpected high traffic'.

$$Threshold = 3 \times \sigma^2 + \mu \tag{4}$$

The *Sensor Correlation Analysis (SCA) Component* analyzes the correlation among sensors by checking the spatio-temporal correlation among detected events and close sensors. SCA approach works as following: an event E takes a place at sensor S_x and whether during a time period $\Delta T1$ in the future (e.g. 30 Minutes) ($\Delta T1 \neq \Delta T2$) its nearest sensor S_y is affected by the event E, then the event E should be more reliable than the one which does not hold any correlation between two close sensors. The Equation 5 depicts the main principle of SCA component to check the correlation among sensors. The process to find nearest sensors makes use of Street Network data from OSM which is a kind of expert knowledge. The process queries the Street Network data every time an anomaly is sent to this component, the data is stored in PostgreSQL DB by applying the extension for spatial data called PostGIS. Whether the correlation does not hold true, the component discards the event. Figure 2 depicts the SCA approach.

$$Sen_x \Rightarrow Sen_y \Leftrightarrow E(t, Sen_x) \wedge E(t + \Delta t, Sen_y) \tag{5}$$

4. EXPERIMENTS

In order to check whether the SCATS sensors hold some spatio-temporal correlation in the process of anomaly detection we have performed some experiments which apply a statistical baseline model along with the SCA approach as well as compare the detected events against a ground truth. We also apply map matching by plotting both data to compare the results.

Dublin traffic agents such as AARoadWatch[5] and GardaTraffic[6] emit traffic alerts on Twitter. In our experiments these alerts (*Tweets*) are used as ground truth data and compared against the detected events in order to find out how much the SCATS sensors are correlated among them in the process of anomaly detection. On 26 June 2015 the traffic agents has informed 4 events about the traffic in Dublin. For instance, the alert "*DUB: Crash on D'Olier St before College St. This will add to delays in the area.*" was emitter by *AA Roadwatch* at *09:25*.

4.1 Data source

The Sydney Coordinated Adaptive Traffic System (SCATS) is an adaptive urban traffic management system that synchronizes traffic signals to optimise traffic flow across a network [McCann 2014]. SCATS data is time series, because SCATS sensors measure the traffic flow and density over the time,

[5]http://www.theaa.ie/aa/aa-roadwatch.aspx/
[6]http://www.garda.ie/Controller.aspx?Page=111

Table I. Number of anomalous events according to the size of segment by applying SCA approach and not applying SCA (NoSCA) and the number of anomalous events suing SCA which match to any alert from the ground truth data (MGT).

Size	**15**	**30**	**45**	**60**
NoSCA	1929	5234	6210	6759
SCA	32	138	173	223
MGT	0	0	0	0

Table II. Comparing the detected anomalies by applying SCA against traffic alerts (GT) in order to check whether they match (MGT) as well as the percentage of loss candidates per day (LC).

Day	**17/06/2015**	**18/06/2015**	**19/06/2015**	**20/06/2015**	**21/06/2015**	**22/06/2015**
NoSCA	1849	1867	1755	2036	2362	2001
SCA	27	37	24	29	37	35
GT	30	32	9	6	4	6
MGT	0	0	0	0	0	0
LC	98.53%	98.01%	98.63%	98.57%	98.43%	98.25%

that is, it provides information about flow of vehicles and the rate of use (density) of the streets. In Dublin, 506 SCATS sensors are present in their 4 non-overlapped regions (CCITY, NCITY, SCITY and WCITY). The SCATS data stream is emitted in a JSON format and it is high-dimensional with 74 features. However, this work uses a small set of features as follows: *sensor number*, *timestamp*, *latitude*, *longitude* and *flow*, because our approach evaluates the flow of sensor and uses coordinates to find its nearest sensor. The feature selection occurs in the data stream component as can be seen in Section 3. In our experiments we have used SCATS data stream which was measured from 17 to 22 June 2015 as well as 26 July 2015.

4.2 Results

The Table I depicts the number of anomalous events according to the size of segment on 26 June 2015 by applying the SCA approach and without SCA (NoSCA) as well as how many anomalous events (by using SCA approach) match with traffic alerts from the ground truth data at the same day. The result indicates that different segment sizes do not influence the SCA approach in the process of anomaly detection, and thus we evaluate the traffic flow by applying a 15 Minutes segment. Table II presents the result of the detection of anomalies by applying the SCA approach from 17 to 22 June 2015 as well as describes whether any anomaly detected by SCA matches (MGT) with any traffic alert (GT) which was emitter by traffic agents on the same time period. The percentage of loss candidates is also presented and it describes a high rate of loss. None anomaly detected by SCA approach has matched with the traffic alerts as in the experiment performed on 26 June 2015.

Figure 3 shows the map matching between the detected anomalies by applying SCA and the traffic alerts from traffic agents on 26 June 2015. The magenta dots and lines describe the events which are informed by traffic agents in Dublin and the red dots are the anomalous events detected by checking the spatio-temporal correlation among the sensors (SCA). The percentage of loss candidates by applying the SCA approach is 98.34%, that is, only 1.65% of the candidates are considered as anomalous events by the spatio-temporal correlation among SCATS sensors. Considering the low number of events provided by the ground truth such a drastically reduction might be a good, but another reliable source should be considered in order to check the candidates which are discarded in the process. Figure 4 shows the number of anomalies per hour by applying SCA approach in 3 different days which describes the SCATS sensors correlate more among them at night than in the morning or in the afternoon, that is, low traffic flows make the SCATS traffic sensors be more correlated among them. Therefore, considering all results the use of SCA approach is unfortunately poor for detection

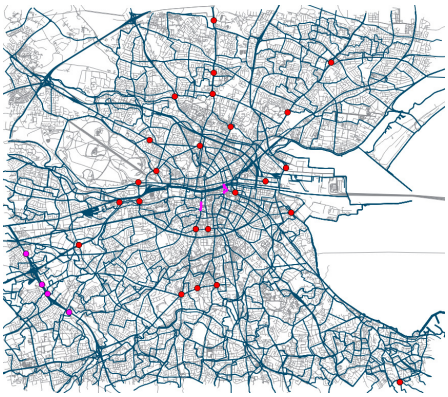


Fig. 3. Comparing ground truth data against the detected events by using SCA approach on 26 June 2015. The magenta dots and lines describe the events which are informed by traffic agents in Dublin and the red dots are the anomalous events detected by using SCA approach.

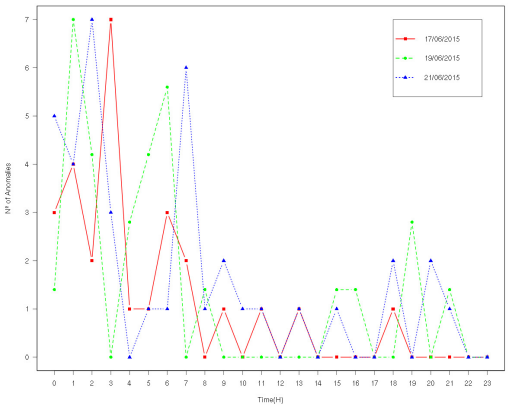


Fig. 4. Number of detected anomalies on 17, 19 and 21 June 2015 by applying SCA approach.

of traffic anomalies.

5. CONCLUSIONS

This work analyzes the spatio-temporal correlation among SCATS sensors in order to find whether such a correlation might support in the process of anomaly detection in an unsupervised manner. However, according to our results the sensors hold a strong correlation at night, but in the morning and in the afternoon such a correlation is weak. We also compare the anomalous events detected (by applying SCA approach) against the traffic alerts which are emitted by traffic agents in Dublin on Twitter. Unfortunately, none of the anomalies have matched with any of the 90 traffic alerts from 17 to 22 June 2015 as well as on 26 June 2015. Therefore, the spatio-temporal correlation among SCATS sensors (SCA approach) is poor for detection of traffic anomalies on static sensors. For future works, we intend to work on an online version of Core Vector Machine (CVM) with uses expert knowledge and traffic alerts to detect anomalies.

Acknowledgements

This research was supported by the National Council for Scientific and Technological Development (CNPq), the European Union's Seventh Framework Programme under grant agreement number FP7-318225, INSIGHT and from the European Union's Horizon 2020 Programme under grant agreement number H2020-ICT-688380, VaVeL. Additionally, this work has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876, project A1.

REFERENCES

Aggarwal, C. *Outlier Analysis*. Vol. 1. Springer, New York, 2013.

Artikis, A., Weidlich, M., Schnitzler, F., Boutsis, I., Liebig, T., Piatkowski, N., Bockermann, C., Morik, K., Kalogeraki, V., Marecek, J., Gal, A., Mannor, S., Gunopulos, D., and Kinane, D. Heterogeneous stream processing and crowdsourcing for urban traffic management. In *Proc. 17th International Conference on Extending Database Technology (EDBT), Athens, Greece, March 24-28, 2014*. OpenProceedings.org, pp. 712–723, 2014.

Bifet, A., Holmes, G., Kirkby, R., and Pfahringer, B. *Data Stream Mining: A Practical Approach*. The university of Waikato, 2011.

Bădoiu, M. and Clarkson, K. L. Optimal core-sets for balls. *DIMACS Workshop on Computational Geometry*, 2002.

Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv.* 41 (3): 15:1–15:58, July, 2009.

Guo, J., Huang, W., and Williams, B. M. Real time traffic flow outlier detection using short-term traffic conditional variance prediction. *Transportation Research Part C: Emerging Technologies*, July, 2014.

Liebig, T., Xu, Z., and May, M. Incorporating mobility patterns in pedestrian quantity estimation and sensor placement. In *Citizen in Sensor Networks*. Springer Berlin Heidelberg, pp. 67–80, 2013.

Liu, S., Chen, L., and Ni, L. M. Anomaly detection from incomplete data. *ACM Trans. Knowl. Discov. Data* 9 (2): 11:1–11:22, Sept., 2014.

Liu, S., Ni, L. M., and Krishnan, R. Fraud detection from taxis' driving behaviors. *IEEE Transactions on Vehicular Technology* 63 (1): 464–472, Jan., 2014.

Liu, W., Zheng, Y., Chawla, S., Yuan, J., and Xing, X. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. ACM, New York, NY, USA, pp. 1010–1018, 2011.

McCann, B. A review of scats operation and deployment in dublin. Tech. rep., ntelligent Transportation Systems, Dublin City Council, Wood Quay, Dublin, 2014.

Pan, B., Zheng, Y., Wilkie, D., and Shahabi, C. Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL'13. ACM, New York, NY, USA, pp. 344–353, 2013.

Pang, L. X., Chawla, S., Liu, W., and Zheng, Y. On detection of emerging anomalous traffic patterns using gps data. *Data Knowl. Eng.* vol. 87, pp. 357–373, Sept., 2013.

Schnitzler, F., Liebig, T., Mannor, S., Souto, G., Bothe, S., and Stange, H. Heterogeneous stream processing for disaster detection and alarming. In *IEEE International Conference on Big Data*. IEEE Press, pp. 914–923, 2014.

Souto, G. and Liebig, T. On event detection from spatial time series for urban traffic applications. In *Solving Large Scale Learning Tasks: Challenges and Algorithms*, S. Michaelis, N. Piatkowski, and M. Stolpe (Eds.). Springer International Publishing, pp. (to appear), 2015.

Stolpe, M., Bhaduri, K., Das, K., and Morik, K. Anomaly detection in vertically partitioned data by distributed core vector machines. *ECML PKDD - Lecture Notes in Computer Science* vol. 8190, pp. 321–336, 2013.

Trilles, S., nd Óscar Belmonte, S. S., and Huerta, J. Real-time anomaly detection from environmental data streams. In *AGILE 2015*, F. Bacao, M. Y. Santos, and M. Painho (Eds.). Lecture Notes in Geoinformation and Cartography. Springer International Publishing, pp. 125–144, 2015.

Yang, S., Kalpakis, K., and Biem, A. Detecting road traffic events by coupling multiple timeseries with a non-parametric bayesian method. *IEEE Transactions on Intelligent Transportation Systems* 15 (5): 1936–1946, March, 2014.

Yang, S. and Liu, W. Anomaly detection on collective moving patterns. *IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing* vol. 7, pp. 291–296, October, 2011.

Análise de Sentimentos baseada em Aspectos usando Aprendizado Semissupervisionado em Redes Heterogêneas

Ivone P. Matsuno[1,2], Rafael G. Rossi[1], Ricardo M. Marcacini[2], Solange O. Rezende[1]

[1] Instituto de Ciências Matemáticas e de Computação (ICMC/USP), Brasil
[2] Universidade Federal de Mato Grosso do Sul (UFMS), Brasil

Abstract. Na Análise de Sentimentos baseada em Aspectos (ASBA) é possível analisar o sentimento de cada aspecto de um produto, por exemplo, a qualidade da câmera, sistema operacional e capacidade de armazenamento de um *Smartphone*. Trabalhos existentes utilizando aprendizado de máquina para ASBA requerem (i) conhecer previamente os possíveis aspectos ou (ii) rotular uma significativa parcela dos dados; o que tornam sua aplicação limitada em cenários reais. Em vista disso, neste trabalho é proposta uma abordagem de aprendizado semissupervisionado que integra diferentes informações em uma única rede tanto para definir se um termo é um aspecto quanto para definir a polaridade dos sentimentos em relação aos aspectos. Os resultados experimentais revelam que a abordagem proposta obtém resultados promissores e competitivos quando comparada com uma abordagem supervisionada.

Categories and Subject Descriptors: G.2.2 [**Graph Theory**]: Graph Labeling; H.2.8 [**Database Applications**]: Data Mining; H.2.4 [**Systems**]: Textual Databases

Keywords: análise de sentimentos, aprendizado de máquina, redes heterogêneas

1. INTRODUÇÃO

Abordagens tradicionais para Análise de Sentimentos (AS) visam classificar, em geral, a polaridade do sentimento de documentos textuais como positiva, negativa ou neutra. Esta classificação é realizada analisando o documento como um todo (AS em nível de documento) ou analisando o sentimento de cada sentença do documento (AS em nível de sentença) [Liu 2012]. Nesses dois casos, não são exploradas informações sobre determinados aspectos de um produto ou serviço analisado, mesmo sendo frequente a existência de sentimentos diferentes para aspectos distintos nos textos analisados [Chen et al. 2014]. Por exemplo, na sentença "*Eu gostei da imagem desta televisão, mas o controle remoto dela é horrível*", temos uma opinião positiva e outra negativa. Além disso, cada opinião refere-se a aspectos diferentes ("imagem" e "controle remoto") da entidade em questão ("televisão") [Kim et al. 2013]. Para lidar com esse tipo de cenário foi proposta a Análise de Sentimentos baseada em Aspectos (ASBA), que potencializa o apoio à tomada de decisão gerando informações mais específicas sobre o sentimento de aspectos de um produto ou serviço. Porém, a ASBA é um processo mais complexo e desafiador [Liu 2012; Jiménez-Zafra et al. 2015].

Os trabalhos mais promissores em ASBA exploram uma combinação de técnicas de Processamento de Linguagem Natural (PLN) e Aprendizado de Máquina (AM) para a extração de aspectos e classificação de sentimentos. Os trabalhos que exploram **aprendizado não supervisionado** geralmente são baseados em Modelagem de Tópicos, como LDA e LSA, para obter o chamado *Topic-Sentiment Model* [Jiménez-Zafra et al. 2015]. Cada tópico é composto por um conjunto de palavras. Uma lista predefinida de palavras positivas e negativas, e uma lista de aspectos são utilizadas para definir a polaridade do sentimento. Como desvantagem, muitos autores afirmam que há uma grande dificuldade em definir um número adequado de tópicos. Ainda, utilizar uma lista predefinida de palavras positivas ou negativas pode não obter resultados satisfatórios para ASBA. Por exemplo, a palavra "*rapidamente*" pode ter um sentimento positivo em "*sistema do celular inicializa rapidamente*", mas negativo em "*bateria do celular descarrega rapidamente*"; o que exige uma análise para cada contexto ou domínio de aplicação. Por fim, há cenários em que uma lista de aspectos não está disponível.

Já os trabalhos que exploram **aprendizado supervisionado** utilizam PLN para extrair atributos gramaticais dos textos e a classificação de aspectos é baseada nesses atributos [Liu 2012; Chen et al. 2014]. Por exemplo, aspectos geralmente são representados por substantivos conectados a advérbios e adjetivos. Assim, os aspectos são rotulados pelos usuários para, posteriormente, aprender um classificador que identifique aspectos em novos textos. De forma análoga, os usuários também rotulam exemplos positivos e negativos que envolvam os aspectos, também possibilitando aprender um classificador que defina o sentimento de aspectos não vistos. Embora obtenham maior acurácia, tais trabalhos exigem um grande esforço humano devido a necessidade de rotular um grande conjunto de textos [Chen et al. 2014; Pontiki et al. 2014].

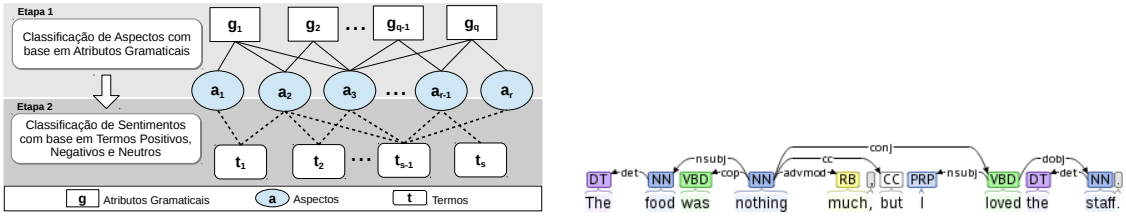
Em vista disso, neste trabalho é proposta uma abordagem para ASBA, denominada ASPHN (*Aspect-Based Sentiment Propagation on Heterogeneous Networks*), que utiliza aprendizado semissupervisionado tanto para classificação de aspectos, quanto para classificação de sentimento dos aspectos. O objetivo é utilizar apenas um pequeno conjunto de exemplos previamente rotulados e realizar o aprendizado por meio de propagação de rótulos em redes heterogêneas [Sun and Han 2012]. Na abordagem ASPHN, são integrados diversos tipos de informações como vértices em uma única rede: atributos linguísticos; candidatos a aspectos e um conjunto de termos. O aprendizado é baseado na propagação da informação de exemplos rotulados por meio das relações topológicas entre os vértices. A abordagem ASPHN foi avaliada experimentalmente e obteve resultados promissores e competitivos quando comparada com uma abordagem supervisionada.

2. ABORDAGEM PROPOSTA: ASPHN (*ASPECT-BASED SENTIMENT PROPAGATION FOR HETEROGENEOUS NETWORKS*)

Dado um conjunto de documentos escritos em língua natural, representando mensagens, revisões, análises, ou notícias sobre assunto específico, o problema da ASBA pode ser definido como extrair opiniões representadas pela tripla $O = (e_i, a_{ij}, s_{ij})$, em que e_i é uma entidade i (e.g. produto ou serviço), a_{ij} é o aspecto j (propriedade) da entidade i, e s_{ij} é a polaridade do sentimento em relação ao aspecto a_{ij} da entidade e_i, por exemplo, positiva, negativa ou neutra. No escopo deste trabalho, a entidade é uma informação prévia do problema, de forma que o objetivo do aprendizado é a identificação de aspectos e classificação dos respectivos sentimentos referentes à entidade definida.

A abordagem proposta denominada ASPHN (*Aspect-Based Sentiment Propagation for Heterogeneous Networks*) utiliza uma rede heterogênea composta por (i) aspectos $\mathcal{A} = \{a_1, ..., a_r\}$; (ii) atributos gramaticais $\mathcal{G} = \{g_1, ..., g_q\}$ conectados aos candidatos à aspectos; e por (iii) termos $\mathcal{T} = \{t_1, ..., t_g\}$ conectados aos aspectos classificados, conforme ilustrado na Figura 1(a). As arestas indicam ausência ou presença das relações, especificamente, (i) quando atributo gramatical está relacionado ao aspecto e (ii) quando um termo ocorre na mesma sentença de um aspecto. Na modelagem aqui proposta, os candidatos à aspectos são termos compostos por substantivos, verbos, adjetivos e advérbios identificados na coleção textual. Já os atributos gramaticais são extraídos das sentenças, bem como da respectiva rede de dependência sintática. Por exemplo, na frase em inglês "*The food was nothing much, but I loved the staff.*" é possível obter a estrutura ilustrada na Figura 1(b), com uso da ferramenta *Stanford CoreNLP*. Por fim, os termos são palavras extraídas dos textos, eliminando-se as *stopwords* (preposições, artigos, conjunções e pronomes).

Após a modelagem da rede heterogênea, o processo avança para a etapa de aprendizado semissupervisionado. Nesse caso, assume-se que uma pequena quantidade de aspectos já foi rotulada. O aprendizado na abordagem ASPHN envolve a propagação desses rótulos, conforme a topologia da rede, baseado em uma extensão aqui proposta do método *Label Propagation using Bipartite Heterogeneous Networks* [Rossi et al. 2014], que é um dos algoritmos do estado da arte para esta tarefa. Para tal, considere que $\mathcal{A} = \mathcal{A}^L \cup \mathcal{A}^U$ é o conjunto de vértices do tipo aspecto, na qual $\mathcal{A}^L$ é o conjunto de aspectos rotulados e $\mathcal{A}^U$ é o conjunto de aspectos não rotulados, e que um conjunto $\mathcal{X}$ refere-se a um segundo tipo de vértice. Ainda, considere que a probabilidade do vértice a_i estar conectado ao vértice



(a) (b)

Fig. 1. (a) Esquema conceitual da rede heterogênea proposta na abordagem ASPHN; (b) Exemplo de atributos linguístico que podem ser extraídos por Processamento de Linguagem Natural.

x_j é calculada pela Eq. 1, em que $w(a_i, x_j)$ retorna 1 quando existe uma aresta conectando os dois vértices, ou 0 caso contrário. De forma recíproca, a probabilidade do vértice x_j estar conectado ao vértice a_i é calculada pela Eq. 2.

$$p(a_i, x_j) = w(a_i, x_j) / \sum_{a_k \in \mathcal{A}} w(a_k, x_i) \qquad (1)$$

$$p(x_j, a_i) = w(a_i, x_j) / \sum_{x_k \in \mathcal{X}} w(a_i, x_k) \qquad (2)$$

$$\begin{bmatrix} \mathbf{F}_{A^L} \\ \mathbf{F}_{A^U} \\ \mathbf{F}_X \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{A^L A^L} & \mathbf{P}_{A^L A^U} & \mathbf{P}_{A^L X} \\ \mathbf{P}_{A^U A^L} & \mathbf{P}_{A^L A^L} & \mathbf{P}_{A^U X} \\ \mathbf{P}_{X A^L} & \mathbf{P}_{X A^U} & \mathbf{P}_{XX} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{A^L} \\ \mathbf{F}_{A^U} \\ \mathbf{F}_X \end{bmatrix} \qquad (3)$$

O problema da propagação de rótulos é modelado pela Eq. 3, em que $\mathbf{F}_{A^L}$ é uma matriz com aspectos rotulados pelo usuário, $\mathbf{F}_{A^U}$ é uma matriz que irá armazenar o peso de cada aspecto não rotulado para cada possível possível rótulo, e $\mathbf{F}_X$ é uma matriz que armazena o peso da contribuição dos vértices de $\mathcal{X}$ para cada possível rótulo. Cada aspecto rotulado na matriz $\mathbf{F}_{A^L}$ recebe o valor 1 na posição correspondente ao rótulo e 0 nas demais posições. Já os valores das matrizes $\mathbf{F}_{A^U}$ e $\mathbf{F}_X$ são inicializados com 0. Também é utilizada uma matriz $\mathbf{Y}$ durante o processo de propagação de rótulos, que é inicializada da mesma forma que a matriz $\mathbf{F}_{A^L}$. As matrizes $\mathbf{P}$ representam as probabilidades de conexão entre vértices por meio das Eq. 1 e 2, por exemplo, $\mathbf{P}_{X A^L}$ indica a probabilidade das conexões entre vértices de $\mathcal{X}$ e aspectos rotulados. Quando não há conexões entre vértices do mesmo tipo, as matrizes são zeradas, como ocorre em $\mathbf{P}_{A^L A^L}$, $\mathbf{P}_{A^L A^U}$, $\mathbf{P}_{A^U A^L}$, $\mathbf{P}_{A^L A^L}$ e $\mathbf{P}_{XX}$. Considerando que $\mathbf{Y}$ é o conjunto original de aspectos rotulados e que $\mathbf{F}_{A^U}$ é inicializada com zero, os passos abaixo são utilizados para resolução da Eq. 3 de forma iterativa:

(1) Propagar rótulos de Aspectos para definir contribuição de $\mathcal{X}$: $\mathbf{F}_X \leftarrow \mathbf{P}_{X A^L}\, \mathbf{F}_{A^L} + \mathbf{P}_{X A^U}\, \mathbf{F}_{A^U}$.
(2) Propagar contribuição de $\mathcal{X}$ para os Aspectos: $\mathbf{F}_{A^U} \leftarrow \mathbf{P}_{A^U X}\, \mathbf{F}_X$ e $\mathbf{F}_{A^L} \leftarrow \mathbf{P}_{A^L X}\, \mathbf{F}_X$.
(3) Manter conjunto original de aspectos rotulados: $\mathbf{F}_{A^L} \leftarrow \mathbf{Y}^L$.
(4) Repetir passos 1, 2, e 3 até convergência.

Na abordagem ASPHN, a primeira etapa é utilizada para classificação de aspectos, em que os vértices $\mathcal{X}$ são atributos gramaticais e os aspectos são rotulados como "*sim*" ou "*não*". Após convergência da primeira etapa, os aspectos classificados como "*sim*" são utilizados na segunda etapa, em que os rótulos dos aspectos a serem propagados são "*positivo*", "*negativo*" ou "*neutro*"; e os vértices $\mathcal{X}$ são os termos. A ideia desta segunda etapa é que, se um termo está conectado com aspectos rotulados, então esta informação é propagada aos aspectos não rotulados conectados a este termo. Após a convergência da segunda etapa, a rede heterogênea representa tanto os aspectos classificados quanto a polaridade do sentimento de cada aspecto, obtido a partir de uma pequena amostra de dados rotulados.

3. AVALIAÇÃO EXPERIMENTAL

A avaliação da abordagem ASPHN foi baseada em dois conjuntos de dados, com a polaridade de cada aspecto anotada por humanos. O primeiro é composto por 3.044 revisões em inglês sobre restaurantes

e o segundo por 3.048 revisões em inglês sobre laptops, disponibilizados em [Pontiki et al. 2014]. A extração de atributos gramaticais foi realizado por meio da ferramenta *Stanford CoreNLP*, gerando 74 atributos compostos por estruturas gramaticais e dependências sintáticas.

A avaliação é baseada na taxa de acerto média considerando validação cruzada usando 10-*folds*. O processo de aprendizado semissupervisionado da abordagem ASPHN foi simulado com quatro tamanho de amostras rotuladas, selecionando-se aleatoriamente {1%, 10%, 20% *e* 30%} do conjunto de treinamento em cada iteração da validação cruzada. Os demais exemplos foram considerados como exemplos não rotulados por parte da abordagem ASPHN. A abordagem ASPHN foi comparada com uma abordagem supervisionada, que utiliza 90% de dados rotulados em cada iteração da validação cruzada. Foi selecionado o algoritmo *Naive Bayes* para a comparação, uma vez que este permite verificar a contribuição (probabilidade) de cada atributo para cada classe (similar à abordagem ASPHN).

	Abordagem Semissupervisionada (ASPHN) Exemplos Rotulados (%)				Abordagem Supervisionada
Etapa 1 – Classificação de Aspectos	**1**	**10**	**20**	**30**	-
Restaurantes	73.62	76.84	77.06	77.33	77.94
Laptops	72.13	73.35	73.98	74.71	75.40
Etapa 2 – Classificação de Sentimentos	**1**	**10**	**20**	**30**	-
Restaurantes	59.93	59.94	59.96	60.03	59.20
Laptops	42.83	49.66	54.67	56.86	59.99

Fig. 2. Comparação da taxa de acerto entre a ASPHN com uma abordagem supervisionada.

Os resultados experimentais, apresentados no quadro da Figura 2 indicam que a abordagem proposta ASPHN, mesmo utilizando uma quantidade muito inferior de exemplos rotulados, obtém resultados competitivos com uma abordagem supervisionada. Uma análise estatística utilizando o teste *T-Student* não indica diferença significativa entre as duas abordagens quando são utilizados 10% ou mais de dados rotulados na abordagem ASPHN.

4. CONSIDERAÇÕES FINAIS

Neste trabalho foi proposta a abordagem ASPHN (*Aspect-Based Sentiment Propagation for Heterogeneous Networks*), que permite identificar aspectos e classificar a polaridade do sentimento de cada aspecto usando aprendizado semissupervisionado. Não há na literatura uma abordagem que explora redes heterogêneas para análise de sentimentos baseada em aspectos e os resultados preliminares obtidos indicam que este é um caminho promissor.

Entre as limitações e direções para trabalhos futuros, a abordagem deve ser comparada com outras estratégias de aprendizado semissupervisionado, bem como em textos escritos na língua portuguesa. Ainda, os autores pretendem integrar uma estratégia de aprendizado ativo para apoiar a rotulação da amostra inicial, com avaliação em outros domínios e conjuntos de dados.

Agradecimentos: Os autores agradecem à FAPESP (Processos 2014/08996-0 e 2011/12823-6) e PROPP/UFMS (Protocolo SigProj 160343.669.169765.12112013) pelo auxílio fornecido para desenvolvimento deste trabalho.

REFERENCES

Chen, Z., Mukherjee, A., and Liu, B. Aspect extraction with automated prior knowledge learning. In *Proc. of ACL*. pp. 347–358, 2014.

Jiménez-Zafra, S. M., Martín-Valdivia, M. T., Martínez-Cámara, E., and Ureña-López, L. A. Combining resources to improve unsupervised sentiment analysis at aspect-level. *Journal of Information Science*, 2015.

Kim, S., Zhang, J., Chen, Z., Oh, A. H., and Liu, S. A hierarchical aspect-sentiment model for online reviews. In *AAAI*, 2013.

Liu, B. Sentiment analysis and opinion mining. *Morgan & Claypool Publishers*, 2012.

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proc. SemEval*. pp. 27–35, 2014.

Rossi, R. G., Lopes, A. A., and Rezende, S. O. A parameter-free label propagation algorithm using bipartite heterogeneous networks for text classification. In *Proc. Symposium on Applied Computing*. ACM, pp. 79–84, 2014.

Sun, Y. and Han, J. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers, 2012.

Mineração de Preferências do Usuário em Textos de Redes Sociais usando Sentenças Comparativas

Fabiola S. F. Pereira and Sandra de Amo

Universidade Federal de Uberlândia, Brazil
fabfernandes@comp.ufu.br deamo@ufu.br

Resumo. Opiniões comparativas representam uma das maneiras mais genuínas dos usuários expressarem suas preferências sobre duas ou mais entidades. Com o advento das redes sociais, é cada vez mais comum encontrar postagens nas quais os usuários exprimem suas opiniões através de comparações para seus amigos. Essas comparações são tanto entre dois produtos, entre diferentes ambientes e até mesmo entre duas ou mais pessoas. Neste artigo é endereçado o problema de mineração de preferências dos usuários a partir do conteúdo das redes sociais. Primeiro, um modelo de representação de preferências dos usuários utilizando sentenças comparativas postadas em redes sociais é proposto. Depois, com base nesse modelo, um framework é estruturado, consolidando as etapas do processo de transformação de texto em preferências. Experimentos preliminares indicam o potencial da abordagem proposta.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.2.7 [**Natural Language Processing**]: Text Analysis

Keywords: comparative sentences, opinion mining, preference mining, social media mining

1. INTRODUÇÃO

Com o grande volume de dados disponível atualmente, explorar a maneira como as preferências do usuário são obtidas é um tópico de pesquisa que tem recebido cada vez mais atenção [Amo et al. 2015], [de Amo and Oliveira 2014]. Um usuário pode expressar seus gostos explicitamente, através de sistemas especialistas que fazem perguntas específicas do tipo: dadas essas duas obras de arte, qual é sua preferida? Ou ainda, qual a nota que você avalia esse filme? Ou, as preferências podem ser capturadas implicitamente, aplicando algoritmos de mineração de preferências que investigam opiniões e escolhas passadas, sem exigir qualquer esforço específico por parte do usuário, o que torna essa segunda abordagem mais interessante e desafiadora [Amo et al. 2015].

A possibilidade de cada vez mais as pessoas se expressarem e interagirem livremente nas redes sociais, faz desses sistemas uma fonte rica de informações sobre as opiniões e comportamentos de seus usuários. Além do conteúdo textual publicado, é possível extrair interações e relacionamentos. Considerando a hipótese de que quanto mais informação embutida nos dados de preferência, mais eficiente será o modelo produzido, torna-se interessante utilizar redes sociais como fonte de captura implícita de preferências.

Ao observar a estrutura de textos que expressam opiniões, sentenças comparativas são boas fontes para obtenção de pares de preferências do usuário. De acordo com [Jindal and Liu 2006a], sentenças comparativas são frases que expressam uma relação entre dois ou mais itens baseada em suas similaridades ou diferenças. Por exemplo, quando um usuário u publica uma mensagem do tipo *"prefiro jogos XBox do que PS4"*, claramente pode-se identificar a preferência de u por jogos XBox sobre o PS4.

Assim, este artigo tem como objetivo explorar a ideia de que é possível inferir preferências a partir de textos em linguagem natural utilizando algoritmos de mineração. Para tanto, é proposto o framework

Os autores agradecem as agências brasileiras CAPES, FAPEMIG e CNPq pelo financiamento deste trabalho.

TextPref para mineração de preferências do usuário a partir de opiniões comparativas em redes sociais. O TextPref possui quatro módulos e combina técnicas de mineração de opiniões com informações de redes sociais para transformar sentenças comparativas em um conjunto de pares de itens do tipo (i_1, i_2), significando que i_1 é preferido a i_2 (*preferências pairwise*). Além da relação de preferência entre itens, uma preferência minerada pelo framework contém informações adicionais da popularidade daquela preferência na rede social e a intensidade com que o usuário a expressou.

Este artigo está organizado da seguinte maneira: na Seção 2 são destacados os trabalhos correlatos a partir dos quais as contribuições deste artigo foram baseadas. A Seção 3 descreve o modelo de preferências sociais utilizado, bem como o framework TextPref. A Seção 4 descreve experimentos preliminares que justificam o uso de sentenças comparativas na tarefa de mineração de preferências em redes sociais. Por fim, a Seção 5 conclui o artigo e aponta direções futuras desta pesquisa em andamento.

2. TRABALHOS CORRELATOS

De acordo com [Jindal and Liu 2006a], opinião comparativa é aquela que compara entidades baseada em alguns aspectos em comum entre elas. Formalmente, pode ser representada por uma sêxtupla (E_1, E_2, A, PE, h, t), onde E_1 e E_2 são conjuntos de entidades sendo comparados, A é o conjunto de aspectos em questão, $PE \in \{E_1, E_2\}$ é o conjunto de entidades preferidas, h é o emissor da opinião e t o momento em que ela foi emitida. Uma sentença comparativa é uma sentença que contém uma opinião comparativa. Por exemplo, considere a seguinte sentença comparativa: *"@maria jogos do wii u possuem gráficos melhores do que jogos do xbox1 e ps4"*, publicada pelo usuário João em 12/06/2015. A opinião comparativa extraída dessa sentença é: *({jogos wii U},{jogos ps4, jogos xbox1},{gráficos},{jogos wii U}, João, 12/06/2015).*

Os trabalhos mais representativos nessa área são [Jindal and Liu 2006a], [Jindal and Liu 2006b] e [Ganapathibhotla and Liu 2008]. São tratadas as tarefas de mineração de sentenças comparativas, identificação de elementos da comparação e identificação da entidade preferida, respectivamente. Os primeiros módulos do framework TextPref proposto neste trabalho baseiam-se nessa sequência de trabalhos correlatos sobre o tópico de opiniões comparativas.

3. MINERAÇÃO DE PREFERÊNCIAS EM TEXTOS DE REDES SOCIAIS

Antes de apresentar o framework proposto neste artigo, primeiro é necessário apresentar o modelo das preferências que serão mineradas. É basicamente um modelo que absorve tanto informações textuais, como por exemplo a intensidade de uma preferência, quanto informações sociais, como por exemplo a popularidade de uma opinião. Em seguida, o framework TextPref é apresentado.

3.1 Modelo de preferências sociais

Tomando como base o modelo proposto por [Jindal and Liu 2006a] e discutido na Seção 2, neste artigo é proposta uma extensão desse modelo considerando duas novas variáveis: *grau de preferência* (λ) e *grau social* (φ). Essa extensão permite enriquecer com mais informações as preferências mineradas das redes sociais.

O *grau de preferência* [Costa and de Amo 2014] refere-se à intensidade de uma preferência entre dois conjuntos de objetos a partir da maneira como o usuário escreve sua comparação. Por exemplo, se um usuário u_1 possui duas postagens p_1 = *"prefiro muito mais o XBox ao PS4"* e p_2 = *"o XBox é melhor que o Wii"*, é possível inferir que a preferência dele pelo XBox em relação ao PS4 é maior do que sua preferência pelo XBox em relação ao Wii. Formalmente, sejam E_1 e E_2 dois conjuntos de entidades sendo comparados pelo usuário u. $\lambda_{E_1E_2} \in [0, 1]$ representa o grau de preferência de u pelas entidades de E_1 em relação às entidades de E_2. Quanto maior $\lambda_{E_1E_2}$, mais intensa é a preferência. Uma estratégia para quantificação de $\lambda_{E_1E_2}$ é montar um *ranking* a partir de um dicionário de expressões linguísticas que representam comparações.

O *grau social* é responsável por agregar informações sociais à preferência do usuário. A proposta é agregar a uma preferência um grau que represente o quanto o texto em que ela foi expressa teve impacto na rede social. Uma estratégia para obtenção do grau social é, por exemplo, medi-lo em função da interação gerada por uma opinião publicada. Considerando os conceitos da rede social Twitter: *favorito* (uma postagem pode ser assinalada como favorita por diferentes usuários), *menção* (um texto pode conter menções a outros usuários) e *retweet* (quando uma postagem é replicada por outros usuários), o grau social φ_p de uma postagem p pode ser definido como em [rec 2014]: $\varphi_p = f_p + rt_p + m_p$, onde $f_p, rt_p, m_p \in \mathbb{N}$ e f_p é o número de vezes que p foi marcado como favorito, rt_p é o número de vezes que p foi reproduzida (*retweets*) e m_p é a quantidade de menções em p.

Assim, uma preferência publicada em uma rede social será representada através de uma óctupla do tipo: $(E_1, E_2, A, PE, h, t, \lambda, \varphi)$. Como exemplo, suponha a seguinte postagem do usuário *John* no Twitter, no dia 12/12/2014: *"@cris os gráficos do XBox são muito melhores do que no Wii."* Considerando que $\lambda = 0.9$ para o termo *muito melhor* e que esse texto foi marcado como favorito por 1 usuário, contém 1 menção e foi compartilhado 7 vezes, tem-se a seguinte preferência minerada: *({XBox}, {Wii}, {gráficos}, {XBox}, John, 12/12/2014, 0.9, 9).*

3.2 O framework TextPref

O framework TextPref é uma sequência de passos que devem ser seguidos para atingir o objetivo de transformar texto em preferências, composto por quatro módulos. A Figura 1 é uma visão geral dos módulos do TextPref. TextPref é definido como um framework pois cada etapa tem uma entrada e uma saída bem definidas através do modelo de preferências proposto. A motivação para estruturar essa sequência de passos em um framework é a oportunidade observada de desenvolver novos algoritmos dentro de cada etapa.

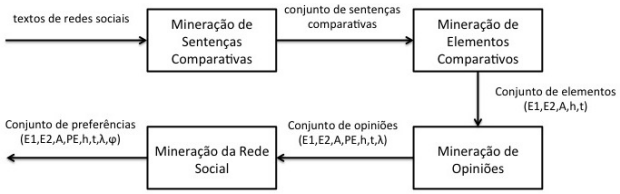


Fig. 1. Framework TextPref

Mineração de Sentenças Comparativas. O objetivo desse módulo é identificar, a partir de um corpus, quais sentenças são comparativas de acordo com a definição de [Jindal and Liu 2006a] (Seção 2). Como parte deste trabalho de pesquisa, em [Pereira 2015], um algoritmo genético foi proposto para realização dessa tarefa. **Mineração de Elementos Comparativos.** A partir de um conjunto de sentenças comparativas, é necessário identificar quais são as entidades (E_1 e E_2) e aspectos (A) envolvidos na comparação de cada sentença. Além disso, informações de data (t) e emissor (h) da opinião devem ser mapeadas. A estratégia de [Jindal and Liu 2006b] pode ser utilizada nesse módulo. **Mineração de Opiniões.** Nessa etapa, o objetivo é encontrar o conjunto de entidades preferidas PE (o trabalho [Ganapathibhotla and Liu 2008] endereça esse problema), bem como o grau de cada preferência λ (questão em aberto na literatura). **Mineração da Rede Social.** É neste módulo que o *grau social* φ é obtido. É cada vez maior a tendência de utilizar a popularidade de um usuário ou postagem (*user/tweet engagement*) como métrica em sistemas de recomendação [rec 2014]. Além da popularidade, é possível desenvolver modelos e algoritmos que utilizam outros tipos de informações sociais para enriquecer a informação de preferência. Por exemplo, captura da estrutura da rede.

Por fim, a saída do framework será um conjunto de preferências enriquecidas com informações textuais e sociais que podem servir como entrada para algoritmos que constroem modelos de recomendação a partir de amostras de preferências do usuário. Um exemplo é o trabalho [de Amo and Oliveira 2014].

4. EXPERIMENTO INICIAL

O objetivo do experimento realizado neste artigo é mostrar que é possível obter sentenças comparativas com alto nível de interação social a partir de publicações em redes sociais. Foi construída uma grande base de dados a partir do Twitter (TW-Full) contendo postagens sobre os *consoles* PlayStation, XBox e Wii em língua inglesa. Para identificação de opiniões, como parte desta pesquisa, em [Pereira 2015] foi proposto um algoritmo genético para mineração de sentenças comparativas, cuja acurácia atingiu 73% sobre uma pequena amostra dessa base construída (TW-Sample), superando as abordagens do estado-da-arte. Esta proposta refere-se ao primeiro módulo do framework TextPref. Agora, neste experimento, foi considerado o modelo obtido em [Pereira 2015] para a mineração de toda a base TW-Full coletada (˜5 milhões de tweets), configurando uma quantidade expressiva de textos[1]. A Tabela I sintetiza os valores obtidos. Vale ressaltar que, como os textos do Twitter são textos curtos, um *tweet* foi tratado como uma sentença. Com esse experimento, foi possível detectar que redes sociais são fontes promissoras de sentenças comparativas e que opiniões geram um alto nível de interação social.

	TW-Sample	TW-Full
# sentenças	1500	4970000
# sentenças comparativas	199 (13.26%)	815080 (16.4%)
período	Dez 2014	Dez 2014 - Jun 2015
média de menções/sentença comparativa	1	1.1
média de retweets/sentença comparativa	1	4
média de favoritos/sentença comparativa	4	6

Table I. Estatísticas da base de dados coletada do Twitter

5. CONSIDERAÇÕES FINAIS E DIREÇÕES FUTURAS

Neste artigo foi proposta a utilização de sentenças comparativas para mineração de preferências em redes sociais. Primeiro, um modelo de preferências sociais foi definido e, em seguida, o framework apresentado TextPref consolida as etapas de um processo de transformação de texto em preferências do usuário. Experimentos preliminares mostraram que redes sociais são fontes promissoras de sentenças comparativas para mineração de preferências.

Muito trabalho ainda precisa ser feito. Esta pesquisa segue na direção de explorar cada módulo do framework TextPref, propondo algoritmos eficientes para as tarefas de mineração. Em específico, no módulo de Mineração da Rede Social, o objetivo será combinar informações estruturais da rede com algoritmos de influência social para predição de preferências.

REFERÊNCIAS

RecSysChallenge '14: Proceedings of the 2014 Recommender Systems Challenge. ACM, 2014.

Amo, S. d., Diallo, M. S., Diop, C. T., Giacometti, A., Li, D., and Soulet, A. Contextual preference mining for user profile construction. *Inf. Syst.* 49 (C): 182–199, 2015.

Costa, J. R. and de Amo, S. Improving pairwise preference mining algorithms using preference degrees. In *29th Brazilian Symposium on Databases*. pp. 107–116, 2014.

de Amo, S. and Oliveira, C. Towards a tunable framework for recommendation systems based on pairwise preference mining algorithms. In *Advances in Artificial Intelligence*. Vol. 8436. pp. 282–288, 2014.

Ganapathibhotla, M. and Liu, B. Mining opinions in comparative sentences. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*. COLING '08. pp. 241–248, 2008.

Jindal, N. and Liu, B. Identifying comparative sentences in text documents. In *29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '06. pp. 244–251, 2006a.

Jindal, N. and Liu, B. Mining comparative sentences and relations. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*. AAAI'06. pp. 1331–1336, 2006b.

Pereira, F. S. F. Mining comparative sentences from social media text. In *2nd Data Mining and Natural Language Processing (DMNLP) Workshop*. pp. (to appear), 2015.

[1]Base de dados e gráficos com estatísticas disponíveis em `http://lsi.facom.ufu.br/~fabiola/comparative-mining`