A Tool for Classification of Sequential Data

Giacomo Kahn[1], Yannick Loiseau[1,2], and Olivier Raynaud[1,2]

[1] Université Clermont Auvergne, Université Blaise Pascal, BP 10448, F-63000 CLERMONT-FERRAND, FRANCE
[2] CNRS, UMR 6158, LIMOS, F-63178 AUBIERE, FRANCE
giacomo.kahn@isima.fr, home page: http://fc.isima.fr/~kahngi/

Abstract. Classification of sequential data (data obtained from series of actions in chronological order) has many applications in security, marketing or ergonomy. In this paper, we present a tool for classification of sequential data. We introduce a new clean dataset of web-browsing logs, and study the case of implicit authentification from web-browsing. We then detail more of the functioning of the tool and some of its parameters.

Keywords: Machine Learning, Classification, Web Usage Mining

1 Introduction and related work

Event-related data can have the form of a succession of actions or events in chronological order. Data mining of such data has many applications in fields such as security (intrusion detection [1]), marketing (e.g. navigation in e-commerce hierarchy) or ergonomy (study of succession of actions in work-related applications). Those applications require the search of some meaningful patterns in the data. A pattern is a structure that appears with regularity in the data. It can be an itemset, a sequence, a sub-word, an association rule... In this context, meaningful means maximizing some metric, such as the *support* or the *lift*. Different algorithms exist to mine either of those. An interesting property for patterns is the closure. A pattern p is closed if there is no pattern p', superset of p and $support(p) = support(p')$. Formal Concept Analysis (FCA) is a mathematical framework that deals with closed sets. Many algorithms from FCA allow to enumerate closed sets (in the form of concepts) and there exist a number of interesting metrics based on concept lattices such as stability or robustness of a concept.

The enumeration of these patterns alone is not sufficient in many cases and is only one step of a decision-making process. For example, in a context of security, one might want to find meaningful patterns as the first step of classification or prediction. In marketing, one might use patterns to construct groups of consumers or to find interesting association rules.

In [2,3], the authors introduce a tool for classification in the binary case, based on positive and negative examples in concept lattices. However, by using this binary classifier to the $1-n$ case (n being the number of classes), all anonymous

behaviours will be classified as contradictory. Other works of mining in FCA include the mining of sequences in [4] and of graphs in [5]. In [6], the authors defined *emerging patterns* as patterns appearing frequently in a class, but being hard to find in other classes. Confer [7, 8] for surveys on emerging patterns. An emerging closed-pattern classifier can be described as an extension to the $1 - n$ case of the binary concept lattice classifier, and can be used to predict the class on previously unseen objects. In [9], the authors present another generalisation to n classes of the closed-set based classifier. In particular, the authors introduced the use of the $tf \times idf$ for the selection of the closed patterns.

In this paper, we present a tool for classification of sequential data, based on closed-patterns. This tool implements the classifier presented in [9]. We show some results of our tool on a dataset of web navigation logs from more than 3000 users over a six-month period.

This paper is organised as follow: in section 2 we explain the functioning of the classifier and give more details about the tool and its parameters, in section 3 we show a case study and propose a clean dataset for experimentation, finally we conclude and give some perspectives of our work.

2 Implementation

2.1 General parameters

In this section, we describe the classifier implemented by our tool. The tool includes a whole experimental process, from the building of transactions from raw data to detailed results of classification. We mention some of the parameters accepted by each steps.

Building transactions Our tool allows us to group the data into transactions. The transactions can be of fixed size, or created with respect to a time stamp present in the original data. In our case study, the size is fixed and is equal to 10. The data file from where the transactions are built can be of arbitrary size.

Extraction of own patterns We call own patterns the patterns we believe to be respresentative of each class. For each class, we compute the patterns that verify some property or threshold for a given metric (e.g. support or $tf \times idf$). With some metrics, the space of those patterns is prunable. The number of patterns we want to keep as well as their maximum size is a parameter. The nature of the pattern is also a parameter: as of today, one can choose between closed itemset or sequence. For a given class c, we denote the set of own patterns by P_c.

Profile of a class There exist different ways to compute the profile of a class. In our tool, we chose to define a common vector profile $\mathcal{V} = \bigcup_{c \in C} P_c$ that is the union of all own patterns for all classes. We then compute its numerical components for each classes from either the *support*, the *lift* or the $tf \times idf$. This vector allows us to embed all classes in a common space. This numerical value can be seen as the distance from the origin of the space, in each dimensions of

the vector. For exemple, let α and β two classes. $P_\alpha = \{A, B, C\}$ and $P_\beta = \{C, D, E\}$ then the vector $\mathcal{V} = P_\alpha \cup P_\beta$ will have 5 component (A, B, C, D, E). For each class c, we compute a numerical value k_i^c for each component, giving $\mathcal{V}_\alpha = (k_A^\alpha, k_B^\alpha, k_C^\alpha, 0, 0)$ and $\mathcal{V}_\beta = (0, 0, k_C^\beta, k_D^\beta, k_E^\beta)$.

Profile of an anonymous transaction This step accepts the same parameters as the construction of the profile of a class. We can also choose the number of anonymous transaction that will be submitted to the classifier in the next step. For example, in Fig. 3, the number of anonymous transactions recieved by the classifier goes from 1 to 30.

Identification step The goal is to guess the class corresponding to an anonymous set of transactions. After the computation of a profile for this anonymous set, we compute the nearest neighbor in the common space defined previously. The tool implements different similarity functions: *euclidean* distance, *cosine* similarity, *Kulczynski* measure, and *Dice* similarity. The heuristics gain in efficiency when they are provided with a higher number of anonymous transactions that allows them to construct a finer profile for the anonymous user.

Global parameters Other parameters for experimentations include the number of runs, the verbosity level, the format of the data, the possibility to only compute stats on the data, the use of a fuzzy approach and some parameters for binary classification.

Bayesian Classifier Our tool implements two smoothed Bayesian classifiers: a traditional Bayes classifier and a pattern-based Bayes classifier. Those classifiers allow to compare the results during the experimentations.

2.2 Fuzzy approach

The inclusion of a pattern in a transaction is a binary measure. When working with own patterns of significant size, this strict inclusion will often be false. We consider a fuzzy approach for the support during the computation step of an anonymous profile. We will use a inclusion level instead of a binary measure. The fuzzy support may then be computed as the average of the inclusion levels on the set of transitions.

The fuzzy inclusion level $inc(P, T)$ can be computed as the proportion of the own pattern P included in the transaction T:

$$inc(P, T) = \frac{||P \cap T||}{||P||} \tag{1}$$

To adjust to different cases and be able to represent a wide range of inclusion, from intersection to strict inclusion, we use a transfer function to transform the simple level of inclusion of eq. 1. In the tool, those functions are defined by specifying points on a 2-dimensional space. Two points are fixed, $(0, 0)$ and $(1, 1)$. Some transfer functions are illustrated in Fig. 1.

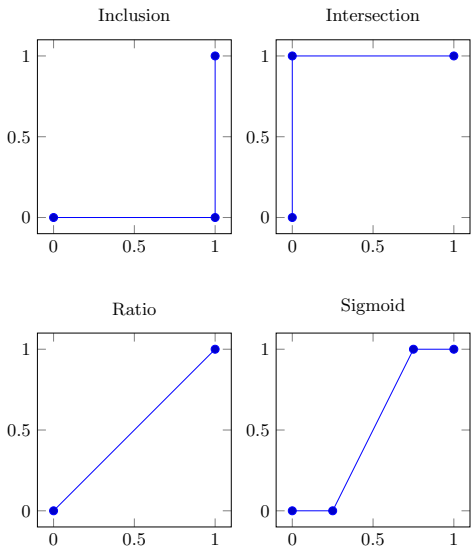

Fig. 1. Different transfer functions

The coordinates of the two points that define the transfer function are configuration parameters. In Fig. 1, the parameters for the inclusion are $[(1,0);(1,0)]$. This is equivalent to the binary measure of inclusion. For the intersection, the parameters are $[(0,1);(0,1)]$. Those parameters mean the measure is equal to one as soon as the intersection is not empty. For the simple ratio or a more sigmoidal function, the parameters are resp. $[(0,0);(1,1)]$ and $[(0.25,0);(0.75,1)]$.

2.3 Configuration file

The parameters are given to the tool by a configuration file in *.yml* format. For the results of our case study, presented in Table 2, the file is presented in Fig. 2.

With this file as argument, the tool will recieve from 1 to 30 anonymous transactions, and run 10 executions. The random seed can be fixed to reproduce experimentations. The data comes from the directory `Data/150users`, and is in *csv* format. The transactions are built of fixed size 10. The identification method is H_1 (closed itemsets and $tf \times idf$ metric), with at most 40 own closed-patterns of maximum size 5. The similarity measure used is Kulczynski. The profiler is the metric used to compute the numerical coordinate of the common vector. When not specified, the method used for inclusion of the pattern is the strict inclusion.

3 Case study

Our case study is about implicit identification in web-browsing. Implicit identification is studied in [10] and in a web-browsing context in [11–14]. The challenge

```
---
name: H1 on csv data
verbose-level: WARNING
number-of-categories: [150]
anonymous-transactions-sizes: [1, 2, 5, 10, 20, 30]
nb-runs: 10
random-seed: 0

data:
        source: Data/150users
        format: csv-normal
        transaction-size: 10

identification-methods:
        - type: Closed
        name: H1
        closed-method: Charm
        weight: TfIdf
        max-pattern-size: 5
        max-own-patterns: 40
        distance: Kulczynski
        profiler: Support
```

Fig. 2. Configuration file for the case study

is to recognise a user amongst n. The classifier has to guess the corresponding user from an anonymous behaviour. If it fails to recognise the declared user, then the identity is not confirmed. In a security context, this situation can lead to restrictions in the system, or to the request of some explicit means of identification. The parameters used in this study are detailed in the configuration file of Fig. 2.

3.1 Data description

Our data comes from Blaise Pascal university proxy servers. It consists of 17×10^6 lines of connection logs from more than 3, 000 users and contains the user ID, the time stamp and a domain name for each line. We applied two types of filters on the domain names: blacklist filters and HTTP-request based filters. We used some lists[3] of domain names to remove all domains regarded as advertising. We also filtered the data by the status code obtained after a simple HTTP request on the domain name. After those steps, we still have 4×10^6 lines. We divide the file between the 3K users to obtain the class files. This dataset is available at http://fc.isima.fr/~kahngi/cez13.zip. The studies were conducted on the 150 users with the higher number of requests.

[3] http://winhelp2002.mvps.org/hosts.htm and https://pgl.yoyo.org/as.

Some information about the data is available in Table 1. The table shows some statistics from before preprocessing and after the filters were applied. $\#Users$ represent the number of users, $\#Sites$ represents the cardinal of the whole set of websites for all users and Avg#lines/user is the average number of line per user. We can see that the number of users decreases because some users did not have a single line after the filters. Roughly 40% of the websites were deleted by the filters, and the average number of lines by user was divided by 5.

Table 1. Descriptive statistics of the dataset

	$\#Users$	$\#Sites$	Avg#lines/user
Raw Data	3388	96184	5082
After preprocessing	3370	57654	1145

3.2 Experimental parameters

Our tool implements several heuristics. H_0 considers frequent 1-patterns with the best support, H_0Lift considers frequent 1-patterns with the best lift, H_1 considers closed k-patterns with the best $tf \times idf$, and B is a smoothed Bayes classifier. The $tf \times idf$ is a metric that comes from information retrieval and text mining. It is the product of term frequency and inverse document frequency. It reflects how discriminating a pattern is for a given class. The experiments in [9] show that $tf \times idf$ produces better results than the lift or the support.

Figure 3 shows the kind of results that can be obtained with our tool. The abscissa is the number of anonymous transactions given to the classifier and the ordinate the accuracy of the different heuristics. The dataset is divided as follows: $\frac{2}{3}$ of learning base for the learning step and $\frac{1}{3}$ for the identification step. The division is random. Each test session consists of multiple runs of both those steps. That allows us to smooth the results by using the average accuracy.

$\mathcal{N}_C$	$\mathcal{A}$	Method	Avg accuracy	Min accuracy	Max accuracy	T_C
150	1	H_1	0.31266	0.30213	0.32118	75.796%
150	2	H_1	0.34378	0.32827	0.35666	93.335%
150	5	H_1	0.46909	0.4505	0.48996	99.676%
150	10	H_1	0.67352	0.63714	0.69905	100%
150	20	H_1	0.87778	0.85111	0.92	100%
150	30	H_1	0.94833	0.92333	0.96333	100%

Table 2. Output of the tool

The table generated by our tool contains 15 columns. Those include the number of classes $\mathcal{N}_C$, the number of anonymous transactions recieved $\mathcal{A}$, the method used, the average accuracy, the average number of transactions successfully and

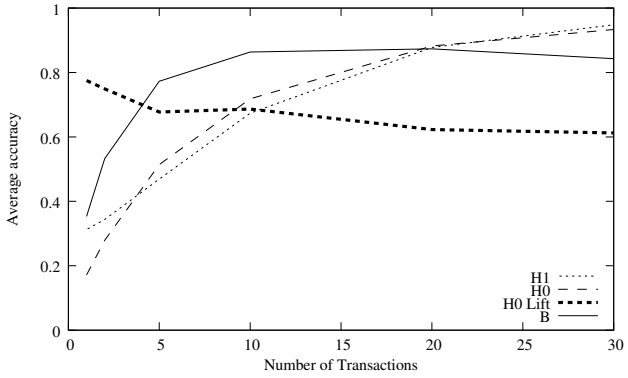

Fig. 3. Average accuracy of the different heuristics as a function of the number of anonymous transactions recieved by the classifier.

not successfully classified, the ratio of classified and not classified tests, and the run time in second. T_C represents the ratio of classified tests. All this information allows the analysis of various aspects of the result. Some are presented in Table 2.

4 Conclusion and perspectives

We presented a tool for classification of sequential data. It includes a lot of features in the construction of the transactions, and different parameters and heuristics for classification. The tool is flexible and adaptable to many contexts of classification and types of data.

The perspectives of our work are to add others means of classification based on other types of patterns (such as closed-sequences, pattern structures, or class association rules), and other types of metrics (for example structural metrics such as stability). We are also considering the use of aggregation functions other than the average for fuzzy support, such as ordered weighted averaging (OWA) operators [15, 16], or some power-means. Moreover, we are considering the integration of different paradigms of user profiles. Another way to construct the profile of a class is using association rules. Class association rules are association rules of the form $A \rightarrow C$ where C is a class and A a subset of items. They are studied in [17]. By attributing scores to the rules and searching for the premisses of the rule in an anonymous transaction, we could classify the anonymous transaction in a given class.

Acknowledgement

This research was partially supported by the European Union's "*Fonds Européen de Développement Régional (FEDER)*" program.

References

1. Azkia, H., Cuppens-Boulahia, N., Cuppens, F., Coatrieux, G.: Log content extraction engine based on ontology for the purpose of a posteriori access control. IJKL **9**(1/2) (2014) 23–42
2. Kuznetsov, S.O.: Complexity of learning in concept lattices from positive and negative examples. Discrete Applied Mathematics **142**(1-3) (2004) 111–125
3. Kuznetsov, S.O.: Machine learning and formal concept analysis. In: Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings. (2004) 287–312
4. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: On mining complex sequential data by means of FCA and pattern structures. Int. J. General Systems **45**(2) (2016) 135–159
5. Kuznetsov, S.O.: Learning of simple conceptual graphs from positive and negative examples. In: Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD '99, Prague, Czech Republic, September 15-18, 1999, Proceedings. (1999) 384–391
6. Ramamohanarao, K., Fan, H.: Patterns based classifiers. World Wide Web (1) (2007) 71–83
7. Novak, P.K., Lavrac, N., Webb, G.I.: Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. Journal of Machine Learning Research **10** (2009) 377–403
8. García-Borroto, M., Trinidad, J.F.M., Carrasco-Ochoa, J.A.: A survey of emerging patterns for supervised classification. Artif. Intell. Rev. **42**(4) (2014) 705–721
9. Coupelon, O., Dia, D., Labernia, F., Loiseau, Y., Raynaud, O.: Using closed itemsets for implicit user authentication in web browsing. In: Proceedings of the Eleventh International Conference on Concept Lattices and Their Applications, Košice, Slovakia, October 7-10, 2014. (2014) 131–144
10. Shi, E., Niu, Y., Jakobsson, M., Chow, R.: Implicit authentication through learning user behavior. In: Information Security - 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers. (2010) 99–113
11. Kumar, R., Tomkins, A.: A characterization of online browsing behavior. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010. (2010) 561–570
12. Yang, Y.C.: Web user behavioral profiling for user identification. Decision Support Systems **49**(3) (2010) 261–271
13. Goel, S., Hofman, J.M., Sirer, M.I.: Who does what on the web: A large-scale study of browsing behavior. In: Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012. (2012)
14. Abramson, M., Aha, D.W.: User authentication from web browsing behavior. In: Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2013, St. Pete Beach, Florida. May 22-24, 2013. (2013)
15. Yager, R.R.: Families of OWA operators. **59** (1993) 125–148
16. Yager, R.R.: Constraint satisfaction using soft quantifiers. International Journal of Intelligent Systems in Accounting and Finance Management **12**(3) (2004) 177–186
17. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York City, New York, USA, August 27-31, 1998. (1998) 80–86