

A Lower Bound on the Sample Size needed to perform a Significant Frequent Pattern Mining Task

Stéphanie Jacquemont, François Jacquenet, Marc Sebban

Laboratoire Hubert Curien, Université de Saint-Etienne, 18 rue du Professeur Lauras, 42000 Saint-Etienne (France)

Abstract

During the past few years, the problem of assessing the statistical significance of frequent patterns extracted from a given set S of data has received much attention. Considering that S always consists of a sample drawn from an unknown underlying distribution, two types of risks can arise during a frequent pattern mining process: *accepting a false* frequent pattern or *rejecting a true* one. In this context, many approaches presented in the literature assume that the dataset size is an application-dependent parameter. In this case, there is a trade-off between both errors leading to solutions that only control one risk to the detriment of the other one. On the other hand, many sampling-based methods have attempted to determine the optimal size of S ensuring a good approximation of the original (potentially infinite) database from which S is drawn. However, these approaches often resort to Chernoff bounds that do not allow the independent control of the two risks. In this paper, we overcome the mentioned drawbacks by providing a lower bound on the sample size required to control both risks and achieve a significant frequent pattern mining task.

1 Introduction

In frequent pattern mining (1; 2), one aims to find interesting patterns from a database in the form of association rules, sequences, episodes, correlations, etc. Many algorithms have been proposed in the literature to deal with association rule mining (3; 4; 5), sequential pattern mining (6; 7; 8; 9), graph mining (10; 11; 12; 13), tree mining (14; 15; 16). Chao et al. (17) proposed a generic library for dealing with a large family of frequent patterns. This domain of

* This work is part of the ongoing BINGO2 research project financed by the French National Research Agency.

research has been applied in lots of applications (see (18) for a survey) such as the discovery of customers' behavior in supermarkets, the extraction of patterns of alarms in manufacturing supervision, the modeling of web users, etc. A pattern of a sample S is called *frequent* if its observed frequency is greater than a *minimal support threshold*. For instance, a sequence mining process consists of detecting *frequent subsequences* from a dataset of sequences that are made up of possibly non contiguous symbols. For example, let us assume that the database S is constituted of the following three sequences $S = \{ACGT, ATGAT, CAGTA\}$. By fixing the minimal support threshold to $\frac{2}{3}$, the pattern AGT is considered as frequent since it occurs three times out of the three sequences of S , that actually is $> \frac{2}{3}$.

During the past decade, the scientific community has mainly concentrated its efforts on the reduction of the complexity of the frequent pattern mining methods to deal with large datasets. In this context, the reduction of the search space has constituted one of the main objectives. From an algorithmic point of view, all these approaches have to prove that they are *correct* and *complete*, *i.e.* they must guarantee that (i) all the frequent patterns that are extracted are really frequent in S , and (ii) no frequent pattern of S has been overlooked. However, these properties are not sufficient to guarantee the significance of a frequent pattern mining process. Indeed, S being nothing else but a sample of an unknown target distribution \mathcal{D} , mining algorithms often suppose that the distribution over S is the same as \mathcal{D} from which these data have been drawn. In other words, they make no assessment of the likelihood that a frequent pattern extracted from S is an artifact of the sampling rather than a consistent pattern in the target distribution \mathcal{D} . In the same way, they do not assess the risk of overlooking a pattern that would be in fact frequent according to \mathcal{D} .

More formally, deciding if a pattern in S is frequent or not boils down to comparing its observed proportion \hat{p} with a given support threshold p_0 . If $\hat{p} > p_0$, the pattern is considered as frequent by the mining algorithm. However, the true probability p of this pattern comes under the unknown target distribution \mathcal{D} . Therefore, when an algorithm takes a decision about the status of a pattern, it takes a risk $\alpha \in [0, 1]$ of accepting a false frequent pattern (*i.e.* that appears in S due to chance alone), or a risk $\beta \in [0, 1]$ of rejecting a true frequent one. In this context, $1 - \alpha$ can be called the theoretical *precision*, whereas $1 - \beta$ corresponds to the theoretical *recall* of the algorithm. It is important to note that most frequent pattern mining algorithms do nothing (or little) to control both α and β . As mentioned before, their main goal “only” consists of guarantying their correctness and their completeness *over* S , but nothing is ensured *over the underlying distribution* \mathcal{D} . This can be justified by the fact that, statistically, given a constant number of data in S , reducing one of the two risks implies increasing of the second one.

In this paper, we differently take up this problem, by providing a lower bound

on the size of S needed to theoretically guarantee a *precision* of $(1 - \alpha)$ and a *recall* of $(1 - \beta)$ according to any distribution \mathcal{D} . Therefore, we reject the well known statement that to increase the *recall*, we have to accept to decrease the *precision*, or vice versa. We rather answer the following question: What is the minimal size of S required to satisfy given risks α and β ? We claim that this contribution is novel by comparison with the state of the art. Indeed, we will see that the few approaches attempting to deal with this problem from a theoretical point of view either are based on Chernoff bounds that do not allow the independent control of α and β , or call on statistical tests that require to choose the risk to optimize. Even though our contribution is above all theoretical, we claim that it can provide useful help in many applications. For instance, in domains where the data acquisition is not costly, one can wonder what is the minimal number of examples that are required to optimize the trade-off between the reduction of the algorithmic constraints and the guarantee of a discovery of *true* knowledge. Therefore, in such cases, our theoretical result provides a bound reachable in practice guaranteeing a significant frequent pattern mining task. This is the case for example in the modeling of web users' behavior, where tera-bytes of data are available in log files. On the other hand, in domains where the number of available examples is limited (in molecular biology for instance), it enables us to draw the attention of data miners on the fact that some extracted patterns could be the result of false discovery, and some others could have been omitted despite their significance. In this case, the use of the extracted knowledge must be done with caution.

The rest of the paper is organized as follows. In Section 2, we present the state of the art approaches aiming to assess the significance of the extracted patterns. Section 3 is devoted to the presentation of our bound enabling us to fix in advance α and β ; A first illustration is presented in Section 4 on a real database. In Section 5, we discuss about the valuation of the parameters of our bound, and we present a larger series of experiments.

2 Related Work

2.1 Bottleneck of frequent pattern mining algorithms

Let us suppose we carry out a series of experiments consisting of tossing a coin $N = 10$ times. Let S be the resulting sample of 10 itemsets constituted in this case of only one item ($\langle tails \rangle$ or $\langle heads \rangle$). Suppose we observe in S respectively 8 $\langle tails \rangle$ and 2 $\langle heads \rangle$. By fixing the support threshold to $p_0 = 0.5$, the pattern $tails$ will be considered as frequent in S , because its observed frequency $\hat{p} = 0.8$ is higher than p_0 , while the pattern $\langle heads \rangle$

will not be. Does it mean that the extracted knowledge “*<tails> is more frequent than <heads>*” is significant? In fact, we can easily prove that such a combination of *<heads>* and *<tails>* can “often” occur over only 10 trials, without challenging the balance of the coin itself. We can note that the size of S has a direct impact on the significance of the result. During the past few years, several papers have drawn the attention of data miners on the risks of extracting regularities from data in the form of a random artifact. The previous example is a good illustration of this problem that can arise in a frequent pattern mining process. Let us describe now some possible solutions that have been presented in the literature, and that take into account the size of S to overcome the mentioned drawback.

2.2 Modifying the support threshold p_0 using Chernoff bounds

Rather than directly comparing the observed frequency \hat{p} in S with p_0 , a first solution consists of bounding p_0 in order to take into account the estimate error $|\hat{p} - p|$ due to the use of a sample S of finite size N , where p is the true probability of the pattern under the unknown theoretical distribution \mathcal{D} .

A well-known non parametric approach that deals with this problem is based on Chernoff bounds that state that the estimate error between a random variable X observed on a sample S and its expected value $E(X)$ according to \mathcal{D} is lower bounded by ϵ , such that

$$\forall \epsilon \in]0, 1[, P(|X - E(X)| \geq \epsilon) < e^{-2N\epsilon^2}. \quad (1)$$

Eq.1 states that, obviously, the higher the sample size N , the smaller the estimate error. Chernoff bounds have been widely used in statistical learning theory for many years, and more recently in frequent pattern mining by *sampling-based methods* (19; 20) to deal with the statistical relevance of the extracted patterns. Basically, sampling-based data mining methods aim to reduce the potentially huge I/O overhead in scanning a database DB (that potentially can not be stored in memory) for discovering frequent patterns. Their goal consists of sampling the original database into a sample S and extract regularities from this subset while guaranteeing the accuracy of the extracted knowledge. Even if the sample S is here not drawn from an unknown underlying distribution \mathcal{D} (but rather from an existing large database DB), this framework looks like ours, especially since those Chernoff bounds can also be used to provide a theoretical size of S ensuring an upper bound of the estimate error. Indeed, let the observed frequency \hat{p} be the random variable X of Eq.1 computed from S , and p_{DB} its expected value $E(X)$ over the whole database DB (potentially large), the Chernoff bounds can be rewritten

as follows:

$$P(|\hat{p}_S - p_{DB}| \geq \epsilon) < e^{-2N\epsilon^2} \quad (2)$$

Ineq.2 can be used in different ways. First, given a size N , solving for ϵ this inequality equal to a given probability provides a slack value of the support threshold p_0 . On the other hand, given a value ϵ , solving for N Eq.2 equal to δ provides a lower bound of the sample size N satisfying the estimate error ϵ . Despite its obvious advantages, the use of the Chernoff bounds has a limitation. Indeed, as used in (19; 20), the symmetry due to the absolute value in Eq.2 indicates that the risk of a bad estimation \hat{p} is equally distributed around p_{DB} . In other words, the risk that a pattern occurs in S less often than expected in DB is equal to the risk that a pattern occurs more often than expected. In this context, Chernoff bounds does not allow the distinction between the false positive rate α and the false negative rate β , as defined in the introduction. This can be a problem in domains where α and β have to be independently handled. For instance, suppose that a vaccine is administered to a patient according to the frequent presence or not of a pattern in his DNA. Missing a patient who has the disease (*i.e.* overlooking a true frequent pattern) would not have the same medical effect than the one consisting of administering the vaccine to a healthy person (*i.e.* admit a false frequent pattern).

In (21), Toivonen presents another sampling method for discovering relevant association rules. The algorithm also picks a random sample S from the original database DB , then it determines from S all frequent associations rules that probably hold in DB ; finally it verifies with DB if they are actually frequent. To control the risk of overlooking true frequent patterns, Toivonen replaces the support threshold p_0 by a lower bound based on the Chernoff bounds so that misses are avoided with a high probability. However, Toivonen only deals with β . Indeed, by using DB to verify if the extracted patterns are actually frequent, the risk α of false positive is intrinsically null. However, this way of proceeding is only possible if the original database DB is available. While this condition is fulfilled in Toivonen's framework, it is an unacceptable constraint in ours which assumes that S has been drawn from an unknown theoretical distribution.

Recently, Laur et al. proposed in (22; 23) an approach that not only makes use of Chernoff bounds but also deals with both risks α and β . Given a sample S , they provide a bound for p_0 that ensures either a *precision* equal to 1 with a high probability while controlling the *recall*, or a *recall* equal to 1 with a high probability while limiting the degradation of the *precision*. Even if this approach is theoretically well founded, the user has to choose the criterion (*recall* or *precision*) he wants to optimize, that can be a tricky task in domains

where both errors α and β are definitely undesirable.

2.3 Modifying the support threshold p_0 using statistical tests

A second solution to check the relevance of a discovered pattern is to resort to statistical tests that involve two hypotheses, a *null hypothesis* H_0 and an alternative one H_a . Usually, H_a is made to describe an interesting situation (*e.g.* a frequent pattern), while H_0 characterizes the irrelevant situation (*e.g.* a non frequent pattern). When a test is performed, two types of errors can occur: The first one, called Type I error, comes from the acceptance of the hypothesis H_0 while H_a is true; the second one, called Type II error, corresponds to the wrong decision to accept H_a while H_0 is true. Therefore, adapted to the context of frequent pattern mining, the Type I error can be defined as describing the risk α of accepting a false frequent pattern, while the Type II error can be defined as being the risk β of rejecting a true frequent one. In this context, it is important to recall that there exists a statistical trade-off between these two risks. Given a sample size S , β actually increases if one wants to reduce α and vice versa. In the following, we present some state of the art approaches that deal with the relevance of extracted patterns using such statistical tests.

In (24), Megiddo & Srikant deal with the evaluation of the quality of association rules extracted from a set of data. They present an approach for estimating the number of false discoveries in order to control the *precision*. Let us consider an association rule $X \Rightarrow Y$, where X and Y are sets of items. As a null hypothesis, they assume that X and Y occur in the data independently. Thus, they test the null hypothesis $H_0 : p(X \cap Y) = p(X) \times p(Y)$ against the alternative one $H_a : p(X \cap Y) > p(X) \times p(Y)$, which, roughly speaking, means that a lot of transactions that contain X also contain Y . They run a statistical test exploiting the property that the observed frequency of an itemset asymptotically follows a normal distribution. To reduce the risk of accepting a false frequent pattern, they increase the support threshold p_0 by $z_\alpha \times \sigma_{\hat{p}}$, where $\sigma_{\hat{p}}$ is the standard deviation of \hat{p} and z_α is the $(1 - \alpha)$ percentile of the normal distribution. Therefore, by a priori tuning the risk α , they can control the *precision*. Nevertheless, by using a small value for α , bounding p_0 by this way results in the decrease of the *recall*.

Recently, in (25), Webb presents two new approaches to applying statistical tests in pattern discovery to assess the quality of a pattern. First, he suggests the split of the sample S into an *exploratory* set, from which a pattern extraction is achieved, and a *holdout* set used to assess the quality of each pattern. Despite promising experimental results, this approach is above all empirical and does not provide any bound that enables both risks to be reduced. Webb also presents an approach based on the Bonferroni adjustment

(26). When a statistical test is applied many times during an assessment, a special problem arises: if α corresponds to the risk of taking a wrong single decision, repeating the test many times globally increases that risk (26). To overcome this drawback, several strategies have been proposed (27). A famous one is the Bonferoni adjustment that uses a risk α/n when performing n hypothesis tests. However, if n is large, such adjustment turns out to be strict and leads to the increase of the other risk β .

Another solution consists of using Holm procedure (28) that takes into account the *p-value* of each test and orders them to tune a less strict risk. Such a strategy is also used in the BH procedure (29) that aims to set α while controlling the so-called *false discovery rate*. However, both of these adjustments require the computation of the *p-values* of the n tests which depend on the current application. In our paper, we will provide a more general tool whatever the application we deal with. Moreover, note that we aim to determine a relationship between the **number N of data** to mine and fixed risks α and β . In the adjustment procedures mentioned before, α and β are linked to the **number n of statistical tests** when testing multiple hypotheses. Therefore, both objectives cannot be directly connected.

In (30), Lee et al. present the DELI algorithm which is based again on a sampling method which generates a sample S from a database DB . To maintain in S an accurate set of association rules, a confidence interval is built for the true probability p of an association rule in DB , such that $p \in \hat{p} \pm z_{\alpha/2} \sqrt{\frac{\hat{p}(|DB| - \hat{p})}{|S|}}$, where \hat{p} is the support of the rule in S , α is the Type I error, and z_{α} is the $(1 - \alpha)$ percentile of the normal distribution. By fixing α , the authors show that one can determine a suitable size of S satisfying the Type I error. As we can note, this approach has two main drawbacks. First, only the Type I error α is used to assess the statistical significance of the patterns. Therefore, the size of S deduced from the confidence interval does not take into account the Type II error β . On the other hand, the computation of this interval requires the use of the size of the original database DB . As we mentioned before, our more general framework does not require to have DB .

Finally, note that other statistical test-based investigations have dealt with the assessment of the significance of patterns in data mining. They use efficient tests (such as the Chi-square test and Fisher exact test) to statistically measure the level of dependency between the components of a pattern. An often used strategy consists of verifying if the extracted structure would also be discovered from a random sample having same margins (see (31) for example).

The approaches we presented in this survey either impose a symmetry condition on the estimate error, or minimize only one risk given a sample set size, or require the calculation of *p-values* of a specific set of statistical tests. No one offers theoretical results that provide a bound on the size of S satisfy-

ing arbitrary chosen parameters α and β . In the following, we fill this gap by proposing a statistical approach that exploits the asymptotic convergence of the distribution of frequent patterns. We provide a bound on N , easily computable, allowing the independent control of both risks α and β .

3 A statistical view of the recall and the precision

3.1 Risks of rejecting true frequent patterns and accepting false ones

Let $\hat{p}(w)$ be the proportion of data in the set S that contain a given pattern w . Let us recall that w is called *frequent* if $\hat{p}(w)$ is higher than a *minimal support threshold* p_0 . In fact, $\hat{p}(w)$ is nothing else but an estimate of the real probability $p(w)$ over \mathcal{D} . Since $p(w)$ is unknown, one can formulate a hypothesis on its real value and perform a statistical test. As usually done in the standard approaches, we suggest to describe by the null hypothesis H_0 the situation where $\hat{p}(w)$ is not high enough to consider w as being frequent. As done in (24), we suggest to keep the maximal value p_0 that prevent w from being accepted as frequent. Therefore, we test the null hypothesis $H_0: p(w) = p_0$, against the alternative one H_a , which describes an interesting discovery, *i.e.* $H_a: p(w) > p_0$.

Type I error: α represents the risk of rejecting H_0 while it is true. In our frequent pattern mining context, α corresponds to the risk of *accepting a false frequent pattern*. Therefore, $1 - \alpha$ exactly describes the theoretical *precision* of the algorithm over the distribution \mathcal{D} . For instance, with a support threshold p_0 of 10%, observing $\hat{p}(w) = 10.2\%$ in S does not mean that w is definitely frequent in the target distribution \mathcal{D} . To be able to take a well-founded decision, we can a priori fix α (usually 5%, but it can depend on the application we deal with), and then compute a bound of rejection k , satisfying α . More formally,

$$\alpha = P(\hat{p}(w) > k | H_0 \text{ true}). \quad (3)$$

The number of data of S that contain w is a binomial random variable with success probability $p(w)$. According to the size N of S and the support threshold p_0 , we can use either the normal or the Poisson approximation. In our context, we aim to provide a theoretical bound on N that will be by nature quite large. Moreover, since we are looking for frequent patterns, we can assume that p_0 will be chosen sufficiently large otherwise the framework would be the one of exceptions or rare events that is the matter of another research domain. Therefore, using the central limit theorem, we will consider in the

following that the proportion $\hat{p}(w)$ follows a normal distribution \mathcal{N} , such that

$$\hat{p}(w) \approx \mathcal{N} \left(p(w), \sqrt{\frac{p(w)(1-p(w))}{N}} \right).$$

Equation 3 can be rewritten

$$\alpha = P \left(\frac{\hat{p}(w) - p(w)}{\sqrt{\frac{p(w)(1-p(w))}{N}}} > \frac{k - p(w)}{\sqrt{\frac{p(w)(1-p(w))}{N}}} | H_0 \text{ true} \right). \quad (4)$$

Since H_0 is true, we have to replace $p(w)$ by its value under H_0 . We get

$$\alpha = P \left(\frac{\hat{p}(w) - p_0}{\sqrt{\frac{p_0(1-p_0)}{N}}} > \frac{k - p_0}{\sqrt{\frac{p_0(1-p_0)}{N}}} \right). \quad (5)$$

We can then easily deduce the bound k which corresponds to the $(1 - \alpha)$ -percentile z_α of the normal distribution:

$$k = p_0 + z_\alpha \sqrt{\frac{p_0(1-p_0)}{N}}. \quad (6)$$

To recap, by fixing a risk α , Equation 6 gives us the bound of rejection of H_0 . For example, let us suppose we are mining $N = 10000$ data. Let us fix the support threshold $p_0 = 10\%$ and the risk $\alpha = 5\%$ ($z_\alpha = 1.645$ by reading the table of the normal distribution). Plugging these values in Equation 6, we get $k = 0.1 + 1.645 \times \sqrt{\frac{0.1 \cdot 0.9}{10000}} = 0.105$. Therefore, a pattern w with a support $\hat{p}(w) = 10.2\%$ will be in fact rejected in order to control the risk of accepting false positives.

3.1.0.1 Type II error β : Regarding β , it describes the probability to *reject a true frequent pattern*. In contrast to α , β can be calculated according to the previously computed bound k . Since $H_a: p(w) > p_0$ is true, we have to set a given value for $p(w)$ satisfying the constraint $p(w) > p_0$. Let p_a be this value (see Section 5 for a discussion about p_a). We get

$$\beta = P(\hat{p}(w) < k | H_a \text{ true}). \quad (7)$$

As previously done for α ,

$$\beta = P \left(\frac{\hat{p}(w) - p(w)}{\sqrt{\frac{p(w)(1-p(w))}{N}}} < \frac{k - p(w)}{\sqrt{\frac{p(w)(1-p(w))}{N}}} \mid H_a \text{ true} \right). \quad (8)$$

By replacing $p(w)$ by its value under H_a , we get

$$\beta = P \left(\frac{\hat{p}(w) - p_a}{\sqrt{\frac{p_a(1-p_a)}{N}}} < \frac{k - p_a}{\sqrt{\frac{p_a(1-p_a)}{N}}} \right). \quad (9)$$

Since k is known thanks to Eq. 6, the $(1 - \beta)$ -percentile z_β is also known, and β can be easily deduced from the normal distribution. To continue with our previous example (assuming that $N = 10000$ data), let us suppose that $p_a = 11\%$, then $\beta = 5.5\%$ (by reading the table of the normal distribution). Therefore, for a true support of 11%, the probability to falsely accept the null hypothesis based on a finite sample of $N = 10000$ data is 5.5%.

3.2 Lower bound on N

The ideal objective of a frequent pattern mining process is to reduce not only α but also β . However, as mentioned before, there exists a trade-off between these two risks. With a **constant** number of data N , β actually increases if one reduces α and vice versa. A solution to overcome this drawback consists of determining how many data N would be needed to not exceed a priori fixed α and β risks. This is the matter of the next theorem.

Theorem 1 *To ensure that the false positive rate and the false negative rate do not exceed respectively fixed risks α and β , the lower bound N_{low} of the size of the sample S on which the frequent pattern mining algorithm must be run is equal to*

$$N_{low} = \left[\frac{z_\beta \sqrt{p_a(1-p_a)} + z_\alpha \sqrt{p_0(1-p_0)}}{p_a - p_0} \right]^2, \quad 0 < p_0 < p_a < 1.$$

Proof 1 *The proof is straightforward. We can deduce from Equation 9 that*

$$k = p_a - z_\beta \sqrt{\frac{p_a(1-p_a)}{N}}, \quad (10)$$

where z_β is the $(1 - \beta)$ -percentile of the normal distribution. Equating Equations 6 to 10, we can deduce that

$$p_0 + z_\alpha \sqrt{\frac{p_0(1-p_0)}{N}} = p_a - z_\beta \sqrt{\frac{p_a(1-p_a)}{N}}. \quad (11)$$

Extracting N from Equation 11, we obtain the lower bound. \square

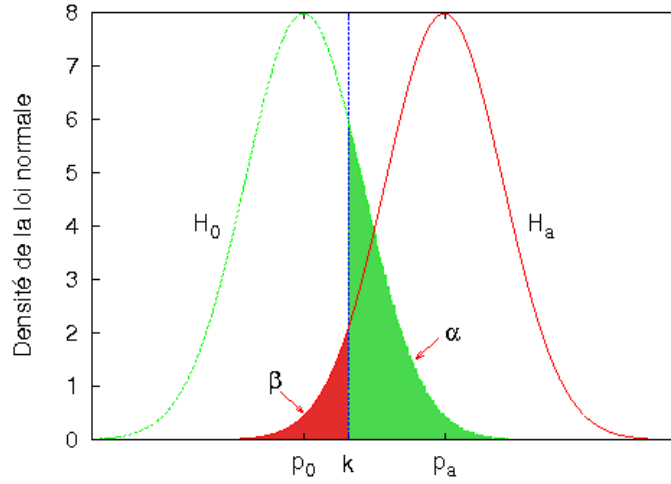


Fig. 1. Trade-off between Type I (light grey area) and Type II errors (dark grey area). p_0 (resp. p_a) is the expectation of $\hat{p}(w)$ under H_0 (resp. H_a).

Let us now describe the meaning of this bound. It is important to note that there is a direct relationship between β and p_a given a fixed number of data. Indeed, as described in Figure 1, p_a is the expectation of $\hat{p}(w)$ under the alternative hypothesis H_a . β corresponds to the density of the normal distribution beneath the bound k of rejection of H_0 . Therefore, the farther p_a is from p_0 , the lower the risk β . Since β and p_a are parameters in our lower bound, reducing both implies an increase of the needed number of data. The same remark can be done between α and β . Reducing α for a given size N implies the increase of β . Therefore, reducing both risks results in the increase of the required number of data.

To illustrate this lower bound, the chart of Figure 2 shows the evolution of N_{low} according to α , β , p_0 and p_a . For the sake of legibility we choose $\alpha = \beta$. We plot two curves with two different values of p_a . We can note that the smaller $p_a - p_0$, the larger the lower bound. A further discussion about the valuation of p_a is presented in Section 5.

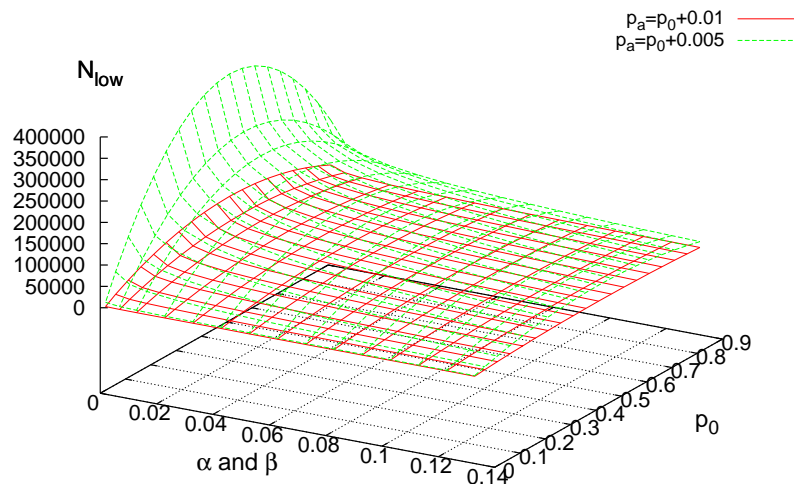


Fig. 2. N_{low} according to α , β , p_0 and p_a .

4 Illustration on a real world sequence mining task

Let us illustrate the impact of our bound in a real world sequence mining task. We carry out a series of experiments on the ATIS (Air Travel Information Service) corpus. This database consists of travel information requests performed in english. We have an original set Ω of 14044 sentences from which we draw samples S_i of increasing size $|S_i|$ (from 10 to 14044) and we extract frequent patterns with a support threshold of 10% with SPAM (32) which is a well-known sequence mining tool. In this series of experiments, to allow the analysis of the behavior of our bound, we assume that Ω represents the theoretical underlying distribution \mathcal{D} from which the samples S_i have been drawn. In order to assess the effect of the size $|S_i|$ on the quality of the extracted knowledge, we have to be able to measure the empirical values of α and β , that we will call $\hat{\alpha}$ and $\hat{\beta}$. $\hat{\alpha}$ is the observed proportion of patterns that have been extracted as frequent from S_i while they are not frequent in the target population Ω . $\hat{\beta}$ corresponds to the observed proportion of patterns that are frequent in Ω but overlooked from S_i .

Figure 3 describes, according to an increasing size $|S_i|$ of the sample set S_i and a support threshold $p_0 = 10\%$, the evolution of $1 - \hat{\alpha}$ and $1 - \hat{\beta}$, using a value $p_a = 11\%$. Note that we performed 15 trials, for each size $|S_i|$, and we computed the average in order to reduce the variance of the results. As expected, the higher the number of sequences, the smaller the computed risks $\hat{\alpha}$ and $\hat{\beta}$. We can also note that for small sizes of S_i (< 1000) both risks $\hat{\alpha}$ and $\hat{\beta}$ are high ($> 10\%$) meaning that a lot of extracted patterns are not truly frequent in Ω and many others have been overlooked. This example is a good

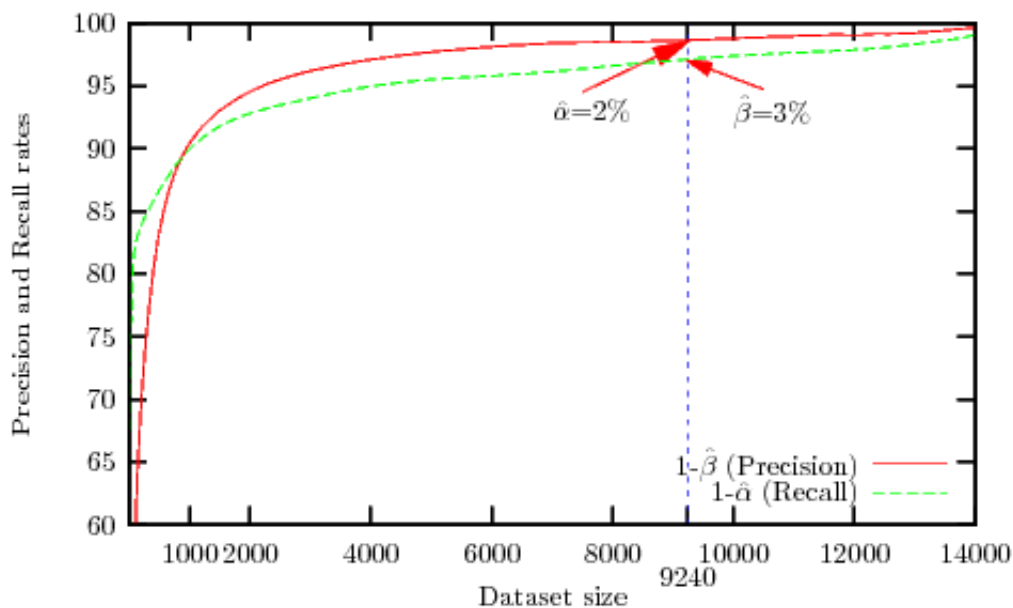


Fig. 3. Evolution of the quality of the results of a sequence mining algorithm according to an increasing size of S_i .

illustration of the bottleneck of standard mining algorithms. Since $\hat{\alpha}$ and $\hat{\beta}$ can be empirically measured, they can be compared with theoretical risks α and β to verify the relevance of our bound. To achieve this task, let us compute N_{low} for given theoretical parameters α , β , p_a and p_0 . For instance, let us set $\alpha = \beta = 5\%$ ($p_a = 10\%$ and $p_a = 11\%$ being already fixed). Plugging these values in our bound yields the value $N_{low} = 10165$. If we observe from Figure 3 the results obtained from 10165 sequences, we can conclude that our bound is relevant because the two observed errors computed on the ATIS database ($\hat{\alpha} = 3\%$ and $\hat{\beta} = 2\%$) actually do not exceed our a priori fixed theoretical risks α and β .

Note that the difference between the observed and the theoretical errors can appear quite substantial on this experiment even if it is on the “safe side”. In fact, the distance between the observed and the theoretical errors directly depends on the sample S_i drawn from the unknown target distribution. But since N_{low} constitutes a lower bound needed, *in the worst case*, to satisfy α and β , our theorem states that we never fall on the “unsafe side”.

5 What about the value of p_a ?

So far, to illustrate our bound, we used a value of p_a “close” to p_0 under the alternative hypothesis H_a . As we explained in the previous section, there is

a strong relationship between p_a and our lower bound N_{low} . More precisely, N_{low} quadratically increases with the drop of the difference between p_a and p_0 . Therefore, the choice of a relevant value p_a remains an important problem that deserves special attention. In statistical inference, it is often states that the valuation of the parameters under H_a has to be fixed according to the considered application. To avoid to be dependent on this application, we study in the following of this section two theoretical ways to set the value of p_a .

5.1 A worst case solution

The first solution to tackle the problem of the valuation of p_a is to consider that a pattern w is truly frequent from the moment that its probability $p(w)$ over \mathcal{D} is greater than the support threshold p_0 . Let N_0 be the number of data such that $\frac{N_0}{N_{low}} = p_0$. Therefore, a pattern w is truly frequent if it occurs at most $N_0 + 1$ times in the N_{low} data. So, we get that

$$p_a = \frac{N_0 + 1}{N_{low}} = p_0 + \frac{1}{N_{low}}.$$

Plugging this value in Equation 10, and equating Equations 6 to 10, we get the following analytical representation of our lower bound, in a polynomial form of order 4 whose solution gives N_{low} (this polynomial has been obtained using MAPPLETM):

$$\begin{aligned} & (-2z_\beta^2 z_\alpha^2 p_0^2 + z_\beta^4 p_0^4 + 4z_\beta^2 p_0^3 z_\alpha^2 + z_\beta^4 p_0^2 - 2z_\beta^2 p_0^4 z_\alpha^2 + z_\alpha^4 p_0^4 + z_\alpha^4 p_0^2 - 2z_\alpha^4 p_0^3 - 2z_\beta^4 p_0^3) \mathbf{N}^4 \\ + & (2p_0(-z_\alpha^2 - z_\beta^4 + z_\beta^2 p_0 - z_\beta^2 + z_\alpha^2 z_\beta^2 + z_\alpha^2 p_0) + 4p_0^3(z_\beta^2 z_\alpha^2 - z_\beta^4) + 6p_0^2(z_\beta^4 - z_\beta^2 z_\alpha^2)) \mathbf{N}^3 \\ + & (1 - 4z_\beta^2 p_0 + 2p_0 z_\alpha^2 z_\beta^2 + 2z_\beta^2 - 2z_\beta^2 z_\alpha^2 p_0^2 + z_\beta^4 - 6z_\beta^4 p_0 + 6z_\beta^4 p_0^2) \mathbf{N}^2 \\ + & (2z_\beta^4 + 2z_\beta^2 - 4z_\beta^4 p_0) \mathbf{N} + z_\beta^4 = 0. \end{aligned}$$

We can see that the lower bound now only depends on the risks α and β , and the support threshold p_0 . p_a is no longer a parameter of our bound, and therefore the solution of this equation provides the exact lower bound guaranteeing at worst α and β given a support threshold p_0 . Nevertheless, this solution constitutes a very pessimistic answer to our problem. For instance, solving this equation setting $\alpha = \beta = 5\%$ and $p_0 = 0.1$, we get $N_{low} = 3.10^{20}$ sequences!

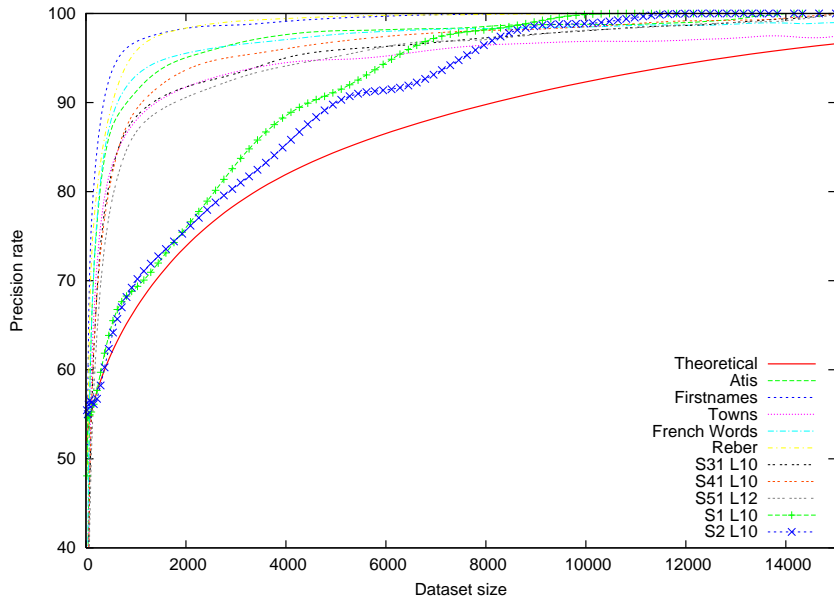


Fig. 4. Comparison between various empirical *precisions* and $1 - \alpha$, when $p_0 = 0.1$.

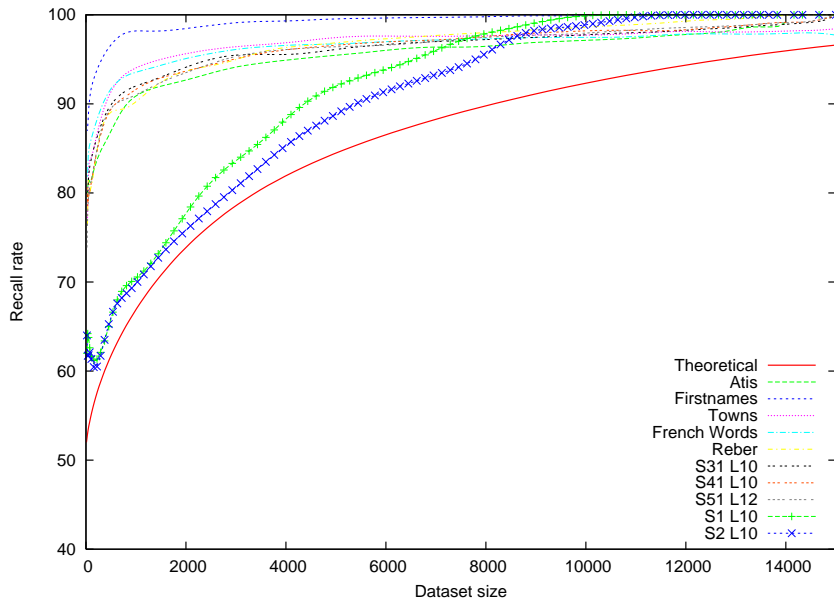


Fig. 5. Comparison between various empirical *recalls* and $1 - \beta$, when $p_0 = 0.1$.

5.2 An average solution

In the previous solution, we assumed that *all the patterns* that have been overlooked followed a normal distribution of expected value $p_a = p_0 + \frac{1}{N_{low}}$, which is actually the worst situation. In practice, each omitted frequent pattern has its own theoretical support p_a that can belong to the interval $]p_0, 1]$. How can we take into account those different possible values of p_a in our bound? We suggest in the following the computation of an average solution \bar{N}_{low} which is the expected value of N_{low} over $]p_0, 1]$, such that:

$$\bar{N}_{low} = \frac{1}{1 - p_0} \times \int_{p_0}^1 N_{low} dp_a. \quad (12)$$

This expected value does not depend on p_a anymore. To assess the relevance of this strategy, we computed for different values of α and β (for the sake of simplicity we set $\alpha = \beta$) the expected value \bar{N}_{low} using Eq.12. These theoretical results are described in Figures 4 and 5 in the form of two curves in solid line. They are compared with various curves of empirical *recall* and *precision* computed from 10 different datasets. Four of them are real databases: ATIS which has already been used in this paper, and three other databases available at the URL <http://abu.cnam.fr/>. FIRSTNAMES is a set of 12437 male and female first names of different origins; TOWNS is a set of 39074 names of french towns; FRENCH WORDS is a set of 250750 french words. We also built 6 artificial databases from probabilistic automata: *Reber* is a set of 15000 sequences generated from the Reber grammar (33) whose target distribution is an automaton constituted of 8 states and an alphabet of 7 letters; We generated 5 other sets of 15000 sequences from 5 automata, each one composed of x states and an alphabet of y letters and denoted $SxLy$ (see Fig.6 for an example of an automaton $S2L2$). Note that such an automaton constitutes a theoretical distribution \mathcal{D} from which it is possible to compute the probability $p(w)$ of any pattern w , using suitable calculation methods (see (34) for example).

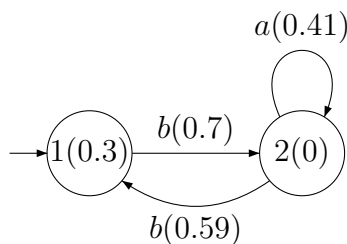


Fig. 6. Automaton $S2L2$ viewed as a target distribution \mathcal{D} .

For each of the databases, we compute with SPAM the set of frequent patterns with a support threshold of 10%. This set will constitute the target distribution. Then we sample sets of growing size (from 10 to 15000) from which we also extract frequent patterns, and we calculate the empirical *precision* ($1 - \hat{\alpha}$) and *recall* ($1 - \hat{\beta}$). The results are shown respectively in Figure 4 and Figure 5, and have to be compared with the curves in solid line of those figures. They confirm that we actually provide a relevant lower bound on the number of sequences needed to at least guarantee a priori fixed theoretical *recall* and *precision*. Whatever the database, its corresponding curves ($1 - \hat{\alpha}$ or $1 - \hat{\beta}$) are always over the theoretical one. Note that the distance between the empirical risks and our lower bound is quite large for some databases, meaning that our bound can remain quite pessimistic. However, it does not challenge its relevance since, as shown with the curves obtained from the automata $S1L10$

and *S2L10*, it may happen that the empirical risks, due to specific sampling effects, are much more close to the theoretical ones.

Note that the theoretical curves described in Figures 4 and 5 only tackle the case of $p_0 = 0.1$. In order to provide a calculating tool that would make the estimation of the minimal number of data easier, we built the theoretical curves for different values of p_0 . Figure 7 describes a set of abaci that directly provide the lower bound \hat{N}_{low} required to guarantee at least a *precision* of $(1 - \alpha)$ and a *recall* of $(1 - \beta)$ (once again, for the sake of simplicity, we set $\alpha = \beta$). We can note that the curves are not the same, that means that the value of p_0 has a direct impact on the lower bound. From a mathematical point of view, this can be easily explained by the fact that p_0 is used in the formulae of N_{low} (see Theorem 1) within a concave function in the form of $p_0(1 - p_0)$ which is maximal for $p_0 = 0.5$. Therefore, for the same values α and β , setting $p_0 = 0.5$ requires more data than for other values. This explains the fact that the curve for $p_0 = 0.5$ is under the others. Therefore, from a statistical point of view, to avoid having large risks α and β , a good strategy consists of choosing a **support threshold p_0 far from 0.5**.

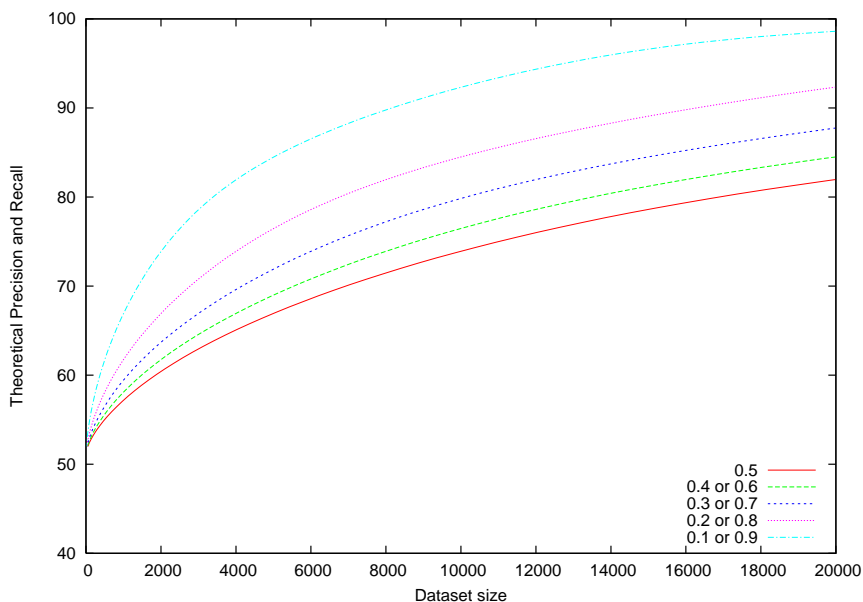


Fig. 7. Evolution of the theoretical *recall* and *precision* according to the lower bound N_{low} and the support threshold p_0 (from 0.1 to 0.9).

6 Conclusion and future work

In this paper, we dealt with the assessment of the significance of a frequent pattern mining process. To perform this task, we presented a lower bound on the number of data required to satisfy theoretical *precision* and *recall*. As far as we know, this constitutes the first attempt to control both criteria by

providing a condition on the number of data we have to deal with. Despite its theoretical nature, we showed that our bound can be very useful in real world data mining applications. We empirically tested our bound in the specific case of sequence mining tasks. However, our work can be adapted to other data mining contexts that require a comparison to a minimal support threshold.

Throughout this paper, we wanted to stay in a theoretical framework in order to avoid a dependence on the application we deal with. This explains why we did not use any information about the sample set S . In the future, we plan to reduce the pessimism of our theoretical bound by integrating background knowledge during the computation of the bound. One solution would consist in using the empirical distribution of the patterns in S to weight each value used in the computation of the integral in Eq.12. However, this deserves further investigations. Indeed, once again, such an empirical distribution is dependent on a finite sample set whose size must be integrated in the model to avoid to have bad estimates.

Finally, note also that our theorem can also constitute a good condition to fulfill in various machine learning domains. Actually, since building a set of N_{low} data allows us to have a good estimate of any pattern w , it also enables us to correctly estimate the probability of any n-gram, which is a special case of pattern. Since n-grams are used in many probabilistic models in machine learning, such as probabilistic automata, stochastic transducers, or Hidden Markov models, we think that N_{low} can constitute a good lower bound to efficiently learn such stochastic models.

References

- [1] B. Goethals, M. J. Zaki, Advances in frequent itemset mining implementations: report on fimi'03, SIGKDD Explorations 6 (1) (2004) 109–117.
- [2] J. Han, H. Cheng, D. Xin, X. Yan, Frequent pattern mining: current status and future directions, Data Mining and Knowledge Discovery 15 (1) (2007) 55–86.
- [3] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: Proceedings of 20th International Conference on Very Large Data Bases, Morgan Kaufmann, 1994, pp. 487–499.
- [4] H. Mannila, H. Toivonen, A. I. Verkamo, Efficient algorithms for discovering association rules, in: KDD Workshop, 1994, pp. 181–192.
- [5] A. Ceglar, J. F. Roddick, Association mining, ACM Computing Surveys 38 (2) (2006) .
- [6] H. Mannila, H. Toivonen, A. I. Verkamo, Discovery of frequent episodes in event sequences, Data Mining and Knowledge Discovery 1 (3) (1997) 259–289.

- [7] M. Zaki, Sequence mining in categorical domains: incorporating constraints, in: Proceedings of the 9th international conference on information and knowledge management, ACM Press, 2000, pp. 422–429.
- [8] M. Garofalakis, R. Rastogi, K. Shim, Mining sequential patterns with regular expression constraints, *IEEE Transactions on Knowledge and Data Engineering* 14 (3) (2002) 530–552.
- [9] J. Pei, J. Han, W. Wang, Mining sequential patterns with constraints in large databases, in: Proceedings of the 11th international conference on information and knowledge management, ACM Press, 2002, pp. 18–25.
- [10] D. Jiang, J. Pei, Mining frequent cross-graph quasi-cliques, *ACM Transactions on Knowledge Discovery from Data* 2 (4) (2009) 1–42.
- [11] P. Hintsanen, H. Toivonen, Finding reliable subgraphs from large probabilistic graphs, *Data Mining and Knowledge Discovery* 17 (1) (2008) 3–23.
- [12] D. Chakrabarti, C. Faloutsos, Graph mining: Laws, generators, and algorithms, *ACM Computing Surveys* 38 (1) (2006) .
- [13] L. Getoor, C. P. Diehl, Link mining: a survey, *SIGKDD Explorations* 7 (2) (2005) 3–12.
- [14] Y. Chi, R. R. Muntz, S. Nijssen, J. N. Kok, Frequent subtree mining - an overview, *Fundamenta Informaticae* 66 (1-2) (2005) 161–198.
- [15] M. J. Zaki, Efficiently mining frequent embedded unordered trees, *Fundamenta Informaticae* 66 (1-2) (2005) 33–52.
- [16] A. Termier, M.-C. Rousset, M. Sebag, K. Ohara, T. Washio, H. Motoda, Dryadeparent, an efficient and robust closed attribute tree mining algorithm, *IEEE Transaction on Knowledge and Data Engineering* 20 (3) (2008) 300–320.
- [17] V. Chaoji, M. A. Hasan, S. Salem, M. J. Zaki, An integrated, generic approach to pattern mining: data mining template library, *Data Mining and Knowledge Discovery* 17 (3) (2008) 457–495.
- [18] J. Han, R. B. Altman, V. Kumar, H. Mannila, D. Pregibon, Emerging scientific applications in data mining, *Communications of the ACM* 45 (8) (2002) 54–58.
- [19] M. Zaki, S. Parthasarathy, W. Li, M. Ogihara, Evaluation of sampling for data mining of association rules, in: In 7th International Workshop Research Issues in Data Engineering, 1997, pp. 42–50.
- [20] C. Raissi, P. Poncelet, Sampling for sequential pattern mining: from static databases to data streams, *IEEE International Conference on Data Mining* (2007) 631–636.
- [21] H. Toivonen, Sampling large databases for association rules, in: Proceedings of the 22th International Conference on Very Large Data Bases, Morgan Kaufmann, 1996, pp. 134–145.
- [22] P.-A. Laur, R. Nock, J.-E. Symphor, P. Poncelet, Mining evolving data streams for frequent patterns, *Pattern Recognition* 40 (2) (2007) 492–503.
- [23] P.-A. Laur, J.-E. Symphor, R. Nock, P. Poncelet, Statistical supports for mining sequential patterns and improving the incremental update process

- on data streams, *Intelligent Data Analysis* 11 (1) (2007) 29–47.
- [24] N. Megiddo, R. Srikant, Discovering predictive association rules, in: *Proceedings of the 4th international conference of knowledge discovery and data mining*, 1998, pp. 274–278.
 - [25] G. I. Webb, Discovering significant patterns, *Machine Learning* 68 (1) (2007) 1–33.
 - [26] J. P. Shaffer, Multiple hypothesis-testing, *Annual review of psychology* 46 (1995) 561–584.
 - [27] G. Gavin, S. Gelly, Y. Guermeur, S. Lallich, J. Mary, M. Sebag, O. Teytaud, Type 1 and type 2 errors for multiple simultaneous hypothesis testing. <http://www.lri.fr/~teytaud/risq/>, *PASCAL theoretical challenge* (2007) 29–47.
 - [28] S. Holm, A simple sequentially rejective multiple test procedure, *Scandinavian Journal of Statistics* 6 (1979) 65–70.
 - [29] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a new and powerful approach to multiple testing, *Journal of the Royal Statistical Society Series B* 57 (1995) 289–300.
 - [30] S. D. Lee, D. W. Cheung, B. Kao, Is sampling useful in data mining? a case in the maintenance of discovered association rules, *Data Mining and Knowledge Discovery* 2 (3) (1998) 233–262.
 - [31] A. Gionis, H. Mannila, T. Mielikäinen, P. Tsaparas, Assessing data mining results via swap randomization, *Transactions on Knowledge Discovery and Data* 1 (3) (2007) .
 - [32] J. Ayres, J. Flannick, J. Gehrke, T. Yiu, Sequential pattern mining using a bitmap representation, in: *Proceedings of the 8th international conference on knowledge discovery and data mining*, ACM Press, 2002, pp. 429–435.
 - [33] A. S. Reber, Implicit learning of artificial grammars, *Journal of Verbal Learning and Verbal Behavior* 6 (1967) 855–863.
 - [34] P. Hingston, Using finite state automata for sequence mining, in: *Proceedings of the 25th Australasian conference on computer science*, Australian Computer Society, Inc., 2002, pp. 105–110.