Interactive Pattern Sampling for Characterizing Unlabeled Data

Arnaud Giacometti and Arnaud Soulet

Université François Rabelais Tours, LI EA 6300
firstname.lastname@univ-tours.fr

Abstract. Many data exploration tasks require a target class. Unfortunately, the data is not always labeled with respect to this desired class. Rather than using unsupervised methods or a labeling pre-processing, this paper proposes an interactive system that discovers this target class and characterizes it at the same time. More precisely, we introduce a new interactive pattern mining method that learns which part of the dataset is really interesting for the user. By integrating user feedback about patterns, our method aims at sampling patterns with a probability proportional to their frequency in the interesting transactions. We demonstrate that it accurately identifies the target class if user feedback is consistent. Experiments also show this method has a good true and false positive rate enabling to present relevant patterns to the user.

Keywords: Pattern mining ; Pattern Sampling ; Unlabeled Data

1 Introduction

Many data exploration tasks are intended to characterize a part of data over another [9]. For instance, it is particularly the case to identify factors of a disease by comparing the data of ill patients to those of healthy ones, or to find fraudulent behaviors by comparing the data of scammers to others. Unfortunately, in practice, the collected data have not always the labels allowing to know what class an individual (healthy or sick) or a behavior (normal or fraudulent) belongs to. Of course, when the class label to characterize is absent, it is possible to use unsupervised analysis techniques (such as clustering, association rules or detection of outliers) to identify and characterize the target class. However, these techniques are often less effective because they focus on the majority trends taking into account all the data. To address this problem, an approach would consist in labeling data during the data preparation phase. Such a labeling process could be facilitated by an active learning method that can even be dedicated to an analysis approach [13]. However, labeling remains a particularly costly and tedious task, especially when the target class to study is really in minority. Furthermore, in many cases, the labeling can be difficult because experts have only an imperfect knowledge of the target class. Actually, this is another reason for the experts want the use of data mining tools. In other words, we are facing a vicious circle: data analysis requires labeling which itself requires

an analysis of data. Thus, the problem is how to label data to identify a target class while characterizing this target class with patterns.

In order to solve this problem, we propose to use the interactive pattern mining framework introduced in [7]. The central idea of this framework is to alternate between three steps. During the *mining step*, our system mines an initial batch of patterns using an adaptation of the two-step random procedure proposed in [3]. During the *interactive step*, the user provides feedback by evaluating whether the patterns of the batch are good descriptors or not of the target class. Then, during the *learning step*, the system updates a model of the target class using the user feedback. Thus, after each interaction with the user, we have a twofold challenge to overcome: i) How can we update the model of the target class integrating the user feedback? and ii) How can we draw patterns from the dataset taking into account the updated model of the target class?

In this paper, we propose a new interactive pattern sampling method to solve these two challenges at the same time. The outline of this paper is as follows. Section 2 reviews some work about active learning and interactive pattern mining. We state precisely our problem in Section 3. Our algorithm proposal is detailed in Section 4 where theoretical properties are presented (due to lack of space, proofs are omitted). Indeed, if the user feedback is consistent with the target class, we demonstrate that the transactions of the target class will be clearly identified and that the mined patterns will describe exactly these transactions. Finally, experiments in Section 5 show that the accuracy of the interactive system increases fairly quickly with the number of user feedback responses.

2 Related Work

To the best of our knowledge, there is no work on the mining of patterns characterizing a target class not known in advance. However, we benefit from the framework of interactive pattern mining [7]. Its primary goal is to present interesting patterns to the user. Even if user feedback is used for labeling data, this problem therefore differs from traditional active learning problems [12], the purpose of which is not to propose interesting queries to the user. This distinction is important for different reasons. First, the queries provided to the user are patterns, not transactions. In most active learning tasks, the feedback requested from the user is directly related to the objects to be classified and not a generalization of these objects (although there are few notable exceptions [10,1]). Second, the selection of the query presented to the user cannot only target the improvement of the classification model unlike conventional active learning. In order that the user continues to interact with the system, the latter has to mine patterns that are interesting for him/her (i.e., that describe the target class). Third, the query presented to the user at each iteration has to be computed in few seconds to maintain a satisfactory interaction. In traditional active learning, this constraint is not very strong because the query is selected from the dataset. It is much more difficult to mine the right pattern due to the huge search space.

In interactive pattern mining, one challenge is to select the relevant patterns while improving the learned model. In case of preference learning, the early methods [14,11] ignored the use of a criterion favoring the diversity of queries for acquiring a complete view of preferences. A recent approach [5] nevertheless showed interest to address this issue as done in active learning. It also showed the importance of randomization to promote good diversity. This randomization need justifies the use of pattern sampling [2]. In this paper, we also take advantage of the statistical properties of sampling to better learn the classification model and to better choose the query (mined patterns) as done in [6].

Another challenge is to mine new patterns at each iteration in few seconds to maintain a satisfactory interaction. This speed requirement is not satisfied by traditional methods of pattern mining. Thus, the first proposals [14,11] were based on a preliminary mining step and then, they re-ranked this preliminary collection of patterns according to the updated criterion stemming from the user preference model. This post-processing approach did not allow the discovery of new patterns. More recently, a beam search method [5] was proposed to extract at each iteration the new patterns that maximize the updated criterion (combining quality and diversity, in that case). Such an approach remains slow and it fails to find various patterns. In this context, pattern sampling [2] is an attractive technique because it gives a fast access to all the patterns, guaranteeing a very good diversity. In this paper, rather than using a stochastic method [2] or a SAT framework [4], we adopt the two-step procedure [3] that is linear with the database size.

3 Problem Statement

This section formulates the problem of characterizing a class from an unlabeled dataset, using pattern sampling and an interactive approach. Before, we remind basic definitions about pattern mining and we introduce the notion of oracle.

3.1 Basic definitions

Let $\mathcal{I}$ be a set of distinct literals called items, an itemset (or a pattern) is a subset of $\mathcal{I}$ and the language of itemsets $\mathcal{L}$ is $2^{\mathcal{I}}$ (where 2^S denotes the powerset of S). A transactional dataset $\mathcal{D}$ is a multi-set of itemsets of $\mathcal{L}$. Each itemset, usually called transaction, is a data observation. For instance, Table 1 gives a transactional dataset with 4 transactions $t_1, \ldots, t_4$ described by 5 items A, B, C, D and E. Δ denotes the set of all datasets.

Pattern discovery takes advantage of interestingness measures to evaluate the relevancy of a pattern. More precisely, an interestingness measure for a pattern language $\mathcal{L}$ is a function defined from $\mathcal{L} \times \Delta$ to $\Re$. For instance, the support of an itemset X in a dataset $\mathcal{D}$, denoted $supp(X, \mathcal{D})$, is the proportion of transactions containing X: $supp(X, \mathcal{D}) = |\{t \in \mathcal{D} \ : \ X \subseteq t\}|/|\mathcal{D}|$. Pattern sampling aims at accessing the pattern space $\mathcal{L}$ by a sampling procedure simulating a distribution $p : \mathcal{L} \rightarrow [0, 1]$ that is defined with respect to an interestingness measure m:

$\mathcal{D}$

Trans.	Items				Class
t_1	A	B		E	$+$
t_2	A	B			$+$
t_3		B	C	D	$-$
t_4		B	C		$-$
	Known in advance				Unknown

Table 1: A toy dataset $\mathcal{D}$

Trans.	Init.	B $(-)$	BE $(+)$	BD $(-)$
t_1	0.50 ± 0.5	$\mathbf{0.27 \pm 0.5}$	$\mathbf{0.51 \pm 0.5}$	0.51 ± 0.5
t_2	0.50 ± 0.5	$\mathbf{0.27 \pm 0.5}$	0.27 ± 0.5	0.27 ± 0.5
t_3	0.50 ± 0.5	$\mathbf{0.27 \pm 0.5}$	0.27 ± 0.5	$\mathbf{0.13 \pm 0.3}$
t_4	0.50 ± 0.5	$\mathbf{0.27 \pm 0.5}$	0.27 ± 0.5	0.27 ± 0.5

Table 2: Evolution of weights with feedback

$p(.) = m(.)/Z$ where Z is a normalizing constant. In this way, with no parameter (except possibly the sample size), the user has a fast and direct access to the entire pattern language.

Assume now that the dataset $\mathcal{D}$ is partitioned into two subsets, denoted by $\mathcal{D}^+$ and $\mathcal{D}^-$, such that $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$ and $\mathcal{D}^+ \cap \mathcal{D}^- = \emptyset$. We say that the sub-dataset $\mathcal{D}^+$ contains the set of *positive* transactions, whereas the sub-dataset $\mathcal{D}^-$ contains the set of *negative* transactions. In our toy example (see Table 1), t_1 and t_2 are positive transactions, whereas t_3 and t_4 are negative ones. In our approach, we assume that the sub-datasets $\mathcal{D}^+$ and $\mathcal{D}^-$ are not known in advance, whereas the user want to discover patterns that characterize the subset $\mathcal{D}^+$ of positive transactions. In our toy example (see Table 1), because $supp(BE, \mathcal{D}^+) = 0.5$ and $supp(BE, \mathcal{D}^-) = 0$, the user is definitely interested by pattern BE. But, he/she is less interested by pattern B since $supp(B, \mathcal{D}^+) = supp(B, \mathcal{D}^-) = 1$.

In that context, we assume that an oracle $\mathcal{O} : \mathcal{L} \rightarrow \{-, +\}$ models the user feedback. It means that $\mathcal{O}(X) = +$ (resp. $-$) iff the oracle gives a positive (resp. negative) feedback response for the pattern X. In Table 2, three patterns are drawn (B, BE and BD) and the user feedback is indicated in parentheses. Since the user feedback about the same pattern X may change during the process (the user may have an imperfect knowledge of the set $\mathcal{D}^+$ of positive transactions), we consider that $\mathcal{O}$ is a random variable. Thereby, $\mathbf{P}(+|X)$ will denote the probability of having a positive feedback given X when the oracle is consulted. For instance, in our toy example, because $supp(BE, \mathcal{D}^+) = 0.5$ and $supp(BE, \mathcal{D}^-) = 0$, we could assume that $\mathbf{P}(+|BE) = 1$, meaning that the oracle always gives a positive feedback for BE. On the other hand, because $supp(B, \mathcal{D}^+) = supp(B, \mathcal{D}^-) = 1$, we could assume that $\mathbf{P}(+|B) = 0.5$, meaning that the user could evaluate pattern B positively or negatively according to the objective of the user (i.e., discrimination or characterization of the target class).

3.2 Problem Formulation

In our context, since the user is not interested in all transactions in $\mathcal{D}$, but only in positive transactions in $\mathcal{D}^+$, we do not want to sample the pattern space according to the interestingness measure m evaluated on $\mathcal{D}$, but on $\mathcal{D}^+$. Indeed, the interestingness measure m evaluated on $\mathcal{D}^+$ is better suited because it enables us to focus on the patterns describing the set of positive transactions. Unfortunately, the set of positive transactions in $\mathcal{D}^+$ is not known in advance. Therefore, our problem can be formalized as follows:

Algorithm 1 Interactive pattern sampling

Input: A dataset $\mathcal{D}$ and an oracle $\mathcal{O}$
1: Let F be an empty sequence
2: Let $\omega_F(t) := 0.5$ for all $t \in \mathcal{D}$
3: **repeat**
4: Draw a pattern X from $\mathcal{D}$ according to its weighted support $supp_\omega$
5: Add the user feedback to the sequence F
6: Update the weight vector ω_F using F
7: **until** The user stops the process

Problem 1 *Given a dataset $\mathcal{D}$ containing an unknown set of positive transactions $\mathcal{D}^+$ and an oracle $\mathcal{O}$, our problem consists in building a sequence of patterns $\langle X_1, \ldots, X_k \rangle$ such that the probability to draw a pattern X_i at step i tends to $supp(X_i, \mathcal{D}^+)/Z$ when i tends to $+\infty$ where Z is a normalizing constant.*

Note that at each step i, the oracle $\mathcal{O}$ will be used to evaluate the interestingness of the pattern X_i presented to the user. The next sections show how to choose these patterns X_i and how the user feedback $\mathcal{O}(X_i)$ can be used by the system to improve its knowledge of $\mathcal{D}^+$.

4 Interactive Pattern Sampling Algorithm

4.1 General Principles of the Approach

For addressing the problem formalized in Section 3.2, Algorithm 1 provides a sketch of our interactive system. Its key idea is to associate a weight $\omega_F(t)$ to each transaction $t \in \mathcal{D}$ that maintains an estimation of the class conditional probability $\mathbf{P}(+|t)$ (the probability that a transaction t belongs to $\mathcal{D}^+$). Of course, all these weights $\omega_F(t)$ are initialized to 0.5 because the class is unknown at the beginning (line 2)[1], as shown in the second column in Table 2. But, at the end, the goal is to have $\omega_F(t) = 1$ iff $t \in \mathcal{D}^+$ (0 otherwise). For this purpose, our system alternates between three steps as proposed in [7]:

- **Mining step (line 4):** This step provides patterns by favoring those which are frequent in transactions with high weights ω_F. More precisely, a pattern X is sampled according to a weighted support $supp_\omega$. Typically, after the positive feedback on BE (see Table 2), AB will be more likely to be drawn than BC because the total weight of t_1 and t_2 becomes higher than that of t_3 and t_4.
- **Interactive step (line 5):** During this step, the user evaluates whether the pattern X is a good descriptor or not of the unknown sub-dataset $\mathcal{D}^+$ of positive transactions.

[1] It is also possible to set weights to 0 or 1 if the labels of some transactions are already known.

Algorithm 2 Weighted Support-based Sampling

Input: A dataset $\mathcal{D}$ and a weight vector ω
Output: A random itemset $X \sim supp_\omega(\mathcal{L}, \mathcal{D})$
1: Let weight vector ω' be defined by $\omega'(t) := 2^{|t|} \times \omega(t)$ for all $t \in \mathcal{D}$
2: Draw a transaction $t \sim \omega'(\mathcal{D})$
3: **return** an itemset $X \sim u(2^t)$

– **Learning step (line 6):** The system updates the weight $\omega_F(t)$ of each transaction t containing X. Basically, if the user feedback is positive, the weight $\omega_F(t)$ is increased otherwise it is decreased (see Section 4.3 for more details). For instance, in Table 2, the weight of t_1 is increased after the draw of BE while that of t_3 is decreased after the draw of BD.

In order that our system works, it is necessary to link the user feedback given on patterns (i.e., $\mathbf{P}(+|X)$) to the class conditional probabilities on transactions (i.e., $\mathbf{P}(+|t)$). Our approach is based on this central result which is independent of the mining and learning steps:

Property 1 (Class Conditional Probability). Given a transaction t in $\mathcal{D}$ and a pattern langage $\mathcal{L}$, we have: $\mathbf{P}(+|t) = \sum_{X \in \mathcal{L}} \mathbf{P}(X|t) \times \mathbf{P}(+|X)$.

It is impossible to calculate the exact class conditional probability of a transaction because its calculation depends on the entire pattern language $\mathcal{L}$. Using Property 1, we show in Section 4.3 how we can estimate $\mathbf{P}(+|t)$ given a sequence of user feedback responses. Previously, while $\mathbf{P}(+|X)$ is straightforwardly provided by the oracle, the method used to a draw sequence of patterns X is necessary to further detail $\mathbf{P}(X|t)$. This method is presented in the following Section 4.2.

4.2 Pattern sampling according to the weighted support

In [3], the authors show how to sample patterns following a distribution proportional to their support. In our approach, we propose to sample patterns following a distribution proportional to their *weighted* support. More formally, given a dataset $\mathcal{D}$ and a weight vector w, the weighted support of a pattern X in $\mathcal{D}$, denoted $supp_\omega(X, \mathcal{D})$, is defined by: $supp_\omega(X, \mathcal{D}) = \sum_{t \in \mathcal{D}, X \subseteq t} \omega(t) / \big(\sum_{t \in \mathcal{D}} \omega(t) \big)$.

Algorithm 2 adapts the two-step random procedure [3] to sample patterns according to their weighted supports. Using this algorithm, the weighted support is similar to the usual support at the beginning (when all weights are equal to 0.5). More interestingly, it is easy to see that $supp_\omega(X, \mathcal{D}) = supp(X, \mathcal{D}^+)$ (which solves Problem 1) if all positive transactions in $\mathcal{D}^+$ have 1 as weight and other transactions have 0 as weight after a long sequence of interactions with the user. However, we still have to show how we can learn the weights of the transactions, which is the goal of the following section.

Algorithm 3 Learning the weights

Input: A sequence $F = \{(X_1, f_1, s_1), ..., (X_k, f_k, s_k)\}$ of k user feedback responses
Output: A updated set of weights $\omega(t)$

1: **for all** $t \in \mathcal{D}$ **do**
2: $\quad \bar{\omega}_F(t) := \frac{\sum_{(X_j, f_j, s_j) \in F, X_j \subseteq t} f_j / s_j}{\sum_{(X_j, f_j, s_j) \in F, X_j \subseteq t} 1 / s_j}$
3: $\quad \omega_F(t) := \frac{inf_F(t) + sup_F(t)}{2}$
4: $\quad$ **if** $inf_F(t) > 0.5$ **then** $\omega_F(t) := 1$
5: $\quad$ **if** $sup_F(t) < 0.5$ **then** $\omega_F(t) := 0$
6: **end for**

4.3 Learning the weights of transactions

In this section, we show how we can update the weights of the transactions from the user feedback. Assuming that patterns are sampled using Algorithm 2, given a transaction $t \in \mathcal{D}$, we know that $\mathbf{P}(X|t) = 0$ if $X \not\subseteq t$ and $\mathbf{P}(X|t) = \frac{1}{|2^t|}$ if $X \subseteq t$. Thus, using Property 1, we finally have:

$$\mathbf{P}(+|t) = \sum_{X \in \mathcal{L}} \mathbf{P}(X|t) \times \mathbf{P}(+|X) = \frac{1}{2^{|t|}} \sum_{X \subseteq t} \mathbf{P}(+|X) \tag{1}$$

Using this equation, Algorithm 3 shows how the probabilities $\mathbf{P}(+|t)$ can be estimated from a sequence of user feedback responses, and how these estimations can be used to update the weights of the transactions. Let $F = \{(X_1, f_1, s_1), ..., (X_k, f_k, s_k)\}$ be a sequence of k user feedback responses, where X_k is the pattern drawn at step k in Algorithm 1, $f_k = 1$ if $\mathcal{O}(X_k) = +$ (0 otherwise), and $s_k = supp_\omega(X_k, \mathcal{D})$. At step 2 of Algorithm 3, we start to compute a first estimation $\bar{\omega}_F(t)$ of $\mathbf{P}(+|t)$ using a weighted arithmetic mean. The following property shows that $\bar{\omega}_F(t)$ tends to $\mathbf{P}(+|t)$ when the number of user feedback responses tends to infinity.

Property 2 (Probability Estimations). Given a dataset $\mathcal{D}$, for every transaction $t \in \mathcal{D}$, the weight $\bar{\omega}_F(t)$ converges to $\mathbf{P}(+|t)$ when the number of user feedback responses $|F|$ tends to infinity.

In practice, this property means that the addition of new feedback responses tends to improve the estimation of the probability $\mathbf{P}(+|t)$. In order to evaluate the estimation error, we benefit from a statistical result known as Bennett's inequality which is true irrespective of the probability distribution [8]. After k independent observations of a real-valued random variable r with range $[0, 1]$, Bennett's inequality ensures that, with a confidence $1 - \delta$, the true mean of r is at least $\overline{r} - \epsilon$ where $\overline{r}$ and $\overline{\sigma}$ are respectively the observed mean and standard deviation of the samples and $\epsilon = \sqrt{\frac{2\overline{\sigma}^2 \ln(1/\delta)}{k}} + \frac{\ln(1/\delta)}{3k}$. We use this statistical result to bound the true value of $\mathbf{P}(+|t)$ from a sequence of user feedback responses F:

Property 3 (Bounds). Given a dataset $\mathcal{D}$, a sequence of user feedback responses F and a confidence $1 - \delta$, the probability $\mathbf{P}(+|t)$ for a transaction t is bounded

as follows:

$$\underbrace{\max\{0, \bar{\omega}_F(t) - \epsilon\}}_{inf_F(t)} \leq \mathbf{P}(+|t) \leq \underbrace{\min\{\bar{\omega}_F(t) + \epsilon, 1\}}_{sup_F(t)}$$

with $\epsilon = \sqrt{2\overline{\sigma}^2 \ln(1/\delta)/k} + \ln(1/\delta)/3k$ where $\overline{\sigma} = \sqrt{\bar{\omega}_F(t) - \bar{\omega}_F(t)^2}$ is the empirical standard deviation of $\bar{\omega}_F(t)$.

This property is important because it gives information about the error of the estimation $\bar{\omega}_F$. In Algorithm 3, we use this property to compute $\omega_F(t) = \frac{inf_F(t)+sup_F(t)}{2}$, i.e. a corrected estimation of $\mathbf{P}(+|t)$. Since both bounds tend to $\mathbf{P}(+|t)$, it is easy to see that the corrected estimation $\omega_F(t)$ also tends to $\mathbf{P}(+|t)$ when the number of feedback responses increases. Finally, at lines 4 and 5 of Algorithm 3, we force the weight $\omega_F(t)$ to tend to 1 (resp. 0) when it is certain (with respect to the confidence level) that the probability $\mathbf{P}(+|t)$ is higher than 0.5 (resp. $\mathbf{P}(+|t) < 0.5$). For instance, after the evaluation of BD in Table 2, the final weight $\omega_F(t_3)$ will be zero because $0.13 + 0.3 = 0.43$ is below 0.5.

4.4 Convergence and Complexity

It may be that we do not properly learn the set of positive transactions from the user feedback on the patterns if his/her feedback is not consistent. For instance, if $\mathbf{P}(+|X) = 0$ for all patterns $X \subseteq t$, then we compute $\mathbf{P}(+|t) = 0$ even if t is truly a positive transaction. Therefore, we introduce the notion of consistent oracle:

Definition 1 (Consistency). *Given a set $\mathcal{D}^+ \subseteq \mathcal{D}$ of positive transactions , an oracle $\mathcal{O}$ is consistent with $\mathcal{D}^+$ iff for all transaction $t \in \mathcal{D}$, we have $\mathbf{P}(+|t) > 0.5$ if $t \in \mathcal{D}^+$, and $\mathbf{P}(+|t) < 0.5$ otherwise.*

Using this definition of consistency, and Property 3, it is possible to conclude on the good convergence of Algorithm 1:

Theorem 1 (Convergence). *Given $\mathcal{D}$ with $\mathcal{D}^+ \subseteq \mathcal{D}$ and an oracle $\mathcal{O}$ consistent with $\mathcal{D}^+$, for each transaction $t \in \mathcal{D}$, the weight $\omega_F(t)$ converges to 1 iff $t \in \mathcal{D}^+$ (otherwise to 0) when the number of user feedback responses $|F|$ tends to infinity. Consequently, the weighted support tends to the support in $\mathcal{D}^+$.*

Under the assumption of consistency, Algorithm 1 clearly solves the problem stated in Section 3.2. Interestingly, the time complexity of this approach in $O(k|\mathcal{D}||\mathcal{I}|)$ (where k is the number of mined patterns) is excellent. Finally, as the weights can be calculated without keeping the details of all user feedback, the space complexity of the algorithm is linear with the size of the dataset.

5 Experimental Study

This section has the twofold objective of evaluating the quality of the class learning through user feedback and the quality of the patterns presented to the

$\mathcal{D}$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{D}^+_{min}\rvert$	$\lvert\mathcal{D}^+_{min}\rvert/\lvert\mathcal{D}\rvert$	$\mathcal{D}$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{D}^+_{min}\rvert$	$\lvert\mathcal{D}^+_{min}\rvert/\lvert\mathcal{D}\rvert$
abalone	4,177	28	1,307	0.31	mushroom	8,124	119	3,916	0.48
chess	3,196	75	1,527	0.48	page	941	35	9	0.01
cmc	1,473	28	469	0.32	sick	2,800	58	171	0.06
german	1,000	76	300	0.30	vehicle	846	58	199	0.24
hypo	3,163	47	151	0.05					

Fig. 1: Features of UCI benchmarks

user. Note that the Java source code of the implementation used for this study is available at `www.info.univ-tours.fr/~soulet/prototype/ida17/`.

Protocol We report the experimental evaluations conducted on 9 datasets coming from the UCI Machine Learning Repository (`archive.ics.uci.edu/ml`). Table 1 provides the main features of each dataset. For each dataset $\mathcal{D}$, the minority class of $\mathcal{D}$ corresponds to the set of positive transactions. The cardinality of this minority class, denoted $\mathcal{D}^+_{min}$, is indicated in the last column of Table 1. We first perform experiments using a deterministic oracle (in the sense that its answer is constant for a given pattern). Given a set of positive transactions $\mathcal{D}^+ \subseteq \mathcal{D}$, this deterministic oracle is defined as $\mathcal{O}(X) = +$ if $supp(X, \mathcal{D}^+) > supp(X, \mathcal{D})$ ($-$ otherwise). Intuitively, a user is interested in a pattern if its support in the set of positive transactions $\mathcal{D}^+$ is higher than its support in the dataset $\mathcal{D}$.

First, we evaluate the quality of the *mining step* by considering the number of interesting patterns, i.e patterns positively rated by the user. More precisely, we compute the ratio of positive feedback responses over the last 50 patterns provided to the user i.e., $\mathbf{P}(+) = \sum_{i=k-49}^{k} f_k/50$ given a sequence of user feedback responses $F = \langle (X_1, f_1, s_1), \ldots, (X_k, f_k, s_k) \rangle$ with $k \geq 50$.

Second, a confusion matrix is used for evaluating the quality of the *learning step*. More precisely, we consider that a transaction is classified in the positive class (resp. negative class) if its weight considering the margin of error is greater than 0.5 (resp. < 0.5). Thus, we introduce two sets of transactions defined by: $\mathcal{P}^+ = \{t \in \mathcal{D} \mid inf_F(t) > 0.5\}$ and $\mathcal{P}^- = \{t \in \mathcal{D} \mid sup_F(t) < 0.5\}$. In the very first iterations, it is clear that no class is predicted, i.e. $\mathcal{P}^+ = \mathcal{P}^- = \emptyset$. Then, as the interactions progress, the proportion of classified transactions, defined by $Completeness = \frac{|\mathcal{P}^+ \cup \mathcal{P}^-|}{|\mathcal{D}|}$, increases. In order to evaluate the quality of the learning step, we also use the True Positive Rate (*TPR*) and the False Positive Rate (*FPR*) measures defined respectively by: $TPR = \frac{|\mathcal{D}^+ \cap \mathcal{P}^+|}{|\mathcal{D}^+|}$ and $FPR = \frac{|\mathcal{D}^- \cap \mathcal{P}^+|}{|\mathcal{D}^-|}$.

All experiments were repeated 100 times and the arithmetic mean is used for averaging the values coming from those 100 measurements. Finally, the confidence level $1 - \delta$ is set to 0.8.

Convergence The left part of Figure 2 gives the proportion of positive feedback responses with respect to the number of iterations (i.e., the number of patterns presented to the user). As expected, this quality measure increases as the number

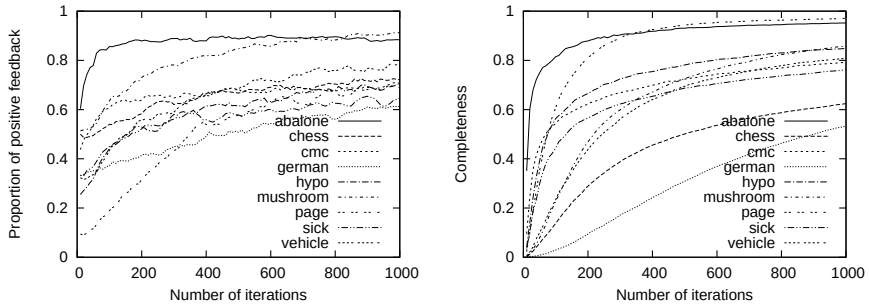

Fig. 2: Proportion of positive feedback responses and completeness

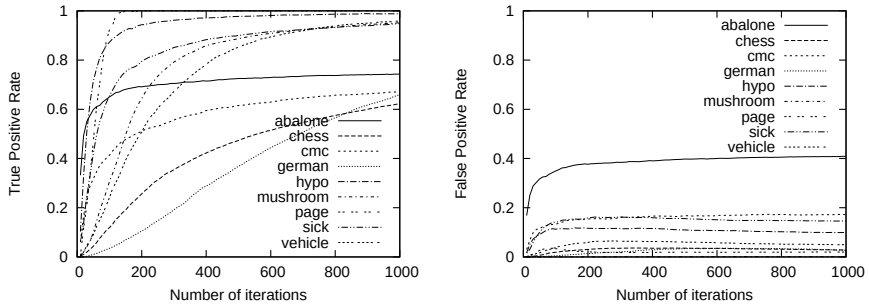

Fig. 3: *TPR* and *FPR* with the number of iterations

of user feedback responses increases, which means that more relevant patterns are presented to the user. Note that in the first iterations, the 4 datasets having the largest ratio $|\mathcal{D}^+_{min}|/|\mathcal{D}|$ (i.e., `abalone`, `chess`, `cmc`, `mushroom`) are also those having the best proportions of positive feedback responses. Indeed, it is easier to find patterns that characterize an important class than a small class as it is the case for `page`. However, after a sufficient number of iterations, the system is efficient to propose relevant patterns even for small positive classes. Furthermore, we can see that the proportion of positive feedback responses does not converge towards 1. This observation can be explained by the nature of the oracle used in the experiments. Indeed, an oracle based on a contrast measure is unfavorable to our sampling method based on a description measure (i.e., support). It can also be explained by the nature of the dataset. Indeed, the set of items of the dataset is in general not adequate to perfectly characterize the target class, i.e. the class of positive transactions.

The right part of Figure 2 gives the completeness (proportion of classified transactions) with the number of iterations. As expected, the completeness converges to 1 meaning that the method will arrive at classifying all transactions. Importantly, we observe that the method quickly learns the class of a majority

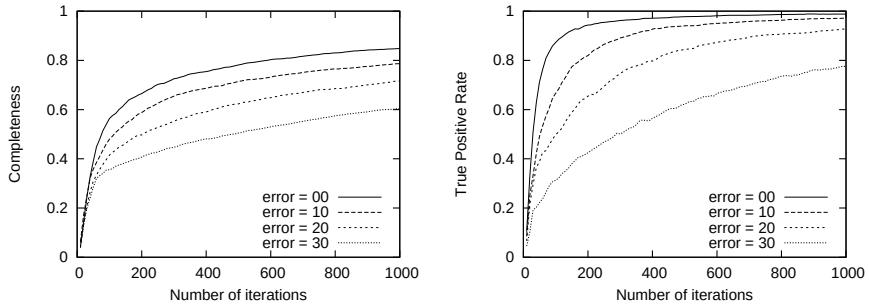

Fig. 4: Completeness and *TPR* according to an oracle with error on `hypo`

of transactions on most datasets. Indeed, after 300 patterns, the completeness is greater than 0.5 for all datasets except `chess` and `german` (in this case, the oracle does not discriminate the two classes well). In order to evaluate more precisely the quality of the learning step, Figure 3 plots the *TPR* and *FPR* with the number of iterations. Except for `chess` and `german` datasets, we observe that the *TPR* (proportion of positive transactions that are correctly classified in $\mathcal{P}^+$) increases and converges to their maximal value very fast. In particular, we can emphasize that it is the case for the datasets `hypo`, `page` and `sick`, even though the set of positive transactions for these datasets is very small (less than 6% of the whole dataset). Concerning the *FPR* (proportion of negative transactions incorrectly classified), we finally observe that it stabilizes to a low value (less than 20%) very fast (in less than 200 iterations) except for `abalone`.

Non-deterministic oracle We now evaluate the impact of non-deterministic oracle by introducing an error component to the oracle. Experiments are carried out on `hypo` with 4 different error probabilities 30%, 20%, 10% and 0% (it means that the oracle gives an opposite feedback in x% of its answers). By observing the *Completeness* and *TPR* in Figure 4, we observe that the convergence is guaranteed, but the required time increases with the error rate. Importantly, it is easy to see that the approach is robust because the error probability has no significant impact on the final value of the *TPR*, meaning that the set of positive transactions is correctly identified whatever the error probability.

6 Conclusion

This paper presents a new method of interactive pattern mining by benefiting from pattern sampling. Beyond its practical efficiency, this technique offers statistical guarantees on the learned class model and therefore, on the convergence of the interactive process. Experiments highlight this good convergence on several benchmarks. The number of classified transactions increases rapidly while the true and false positive rates remain satisfactory even if the target class consists in only few transactions. Besides, even if an end user can only make a limited

number of feedback responses, the good convergence of the system is interesting because it is possible to envisage such a system in a context of crowdsourcing. We would intend to generalize this method to other interestingness measures more sophisticated than support, including measures for identifying contrasts between $\mathcal{D}^+$ and $\mathcal{D}^-$.

Acknowledgements. This work has been partially supported by the Decade project, Mastodons 2017, CNRS.

References

1. Bessiere, C., Coletta, R., Hebrard, E., Katsirelos, G., Lazaar, N., Narodytska, N., Quimper, C.G., Walsh, T.: Constraint acquisition via partial queries. In: Proc. of the 23rd IJCAI. pp. 475–481 (2013)
2. Bhuiyan, M., Mukhopadhyay, S., Hasan, M.A.: Interactive pattern mining on hidden data: a sampling-based solution. In: Proc. of ACM CIKM. pp. 95–104 (2012)
3. Boley, M., Lucchese, C., Paurat, D., Gärtner, T.: Direct local pattern sampling by efficient two-step random procedures. In: Proc. of the 17th ACM SIGKDD. pp. 582–590 (2011)
4. Dzyuba, V., van Leeuwen, M., De Raedt, L.: Flexible constrained sampling with guarantees for pattern mining. Data Mining and Knowledge Discovery pp. 1–28 (2017)
5. Dzyuba, V., Leeuwen, M.v., Nijssen, S., De Raedt, L.: Interactive learning of pattern rankings. Int. Journal on Artificial Intelligence Tools 23(06), 32 pages (2014)
6. Giacometti, A., Soulet, A.: Anytime algorithm for frequent pattern outlier detection. International Journal of Data Science and Analytics 2(3-4), 119–130 (2016)
7. van Leeuwen, M.: Interactive data exploration using pattern mining. In: Interactive Knowledge Discovery and Data Mining in Biomedical Informatics, pp. 169–182. Springer (2014)
8. Maurer, A., Pontil, M.: Empirical Bernstein bounds and sample variance penalization. arXiv preprint arXiv:0907.3740 (2009)
9. Novak, P.K., Lavrač, 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(Feb), 377–403 (2009)
10. Rashidi, P., Cook, D.J.: Ask me better questions: active learning queries based on rule induction. In: Proc. of the 17th ACM SIGKDD 2011. pp. 904–912 (2011)
11. Rueping, S.: Ranking interesting subgroups. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 913–920. ACM (2009)
12. Settles, B.: A practical test for univariate and multivariate normality. Computer sciences technical report 1648, University of Wisconsin, Madison (2010)
13. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of Machine Learning Research 2, 45–66 (2001)
14. Xin, D., Shen, X., Mei, Q., Han, J.: Discovering interesting patterns through user's interactive feedback. In: Proc. of the 12th ACM SIGKDD 2006. pp. 773–778 (2006)