ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT

Pattern Recognition ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

PATTERN RECOGNITION

THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Mining evolving data streams for frequent patterns☆

Pierre-Alain Laur[a], Richard Nock[a,*], Jean-Emile Symphor[a], Pascal Poncelet[b]

[a]Grimaag Département Scientifique Interfacultaire, Université des Antilles-Guyane, Campus de Schoelcher, BP 7209, 97275 Schoelcher, Martinique, France

[b]Ecole des Mines d'Alès, LG2IP/Site EERIE, Parc Scientifique Georges Besse, 30035 Nı̂mes cedex 1, France

Received 7 November 2005; accepted 13 March 2006

Abstract

A data stream is a potentially uninterrupted flow of data. Mining this flow makes it necessary to cope with uncertainty, as only a part of the stream can be stored. In this paper, we evaluate a statistical technique which biases the estimation of the support of patterns, so as to maximize either the precision or the recall, as chosen by the user, and limit the degradation of the other criterion. Theoretical results show that the technique is not far from the optimum, from the statistical standpoint. Experiments performed tend to demonstrate its potential, as it remains robust even under significant distribution drifts.

Keywords: Data streams; Concentration inequalities; Precision; Recall; Accuracy

1. Introduction

A growing body of works arising from Databases and Data Mining deals with data arriving in the form of continuous potentially infinite streams, i.e. an ordered sequence of item occurrences that arrives in a timely manner. Data streams have seen the emergence of crucial problems that were previously not as pregnant for databases, such as the accurate retrieval of informations in a data flow that prevents its exact storage, and whose information may evolve through time. Emerging and real applications generate data streams: trend analysis, fraud detection, intrusion detection, click stream, among many others. Trend analysis is an important problem that commercial applications have to deal with, which is to detect in the data stream significant trends, emerging buzz, and unusually high or low activity [1]. In fraud detection, data miners try to detect suspicious changes in user behavior [2]. Finally, intrusion detection is a critical approach to help protect systems, with the growing importance of network systems security and the sensitivity of the informations stored and manipulated online [3].

A crucial issue in Data Mining that has recently attracted significant attention [3–8] is to build the set of the most frequent patterns encountered in the data stream. Though it is straightforward to formulate, addressing this issue faces two non-trivial problems. The first is the statistical approximation of the true supports by observed supports. The second concerns the drifts that the data stream may face through time.

The rest of this paper is organized as follows. Section 2 states precisely the problem. Our theoretical approach is presented and discussed in Section 3. Section 4 is experimental: it presents and discusses some results that were obtained on readily generable data streams. In Section 5 we make some comparisons with related approaches. Finally, Section 6 concludes the paper with future avenues for research. In order not to laden the paper, an Appendix at the end of the paper contains the proof of a theorem.

☆ Paper also available at http://www.univ-ag.fr/~rnock/Articles/PR06/

* Corresponding author. Fax: +596 596 72 73 62.

E-mail addresses: Pierre-Alain.Laur@martinique.univ-ag.fr (P.-A. Laur), Richard.Nock@martinique.univ-ag.fr (R. Nock), Jean-Emile.Symphor@martinique.univ-ag.fr (J.-E. Symphor), Pascal.Poncelet@ema.fr (P. Poncelet)

URL: http://www.univ-ag.fr/~rnock (R. Nock).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2006.03.006

2. Problem statement

We define *items* as the unit information, *itemsets* to be sets of items [9], and *sequential patterns* to be sequences of items [10]. We use the word *pattern* for a shorthand to both settings, without loss of generality. A pattern is θ-*frequent* if it occurs in at least a fraction θ of the data stream (called its support), where θ is a user-specified parameter.

Basically, our problem is motivated by the fact that the data we store catches a glimpse of a data stream, and the information we mine should take into account the uncertainty generated by this *partial* observation of the whole stream. Our setting is thus a bit more downstream than those of [5,11–14]. Given the nature of the streaming data, there are two sources of error when estimating frequent patterns from the available part of the stream:

(1) it is possible that some patterns observed as frequent might in fact not be frequent anymore from a longer observation of the data stream;
(2) on the other hand, some patterns observed as not frequent may well in fact be frequent from a longer history of the data stream.

The point is that it is statistically hard to nullify both sources of error from the observation of a *subset*, even very large, of the whole data stream [15]. This unsatisfiable goal can be relaxed to the tight control of one source or error, while keeping the other one within reasonable bounds. This goal, which we address in this paper, can be summarized as follows; the user fixes some related parameters and chooses a source of error:

(a) the source of error chosen is nullified with high probability;
(b) the other one incurs a limited loss.

In this paper, we propose a solution to this problem which is statistically near optimal: any other technique that would yield a loss significantly smaller on (b) would not satisfy (a), regardless of its computation time.

Another problem regarding data streams is the *robustness* of the technique, when the stream is subject to distribution drifts. In this case, pattern supports may fluctuate, and mining is as efficient as it makes a fast tracking and update of the frequent patterns.

3. Our approach

Our approach relies on the following model of the data stream. It is supposed to be obtained from the repetitive sampling of a potentially huge *domain* X which contains all possible data sequences, see Fig. 1 (a). Obviously, X is unknown, but we have access to its elements through an unknown distribution $\mathscr{D}$, see Fig. 1(b). We make absolutely

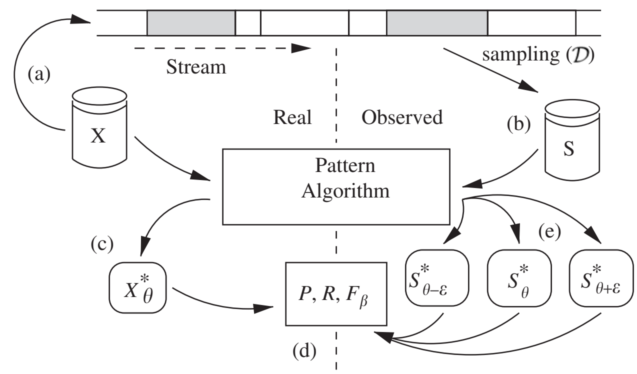

Fig. 1. Our framework. The left hand-side depicts the reality, and the right-hand side what we "see" from the sampling of the stream (see text for details).

no assumption on $\mathscr{D}$, except for the moment that it does not change through time (later, this assumption shall be relaxed). Now, the user specifies a real $0 < \theta < 1$, the *theoretical* support, and ideally wishes to recover all the patterns of X that are θ-frequent with respect to $\mathscr{D}$ (also called *true* θ-frequent). This set is called X_θ, and formally defined below.

Definition 1.

$$\forall 0 \leqslant \theta \leqslant 1, \quad X_\theta = \{T \in X : \rho_X(T) \geqslant \theta\}, \tag{1}$$

with $\rho_X(T) = \sum_{T' \in X : T \leqslant_t T'} \mathscr{D}(T')$, and $T \leqslant_t T'$ means that T generalizes T'.

Ideally, our objective should be to approximate X_θ. However, X is typically huge and the set S of observed data sequences which we have sampled from X in the data stream, has a size $|S| = m$ which is typically of minute order with respect to $|X|$. In our framework, we usually reduce this difference with some algorithm returning a superset S^* of S, having size $|S^*| = m^* > m$. Typically, S^* contains additional generalizations of the elements of S [16]. The key point is that S^* is usually still not large enough to cover X_θ, regardless of the way it is built (see Fig. 2). We can thus relax our objective to solve the following affordable estimation problem:

(**Pb1**) approximate as best as possible the following set:

$$X^*_\theta = X_\theta \cap S^*, \tag{2}$$

for any S and S^* (see Figs. 1 (c) and 2).

Now, $\forall T \in S^*$, we cannot compute exactly $\rho_X(T)$, since we do not know X and $\mathscr{D}$. Rather, we have access to its best unbiased estimator $\rho_S(T)$, which can be easily computed from S: $\forall T \in S^*, \rho_S(T) = \sum_{T' \in S : T \leqslant_t T'} w(T')$, with $w(T')$ the weight (observed frequency) of T' in S. We adopt the following approach to solve problem (**Pb1**):

(**Pb2**) find some $0 < \theta' < 1$ and approximate the set X^*_θ by the set of *observed* θ'-frequent of S^*, that is:

$$S^*_{\theta'} = \{T \in S^* : \rho_S(T) \geqslant \theta'\}. \tag{3}$$

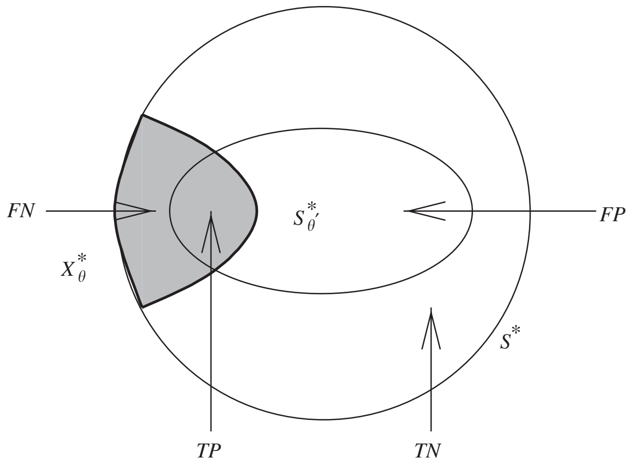

Fig. 2. The error estimation: the set we build, $S^*_{\theta'}$, may suffer two sources of error from X^*_θ (see text for details).

Addressing (**Pb2**) amounts to fixing an accurate value for θ'. Clearly, the naive approach fixing $\theta' = \theta$ does not bring $X^*_\theta = S^*_{\theta'}$; it only guarantees that this holds with probability 1 when $m \to \infty$ (from Borel–Cantelli's lemma, [17]), and it can *only* guarantee a fixed rate of convergence of $S^*_{\theta'}$ towards X^*_θ as m increases (from Glivenko–Cantelli's theorem, and [17,18,15]). Statistically speaking, it is thus hard to find some θ' that nullifies the error incurred, i.e. the weight on $\mathscr{D}$ of $X^*_\theta \Delta S^*_{\theta'}$, for any m. Fortunately, this error is composed of two separate sources that were previously presented in Section 2, and its support can be decomposed as follows:

$$X^*_\theta \Delta S^*_{\theta'} = (X^*_\theta \backslash S^*_{\theta'}) \cup (S^*_{\theta'} \backslash X^*_\theta). \tag{4}$$

It turns out that it is possible to obtain, modulo some *user-fixed* statistical risk δ, some fairly strong constraints on either of its components, i.e. the weight on $\mathscr{D}$ of $X^*_\theta \backslash S^*_{\theta'}$ or $S^*_{\theta'} \backslash X^*_\theta$. What is most interesting is that these constraints hold *regardless* of m.

We now turn to the formal criteria appreciating the goodness of fit of $S^*_{\theta'}$. We define:

$$TP = \sum_{T \in S^*_{\theta'} \cap X^*_\theta} \mathscr{D}(T), \quad FP = \sum_{T \in S^*_{\theta'} \backslash X^*_\theta} \mathscr{D}(T),$$
$$FN = \sum_{T \in X^*_\theta \backslash S^*_{\theta'}} \mathscr{D}(T), \quad TN = \sum_{T \in S^* \backslash (S^*_{\theta'} \cup X^*_\theta)} \mathscr{D}(T).$$

The *precision* allows to quantify the proportion of estimated θ-frequent that are in fact not *true* θ-frequents:

$$\mathtt{P} = TP/(TP + FP). \tag{5}$$

Maximizing $\mathtt{P}$ is equivalent to the minimization of our first source of error. Symmetrically, the *recall* allows to quantify the proportion of *true* θ-frequent that are missed:

$$\mathtt{R} = TP/(TP + FN). \tag{6}$$

Maximizing $\mathtt{R}$ amounts to the minimization of our second source of error. We also make use of a well-known quantity in information retrieval, which is a weighted harmonic average of precision and recall, the $\mathtt{F}_\beta$-measure. Thus, we can adjust the importance of one source of error against the other by adjusting the β value:

$$\mathtt{F}_\beta = (1 + \beta^2)\mathtt{PR}/(\mathtt{R} + \beta^2 \mathtt{P}). \tag{7}$$

Informally, our approach boils down to picking a θ' different from θ, so as to maximize either $\mathtt{P}$ or $\mathtt{R}$. Clearly, extremal values for θ' could address the problem, but they would yield very poor values for F_β, and also be completely useless for data mining purposes. For example, we could choose $\theta' = 0$, and would obtain $S^*_0 = S^*$, and thus $\mathtt{R} = 1$. However, in this case, we would also have $\mathtt{P} = |X^*_\theta|/|S^*|$, a too small value for many domains and values of θ. We would also keep all elements of S^* as *true* θ-frequents patterns, a clearly huge drawback for mining issues. We could also choose $\theta' = 1$, so as to be sure to maximize $\mathtt{P}$ this time; however, we would also have $\mathtt{R} = 0$, and would keep *no* element of S^* as θ-frequent patterns. Fig. 1 (d) gives the possible choices of θ', for some $\varepsilon > 0$ presented below.

3.1. (θ, ε)-*covers*

We adopt the concise probabilistic notation of [19], and define for some predicate P the notation $\forall^\delta P$ which means that P holds for all but a fraction $\leqslant \delta$ of the sets S sampled under distribution $\mathscr{D}$. Equivalently, P holds with probability $\geqslant 1 - \delta$ over the sampling of S on distribution $\mathscr{D}$. The following definition is the cornerstone of our approach.

Definition 2. $\forall 0 \leqslant \theta \leqslant 1$, $\forall 0 \leqslant \varepsilon \leqslant 1$, $\forall S \subseteq X$, we say that S^* is a **sup**-(θ, ε)-**cover** of X iff $\forall T \in X^*_\theta$,

$$\rho_S(T) \geqslant \rho_X(T) - \varepsilon. \tag{8}$$

Respectively, we say that S^* is an **inf**-(θ, ε)-**cover** of X iff $\forall T \in S^* \backslash X^*_\theta$,

$$\rho_S(T) \leqslant \rho_X(T) + \varepsilon. \tag{9}$$

The way we use Definition 2 is simple. Consider that the user has fixed both the theoretical support $0 \leqslant \theta \leqslant 1$, and the *statistical risk* parameter $0 < \delta < 1$. Suppose we can find ε such that:

$$\forall^\delta, S^* \text{ is an inf-}(\theta, \varepsilon)\text{-cover of } X. \tag{10}$$

Now, fix $\theta' = \theta + \varepsilon$, so that we keep $S^*_{\theta+\varepsilon}$. Because (10) holds, we observe $\forall T \in S^* \backslash X^*_\theta$, $\rho_S(T) \leqslant \rho_X(T) + \varepsilon < \theta + \varepsilon$. Thus, we obtain $\forall^\delta$, $S^*_{\theta+\varepsilon} \subseteq X^*_\theta$, which easily yields:

$$\forall^\delta, \mathtt{P} = 1. \tag{11}$$

Thus, there is *no* first source of error, with high probability.

Now, suppose we can find ε such that $\forall^\delta, S^*$ is a sup-(θ, ε)-cover of X, and fix this time $\theta' = \theta - \varepsilon$, so that we keep $S^*_{\theta-\varepsilon}$. Because of the property of S^*, we observe $\forall T \in X^*_\theta$, $\rho_S(T) \geqslant \rho_X(T) - \varepsilon \geqslant \theta - \varepsilon$, which yields $\forall^\delta$, $X^*_\theta \subseteq S^*_{\theta-\varepsilon}$, and finally:

$$\forall^\delta, \mathtt{R} = 1. \tag{12}$$

In that case, there is *no* second source of error with high probability.

Computationally speaking, both sets $S^*_{\theta+\varepsilon}$ and $S^*_{\theta-\varepsilon}$ can be easily built empirically from S^*. Solving problem (**Pb2**) is now reduced to finding an accurate value of ε such that S^* is a sup or inf -(θ, ε)-cover of X with high probability. This is exposed in the following subsection.

3.2. Finding ε

The following theorem gives a value ε which yields with high probability a sup -(θ, ε)-cover of X.

Theorem 1. $\forall X, \forall \mathscr{D}, \forall m > 0, \forall 0 \leqslant \theta \leqslant 1, \forall 0 < \delta \leqslant 1$, *the following holds*: $\forall^\delta$, S^* *is a sup-*(θ, ε)*-cover of* X, *for any* ε *satisfying*:

$$\varepsilon \geqslant \sqrt{(1/(2m)) \ln(|X^*_\theta|/\delta)}.$$

Proof. A standard application of Chernoff bounds yields that the probability for any *fixed* pattern $T \in X$ to observe $\rho_S(T) \leqslant \rho_X(T) - \varepsilon$ is no more than $\exp(-2m\varepsilon^2)$. Using the union bound, the probability that this is observed for *some* pattern $\in X^*_\theta$ is no more than $|X^*_\theta| \exp(-2m\varepsilon^2)$. Solving for ε this quantity equal to δ yields the theorem. □

The same kind of result can be obtained for inf -(θ, ε)-covers, with the same proof. Hereafter, we give the statement of the theorem.

Theorem 2. $\forall X, \forall \mathscr{D}, \forall m > 0, \forall 0 \leqslant \theta \leqslant 1, \forall 0 < \delta \leqslant 1$, *the following holds*: $\forall^\delta$, S^* *is an inf-*(θ, ε)*-cover of* X, *for any* ε *satisfying*:

$$\varepsilon \geqslant \sqrt{(1/(2m)) \ln(|S^* \backslash X^*_\theta|/\delta)}$$

Theorems 1 and 2 say that finding (inf / sup)-(θ, ε)-covers is a fairly easy task. What they do *not* say is whether this simplicity can be replaced by another approach, may be more sophisticated, to find significantly *better* covers. In other words, could there exist equivalents to Theorems 1 and 2 with a significantly smaller ε? In the following subsection, we discuss some properties of our method, and show in particular that the answer to this question is no.

3.3. Near optimality of (θ, ε)-covers

The following argument shows that there are no significant better covers than those proposed in Theorems 1 and 2. Informally, we build to this extent a skewed distribution $\mathscr{D}$ on some very simple X^*_θ, such that with probability $\geqslant \delta$ we "miss" the (θ, ε)-cover for some value of ε slightly smaller than that proposed in Theorems 1 or 2. The following theorem proves the result for sup -(θ, ε)-covers of X.

Theorem 3. $\exists X, \exists \mathscr{D}, \exists m > 0, \exists 0 \leqslant \theta \leqslant 1, \exists 0 < \delta \leqslant 1$ *such that the following holds*: *with probability* $\geqslant \delta$, S^* *is* **not** *a sup-*(θ, ε)*-cover of* X, *for any* ε *satisfying*:

$$\varepsilon \leqslant c\sqrt{(1/(2m)) \ln(|X^*_\theta|/\delta)},$$

for some constant $c < 1$.

The proof of this theorem is postponed to the Appendix. Since failing to obtain a sup -(θ, ε)-covers of X ultimately means failing to have maximal recall, our computation of ε is thus close to the best possible which keeps the guarantees we want on recall. Obviously, the same kind of theorem holds for inf -(θ, ε)-covers of X, and its proof follows that of Theorem 3.

Theorem 4. $\exists X, \exists \mathscr{D}, \exists m > 0, \exists 0 \leqslant \theta \leqslant 1, \exists 0 < \delta \leqslant 1$ *such that the following holds*: *with probability* $\geqslant \delta$, S^* *is* **not** *an inf-*(θ, ε)*-cover of* X, *for any* ε *satisfying*:

$$\varepsilon \leqslant c\sqrt{(1/(2m)) \ln(|S^* \backslash X^*_\theta|/\delta)},$$

for some constant $c < 1$.

The criterion which is not controlled may suffer some loss, but what Theorems 3 and 4 say on this criterion is that the loss it incurs is also statistically near-optimal; a simple argument shows that the *value* of this loss behaves in a very reasonable manner: Theorems 1 and 2 guarantee that $\varepsilon \leqslant (1/m) \log m^*$ for reasonable δ; since generating S^* is as worst reasonably polynomial in m, we can expect $m^* \leqslant m^k$ for some small constant $k > 0$, which yields $\varepsilon \leqslant 1/m^{1-o(1)}$. In other words, $\theta \pm \varepsilon$ converges quite rapidly to θ, and since a similar rate of convergence of the observed frequencies to their expectations holds as well, we observe a fast convergence of $X^*_\theta \backslash S^*_{\theta+\varepsilon} \rightarrow \emptyset$ (for $\theta' = \theta + \varepsilon$) or $S^*_{\theta-\varepsilon} \backslash X^*_\theta \rightarrow \emptyset$ (for $\theta' = \theta - \varepsilon$), ensuring a reasonably fast maximization of the unconstrained criterion as well.

3.4. Discussion

We now shift to a discussion on the way our approach behaves when there is a *distribution drift*, i.e. when $\mathscr{D}$ changes through time. The way we estimate the true probabilities is pointwise, so we cannot easily model their functional variation (i.e. build some regression model of the probabilities as a function of time); however, regardless of the drifts of $\mathscr{D}$, what we need is only to make accurate updates of our predictions on the θ-frequent patterns. It turns out that our approach can be tailored in a very simple way to estimate these changes in X^*_θ. This simply consists in estimating $\rho_S(.)$

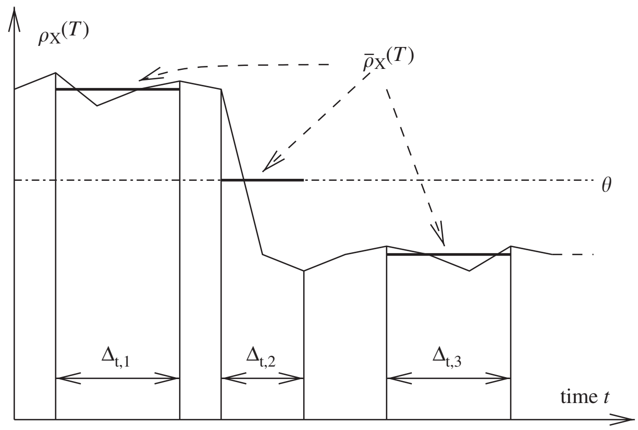

Fig. 3. A moving window makes it possible to track distribution drifts. In this example, we may detect that T is θ-frequent during window $\Delta_{t,1}$ while it is not θ-frequent anymore during $\Delta_{t,3}$ (see text for details).

on the basis of a *moving window*, wide enough to ensure m large enough, and regularly sampling the data stream. All other parameters *do not change*. With this straightforward adaptation, Fig. 3 explains that the distribution drift is estimated with respect to the moving average of the distributions (thick lines, for three windows, $\Delta_{t,1}$, $\Delta_{t,2}$, $\Delta_{t,3}$), and *not* with respect to the true distributions (regular line). In other words, we estimate for any pattern T the fluctuations of a moving average $\overline{\rho}_X(T)$ instead of $\rho_X(T)$. With respect to this change, it is straightforward to show that the results of Theorems 1 and 2 still hold, and thus that we manage, under any such distribution drift, to keep maximal precision or recall with respect to the *average* drift. This smoothes the small local drifts, but keeps the significant variations of $\mathscr{D}$ within the detection range. These variations are those that play the key roles in the shifts of X^*_θ.

There only remains to upperbound $|X^*_\theta|$ and $|S^*\backslash X^*_\theta|$ to compute empirically ε for Theorems 1 and 2, respectively. The true cardinals depend on both the nature (the complexity) of the patterns built, and on the underlying distribution $\mathscr{D}$ (since it depends on θ). Thus, it may be hard to compute them exactly. Since $|X^*_\theta| + |S^*\backslash X^*_\theta| = m^*$, we shall use afterwards in the experiments the same upperbound, m^*, for both cardinals.

4. Experiments

Two kinds of experiments were performed. First, we evaluate how our statistical supports are helpful to mine frequent patterns. Second, we analyze the behavior of our approach according to distribution drifts.

4.1. Evaluation of statistical supports

Experiments are provided on two different settings: itemset databases, and sequential pattern databases.

4.1.1. Itemset databases

We have chosen three real life databases from the Frequent itemsets Mining Dataset Repository [20], whose principal goal is to evaluate and compare association rules algorithms. Fig. 4 gives the details of the databases. To make a fair evaluation of statistical supports, the databases are used to represent X, and a data stream is created by random sampling, out of which a window is saved (S) whose size represents a fixed percent of the original database size. To make these experiments as exhaustive as possible, many parameters have been tested, and Fig. 5 presents each of them. As shown in this figure, two kinds of samplings have been used. The first allows a fine sampling of the database, for small values ranging from 1% to 10% by steps of 1% (column "sampling1" in Fig. 5), and typically gives an idea of what may happen for very large, fast data streams. We have completed this first range with a coarse range of samplings, from 10% to 100% by steps of 3% (column "sampling2" in Fig. 5), which gives a basic idea of the average and limit behaviors of our method. Finally, δ has been chosen to range through a somewhat usual interval of values for common statistical risks, i.e. starting from 1% and stopping at 11% by steps of 2% (see Fig. 5). On the top of our experiments, we have chosen to use an implementation of the a priori

Database	DB size	Total items	Max. size	Avg. size
Accidents	340183	468	51	34
Retail	88163	16470	76	11
Kosarak	990002	41270	2498	9

Fig. 4. Itemset Databases. For each of them, we give, from left to right, the whole number of transactions of the database, the whole number of items, the maximum size of a transaction, and the average size of a transaction.

Database	θ	sampling1	sampling2	δ
Accidents	[.3, .9] / .05	[.01,.1] / .01	[.1, 1] /.03	[.01, .11] / .02
Retail	[.05, .1] / .01	[.01,.1] / .01	[.1, 1] /.03	[.01, .11] / .02
Kosarak	[.05,.1] / .01	[.01,.1] / .01	[.1, 1] /.03	[.01, .11] / .02

Fig. 5. Range of parameters for the experiments. For each parameter, the range of values it takes is given on the form $[a, b]/c$, where a is the starting value, c is the increment, and b is the last value. Thus, the set of values is $\{a, a+c, a+2c, \ldots, b\}$. θ is the minimum theoretical support, δ is the risk parameter. The columns "sampling1" and "sampling2" give the two scales of percentages of the database sampled out of the data stream (see text for details).

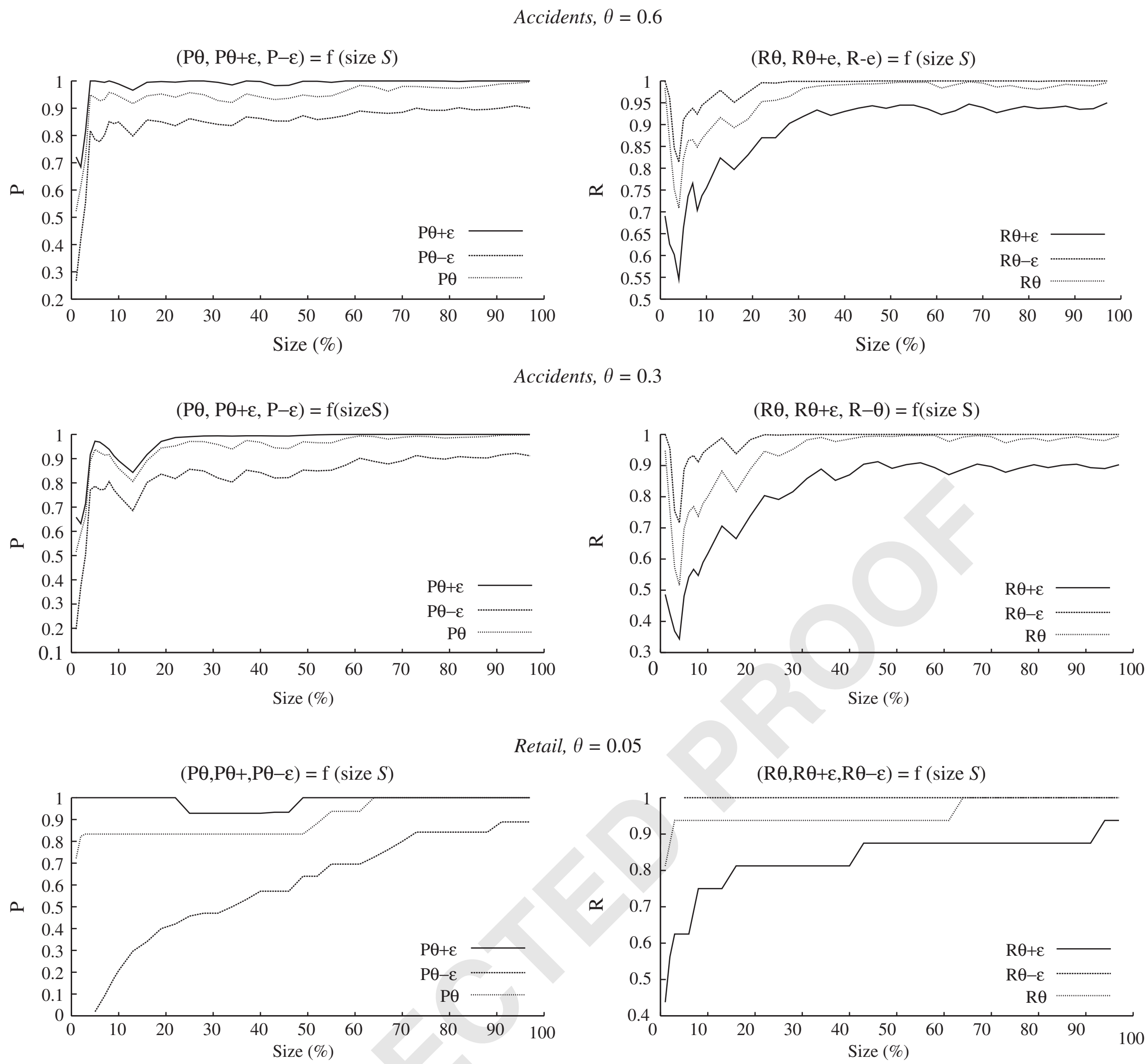

Fig. 6. Three examples of plots for two of our databases, with $\delta = .05$. For three different values of θ, we give the precision (left plot) and recall (right plot) for the three methods consisting in picking $S^*_{\theta-\varepsilon}$, S^*_{θ}, $S^*_{\theta+\varepsilon}$. The x-axis denotes the percentage of the data kept out of the simulated data stream (see text for details).

algorithm [9]. Given the very large number of tests to do for each database, we have written a test generator, which automatically crosses the parameters, and makes all experiments for all possible tuples of parameters. This represents thousands of runs, and due to this very large number and the lack of space, we have chosen to report some plots we consider as representative, and synthesize the whole results. Fig. 6 shows result from experiments on the *Accidents* and *Retail* databases. Each plot describes for one database and one support value, either the precision or recall of the three methods which consist in keeping $S^*_{\theta-\varepsilon}$, S^*_{θ}, and $S^*_{\theta+\varepsilon}$. Notice that the value of the risk parameter is kept constant, i.e. $\delta = .05$.

A first glance at these plots, or the other ones, on whichever of the three databases, reveals that their behavior is almost always the same. Namely:

- the precision equals or approaches 1 for a large majority of storing sizes when $\theta' = \theta + \varepsilon$,
- the recall equals or approaches 1 for a large majority of storing sizes when $\theta' = \theta + \varepsilon$.

These observations are in accordance with the theoretical results of Section 3. There is another phenomenon we may observe: for example, the recall associated to $\theta' = \theta + \varepsilon$ is not that far from the recall of $\theta' = \theta$. Similarly, the precision associated to $\theta' = \theta - \varepsilon$ is not that far from the precision of $\theta' = \theta$. This shows that the maximization of the precision or recall is obtained at a reduced degradation of the other parameter.

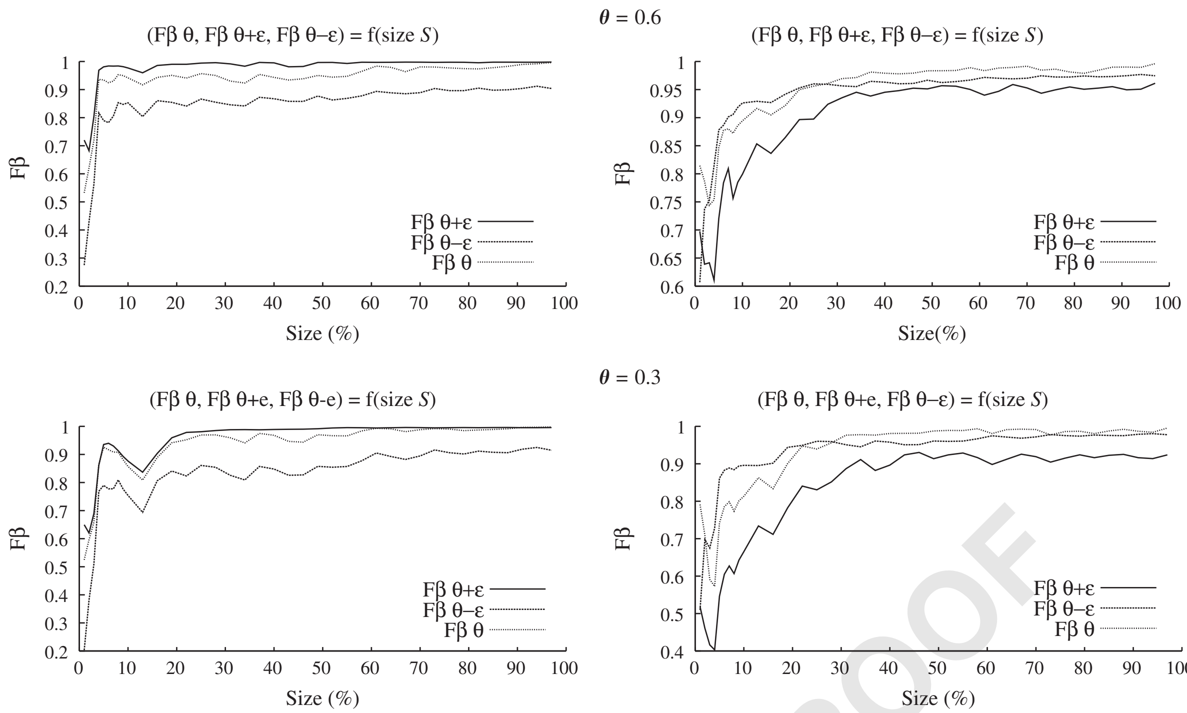

Fig. 7. Two sets of plots of the F_β value from the *Accidents* database, with $\beta = .2$ for the left plots and $\beta = 1.8$ for the right plots (see text for details).

A close look at small storing sizes of the streams (before 10%) also reveals a more erratic behavior without convergence to maximal precision or recall. The behavior for the *Retail* and *Kosarak* databases is also the same. Rather than being due to the statistical supports, we feel that this behavior is linked to the sizes of the databases used. Small databases lead to even smaller storing sizes, and frequent itemsets are in fact trickier to predict. This, we think, may not be expected from larger databases, or even real-world data streams, for which the size of X is much larger. In Fig. 7, two sets of two plots taken from the *Accidents* database plot the F_β measure, against the size of the stream used (in %). The values of β have been chosen different from 1 to make precision and recall have significantly different importances. On each plot, the F_β value displays the advantage that picking $\theta' = \theta \pm \varepsilon$ may have over the choice $\theta' = \theta$, when precision and recall have different importance, i.e. for mining problems with varying misestimation costs.

4.1.2. Sequential pattern databases

In order to evaluate our predictive method with sequential patterns, we have chosen two real life databases from web servers. Fig. 8 summarizes these databases. *Dragons* is obtained from an internet web site[1] from March 21th 2005 to March 28th 2005: the data represent the behavior of this web site usage. The web log size is about 2,54Go. A preprocess was done in order to prune irrelevant data (spiders, robots, etc.). In order to avoid traditional problems when considering raw web logs, URL pages having same values for similar variables were grouped together. Finally, we consider that the session time was set to 4 h. The second database of Fig. 8, named *BuAG*, is obtained from the 3,48Go web log server of some university's library,[2] from January 1st to November 1st 2004. As previously, a preprocess was done and the session time was set to 3 min.

Database	DB size	Total items	Max. size	Avg. size
Dragons	132361	2801	2061	45
BuAG	54798	2121	5722	12

Fig. 8. Sequential pattern databases. For each of them, we give from left to right: the whole number of transactions, the whole number of items, the maximum size of a transaction, and the average size of a transaction.

Experiments similar to itemset databases have been performed. Fig. 9 summarizes the varying parameters (δ was fixed to .05). On the top of our experiments, we have chosen to use a traditional sequential pattern algorithm, `PSP` [21]. Similarly to the itemsets databases, a generator was developed due to the large number of tests.

Fig. 10 shows some results obtained. Similar to itemsets databases, the plots are in accordance with the theoretical results of Section 3. On these results, there is however a

[1] www.elevezundragon.com.

[2] www.univ-ag.fr/buag/.

Database	θ	sampling1	sampling2
Dragons	[.07, .2] / .03	[.02,.1] / .01	[.15, .7] /.05
BuAG	[.08, .2] / .03	[.05,.1] / .01	[.15, .7] /.05

Fig. 9. Range of parameters for the experiments on sequential patterns databases. Conventions follow Fig. 5.

greater difference between the curves for $\theta \pm \varepsilon$, for whichever of the precision or recall, and this difference is as larger as the database stored is smaller. We feel that this is again due to the storage size, but there may also be a setting influence, which makes that sequences are more difficult to handle than (unordered) itemsets.

In Fig. 11, two sets of two plots taken from the *Dragons* database plot the F_β measure, against the size of the stream used (in %). Similar for precision and recall, the results are more contrasted than for itemset databases, as there is no clear winning strategy. However, the results tend to get better when β gives more importance to precision.

4.2. Distribution drifts

Experiments were performed on distribution drifts with the *Accidents* database (see Section 4.1.1). Fig. 12 describes the experimental protocol for drift generation. Basically, the stream is generated by alternating two periods that switch the database used to generate the stream. There is a

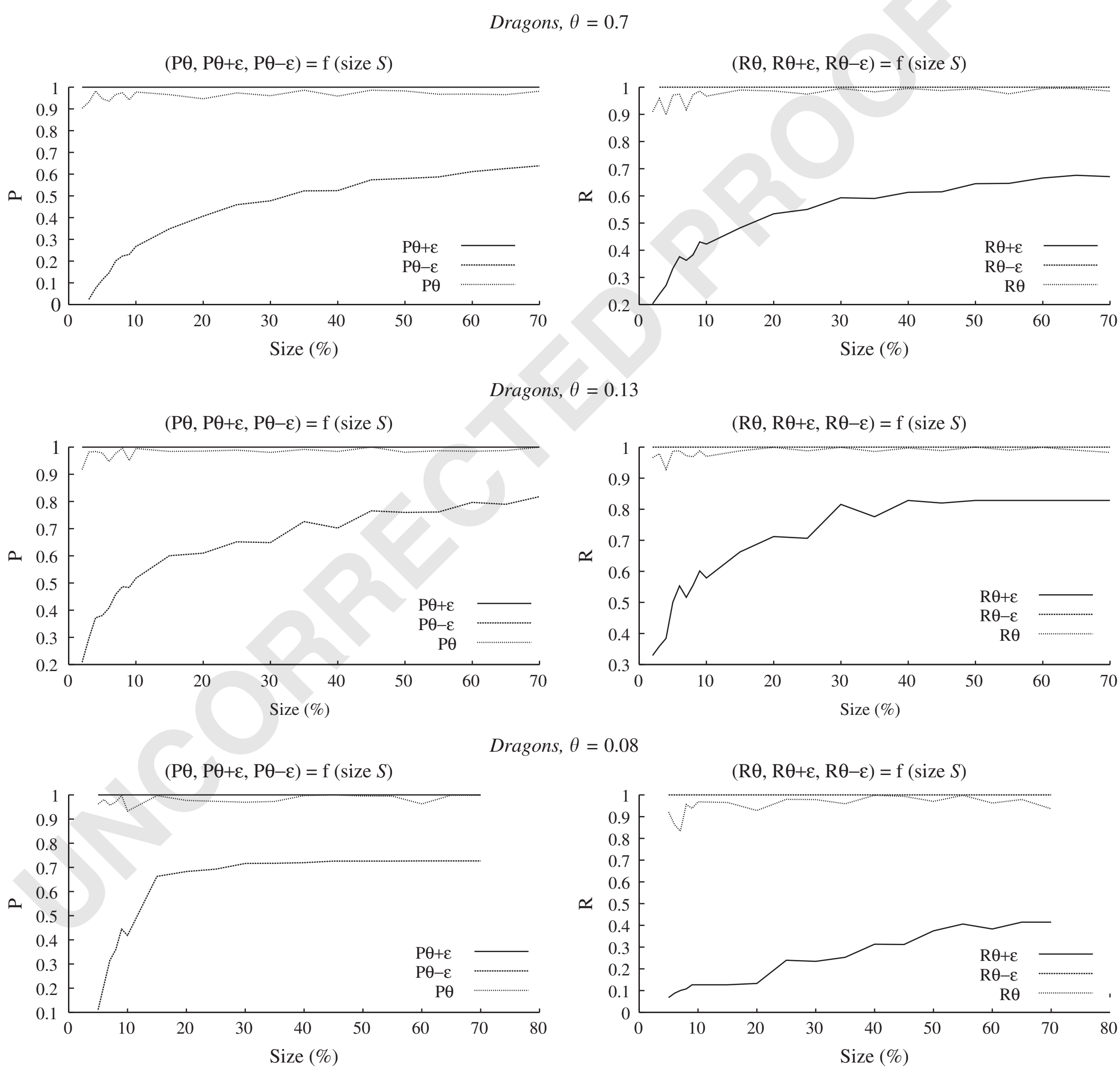

Fig. 10. Examples of plots with $\delta = .05$ and three θ values. For theses values we give the P (left plot) and R (right plot) for the three methods consisting in picking $S^*_{\theta-\varepsilon}$, S^*_θ, $S^*_{\theta+\varepsilon}$.

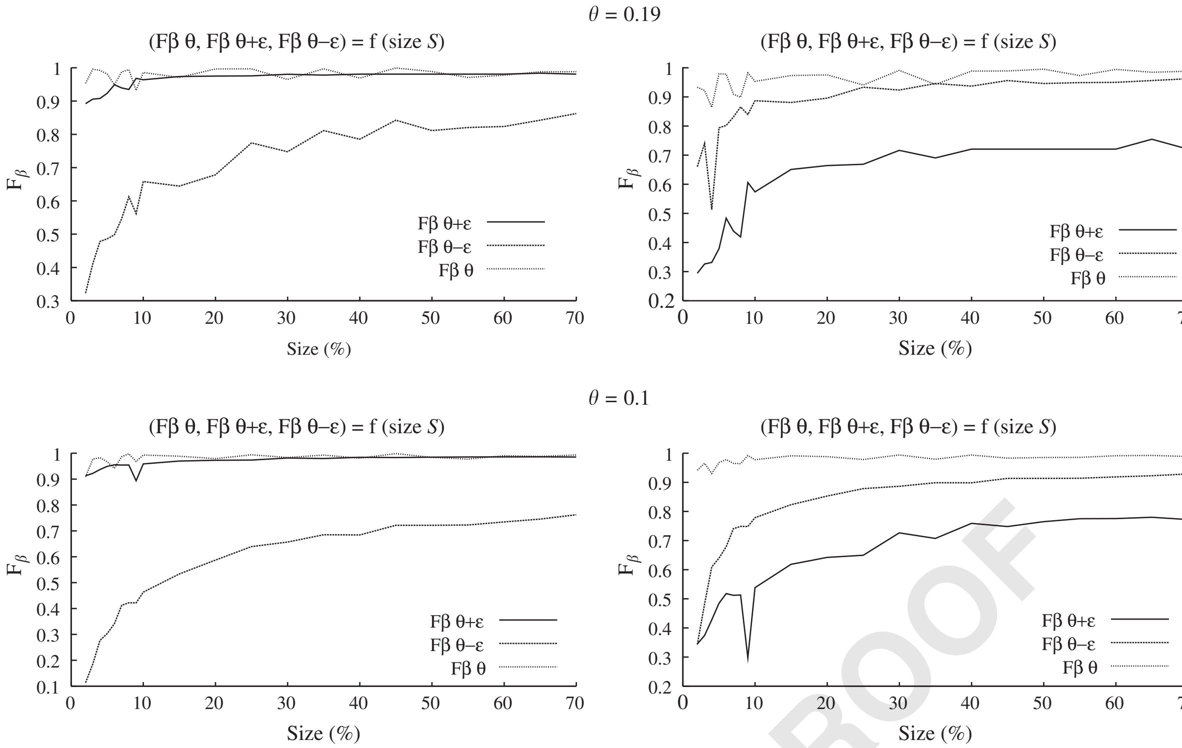

Fig. 11. Two sets of plots of the F_β value from the *Dragons* database, with $\beta = .2$ for the left plots and $\beta = 1.8$ for the right plots.

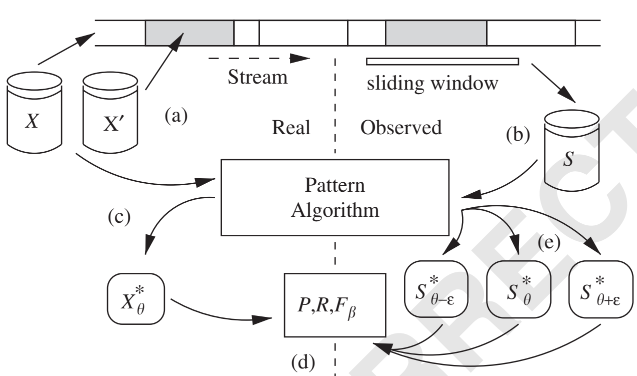

Fig. 12. Our framework with distribution drifts (see text for details).

% driftsize	1.5%	7%	50%
$0-33\%$	$\theta+\varepsilon,\theta,\theta-\varepsilon$	$\theta+\varepsilon,\theta,\theta-\varepsilon$	$\theta+\varepsilon,\theta,\theta-\varepsilon$
$34-66\%$	$\theta+\varepsilon,\theta,\theta-\varepsilon$	$\theta,\theta+\varepsilon,\theta-\varepsilon$	$\theta+\varepsilon,\theta,\theta-\varepsilon$
$67-100\%$	$\theta,\theta+\varepsilon,\theta-\varepsilon$	*n.a.*	*n.a.*

Fig. 13. Precision under drift (summary, see text for details).

so-called *undrift* period, which corresponds to a period where the stream is generated by sampling the usual database, X. There is also a *drift* period, on which we sample a database X' which is some "drifted" version of X, i.e. for which distribution $\mathscr{D}$ is modified. In order to control the drift, X' is obtained by repeatedly sampling X with different parameters (size of X', minimal/maximal repetition of sequences, choice of data sequences, ...). Drift periods are represented by gray sequences in Fig. 12. The database stored, S, is a sliding window which moves along the stream, sampling some mixed database of X and X'.

Again, we use an experiment generator which crosses various parameters. The support, θ, ranges from 40% to 80% by step of 5%. In the stream described in Fig. 12, undrift periods that have $20k$ transactions[3] alternate with drift periods that have $10k$ transactions. The window size ranges from $5k$ transactions to $165k$ transactions by steps of $20k$ transactions. For *each* possible sliding window, we compute precision and recall. Due to the lack of space, we present here the results for $\theta = 40\%$. Fig. 13 summarizes the results for precision P, for three typical window sizes: 1.5%, 7% and 50% of the whole stream (respectively small, intermediate and large sizes). The rows depict three drift ratios, where the ratio is the percentage of transactions in the windows that come from the drifted database X'. Each cell of the table displays, from the left to the right, the

[3] $20k = 20\,000$.

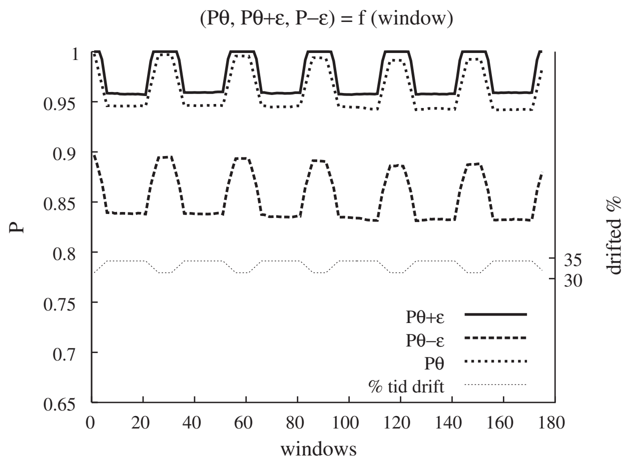

Fig. 14. Precision plots for $\theta' = \theta$, $\theta + \varepsilon$, $\theta - \varepsilon$, for a 50% window size (we recall that $\theta = 40\%$). Notice that the left and right y-axes (drift ratio) do *not* have the same scale.

% driftsize	1.5%	7%	50%
0 – 33%	$\theta - \varepsilon$, θ, $\theta + \varepsilon$	$\theta - \varepsilon$, θ, $\theta + \varepsilon$	$\theta - \varepsilon$, θ, $\theta + \varepsilon$
34 – 66%	$\theta - \varepsilon$, θ, $\theta + \varepsilon$	$\theta - \varepsilon$, θ, $\theta + \varepsilon$	$\theta - \varepsilon$, θ, $\theta + \varepsilon$
67 – 100%	θ, $\theta - \varepsilon$, $\theta + \varepsilon$	$n.a.$	$n.a.$

Fig. 15. Recall under drift (summary, conventions follow Fig. 13).

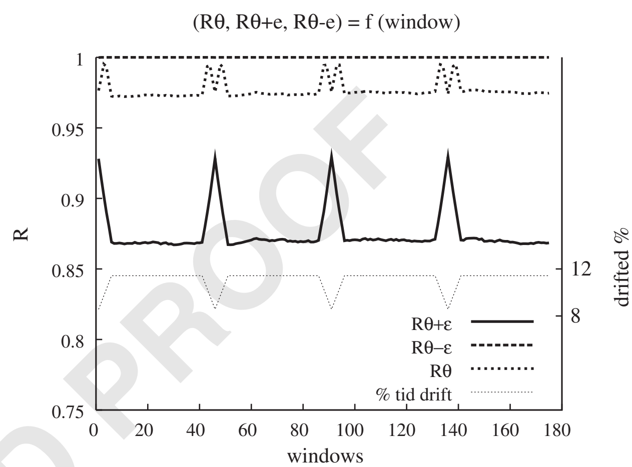

Fig. 16. Recall plots for a particular configuration. Conventions are the same as for Fig. 14.

decreasing order in which the three choices of θ' perform against each other. The left parameter performs the *best*, the right performs the worst, and the middle performs midway between the others. For example, if we have θ, $\theta + \varepsilon$, $\theta - \varepsilon$, it means that plots obtained using $\theta' = \theta$ are the best for the precision. Nonavailable results are indicated by "n.a.". Different phenomena emerge from this table:

- Results for reasonable drifts still follow the theory (a majority for drifts $\leqslant 66\%$), as the order is $\theta + \varepsilon > \theta > \theta - \varepsilon$. Furthermore, the precision approaches its maximum for a window size of 50%, *regardless* of the position of the sliding window on the stream. These are certainly good news for statistical supports.
- The precision tends to increase with the window size.

Fig. 14 gives a snapshot on a particular configuration, in which the drift ratio ranges from 31% to 35%. Remark that the smallest drift brings maximal precision. Again, this is good news, since it is in accordance to a theory initially developed for nondrifted environments, and the drift incurred is clearly not small. Moreover, the order in the plots is constant through time. Finally, the difference between the precisions for $\theta + \varepsilon$ and θ tends to *increase* with the drift ratio. This is also good news for statistical supports. We now proceed in the same way for the recall. The table of Fig. 15 gives results for the recall that follow the conventions of Fig. 13 for the precision. The results appear to be even better than for the precision, since until 66% drift ratio, the order always favors the choice $\theta' = \theta - \varepsilon$. From the precision and recall tables, we can say that for large drifts ($\geqslant 67\%$), the choice $\theta' = \theta$ seems to be the best. We feel that this is partly due to the statistical uncertainty generated by the large drift, which seems to favor the consensus choice $\theta' = \theta$. Fig. 16 presents a snapshot of some results, following Fig. 14. This time, the drift ratio ranges from 8% to 12%. Again, the results obtained follow the theory, since the choice $\theta' = \theta - \varepsilon$ always yield maximum recall. What is more interesting is that this time, the choice $\theta' = \theta$ yields significantly worse results for almost all iterations. Finally, again, the difference between the two choices $\theta' = \theta - \varepsilon$ and $\theta' = \theta$ increases during the drift periods.

5. Related works

A significant body of previous works has addressed the accurate storing of the data stream history. This storage problem consists in finding compact data structures to reduce the size of the data kept out of the stream, while guaranteeing with high probability that the items *observed* as frequent from the stream are still observed frequent inside the data structure [11,13,5]. The first approach was proposed by [7] where they define the first single-pass algorithm. Li et al. [4] use a top-down frequent itemset discovery scheme. A regression-based algorithm is proposed in [22] to find frequent itemsets in sliding windows. Chi et al. [23] consider closed frequent itemsets. In [24], they propose a FP-tree-based algorithm [25] to mine frequent itemsets at multiple time granularities by a novel tilted-time windows technique. It should be more convenient, from a data mining standpoint, to try to reduce the storage uncertainty with an accurate forecasting on the data stream, rather than reducing it to the portion observed. This is the main difference with our framework.

A previous Chernoff-type analysis, due to [26], may be fit to handling data streams as well, but for slightly more restricted problems; in particular, while some of the bounds would typically not be applicable for large S^*, the others would be mainly addressed at controlling the precision of the support estimation, and not the maximization of our criteria (precision or recall). Finally, such results (and ours) do not rely on optimizing *the estimation* of these criteria (utility functions), like for example in [27,28].

Perhaps the works closest to ours are some that have specifically focused in forecasting some properties on data due to a lack of information, either because the data are noisy [29], or because a constraint exists on the data storage that prevents to keep all the information [30]. A first difference with these works is that they focus on approximating (**Pb1**) from Section 3 without emphasis on the components of the solution's accuracy (precision and recall). Thus, they somewhat rely on the sole statistical hardness of the estimation task [15], without drilling down into its components. A second difference, very technical, is that all their bounds are pointwise, i.e. hold for a single itemset, and typically do not yield properties that hold uniformly, i.e. for a whole set of itemsets. That latter case makes it necessary to bring some additional material, such as approximating cardinals or the concept of (θ, ε)-covers, but at this price, we are able to show the statistical near-optimality of our approach (an important issue, not discussed in [30,29]). Finally, the case of distribution drift is not discussed in, or not the subject of, these approaches.

6. Conclusion

There are five main contributions in this paper. First, we discuss the replacement of the conventional minimal support requirement for finding frequent patterns by a statistical support, in cases where storing the entire data is impossible (such as for data streams), so as to keep some convenient properties over the data kept. Then, we provide a method to compute this statistical support, while keeping those relevant properties. The method exploits concentration inequalities for random variables, a tool that has previously been to be helpful from both the theoretical *and* practical standpoints in other domains [31]. We provide a proof that this method is near-optimal from the statistical estimation standpoint. Then, we validate experimentally our approach. A large number of experiments tend to display good points in favor of the applicability and scalability of the method, even under distribution drifts.

There are a number of possible extensions to this work. The most promising extensions to this work certainly concern the application of the technique to relevant data mining subfields, such as incremental mining for computing the near optimal minimal support of semi-frequent patterns [32]. One very promising research direction would also be to integrate our approach with those exploring data structures to maintain items that are observed as frequent with maximal recall [5]. In the framework of data streams, where they are particularly relevant, it would be much more efficient from a statistical standpoint to keep the patterns that are *truly* frequent, better than simply observed as frequent, thus killing two birds in one shot for minimizing approximation errors. Because of the technical machinery used in these papers (e.g. Blum filters [5]), mixing the approaches into a global technique for reducing the error in maintaining frequent itemsets out of data streams may be more than simply interesting: it seems to be very natural.

Appendix A.

We prove Theorem 3. We make the assumption that X^*_θ is a singleton, and θ will be chosen in $(1/2, 1]$: there exists a single θ-frequent itemset T. We also suppose that there are two itemsets in X with respective weight θ (this is T) and $1-\theta$. Given that we sample independently in S the data stream for m itemsets, there is a probability $\geqslant \eta$ to observe $\rho_S(T) < \rho_X(T) - \varepsilon$, with

$$\eta = \binom{m}{m(\theta-\varepsilon)} (1-\theta)^{m(1-\theta+\varepsilon)} \theta^{m(\theta-\varepsilon)}, \tag{13}$$

and $\binom{m}{k} = m!/((m-k)!k!)$ the binomial coefficient. In fact, we could have used for η the tail of the binomial distribution from the terms $k < m(\theta-\varepsilon)$, and this would yield a bound for η stronger than that of Eq. (13). For the sake of readability, we abbreviate $f(m, \theta, \varepsilon)$ the right-hand side of Eq. (13). We make use of the following well-known Stirling-type inequalities:

$$\sqrt{2n\pi}(n/e)^n \leqslant n! \leqslant \exp(1/(12n))\sqrt{2n\pi}(n/e)^n.$$

We obtain the following lowerbound on $f(m, \theta, \varepsilon)$:

$$f(m,\theta,\varepsilon) \geqslant \exp\left(-\frac{1}{12my(1-y)} - \frac{1}{2}\ln(2\pi my(1-y)) - m\left[(1-y)\ln\frac{1-y}{1-\theta} + y\ln\frac{y}{\theta}\right]\right).$$

Here, we have made use of the shorthand $y = \theta - \varepsilon$, which we suppose to be $\in [0, 1]$. The quantity inside the brackets is a Kullback–Leibler divergence, which can be upperbounded with the relationship $\ln(x) \leqslant x - 1$ by

$$(1-y)\ \ln\frac{1-y}{1-\theta} + y\ \ln\frac{y}{\theta} \leqslant \frac{(\theta-y)^2}{\theta(1-\theta)}. \tag{14}$$

Provided m is not too small (in particular, $m \geqslant \max\{4\pi^2, 1 + 1/(3y(1-y))\}$), we may obtain:

$$f(m, \theta, \varepsilon) \geqslant \exp\left(-m\frac{\varepsilon^2}{\theta(1-\theta)} - \ln m\right).$$

Now, provided

$$\varepsilon \geqslant \sqrt{\frac{\theta(1-\theta)}{m}\ln m}, \tag{15}$$

we finally obtain $f(m, \theta, \varepsilon) \geqslant \exp(-2m\varepsilon^2/(\theta(1-\theta)))$. We shall clearly have $f(m, \theta, \varepsilon) \geqslant \delta$ provided

$$\varepsilon = \sqrt{\frac{\theta(1-\theta)}{2m}\ln\frac{1}{\delta}}, \tag{16}$$

which satisfies Eq. (15) whenever $\delta \leqslant 1/m^2$. Choosing θ close to $\frac{1}{2}$ brings the statement of Theorem 3.

References

[1] S. Gollapudi, D. Sivakumar, Framework and algorithms for trend analysis in massive temporal datasets, in: Proceedings of the 13th ACM International Conference on Information and Knowledge Management, 2004, pp. 168–177.

[2] W. Fan, Y.-A. Huang, H. Wang, P.-S. Yu, Active mining of data streams, in: Proceedings of the Fourth SIAM International Conference on Data Mining, 2004, pp. 457–461.

[3] L. Golab, M. Tamer Ozsu, Issues in data stream management, ACM SIGMOD Records 2 (2003) 5–14.

[4] H.-F. Li, S.Y. Lee, M.-K. Shan, An efficient algorithm for mining frequent itemsets over the entire history of data streams, in: Proceedings of the First International Workshop on Knowledge Discovery in Data Streams, 2004.

[5] C. Jin, W. Qian, C. Sha, J.-X. Yu, A. Zhou, Dynamically maintaining frequent items over a data stream, in: Proceedings of the 12th ACM International Conference on Information and Knowledge Management, ACM Press, New York, 2003, pp. 287–294.

[6] E. Demaine, A. Lopez-Ortizand, J.-I. Munro, Frequency estimation of internet packet streams with limited space, in: Proceedings of the 10th European Symposium on Algorithms, 2002, pp. 348–360.

[7] G. Manku, R. Motwani, Approximate frequency counts over data streams, in: Proceedings of the 28th International Conference on Very Large Databases, Springer, Berlin, 2002, pp. 346–357.

[8] R.-M. Karp, S. Shenker, C.-H. Papadimitriou, A simple algorithm for finding elements in streams and bags, ACM Trans. Database Systems 28 (2003) 51–55.

[9] R. Agrawal, T. Imielinski, A.-N. Swami, Mining association rules between sets of items in large databases, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1993, pp. 207–216.

[10] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proceedings of the 11th International Conference on Data Engineering, 1995, pp. 3–14.

[11] M. Charikar, K. Chen, M. Farach-Colton, Finding frequent items in data streams, in: Proceedings of the 29th International Colloquium on Automata, Languages, and Programming, 2002, pp. 693–703.

[12] D. Cheung, J. Han, V. Ng, C. Wong, Maintenance of discovered association rules in large databases: an incremental updating technique, in: Proceedings of the 12th International Conference on Data Engineering, 1996, pp. 106–114.

[13] G. Cormode, S. Muthukrishnan, What's hot and what's not: tracking most frequent items dynamically, in: Proceedings of the ACM International Conference on the Principles of Database Systems, ACM Press, New York, 2003, pp. 296–306.

[14] A. Veloso, W. Meira, M. Carvalho, B. Possas, S. Parthasarathy, M.-J. Zaki, Mining frequent itemsets in evolving databases, in: Proceedings of the Second SIAM International Conference on Data Mining, 2002, pp. 31–41.

[15] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[16] H. Mannila, H. Toivonen, Levelwise search and borders of theories in knowledge discovery, Data Mining Knowledge Discovery 1 (1997) 241–258.

[17] L. Devroye, L. Györfi, G. Lugosi, A Probabilistic Theory of Pattern Recognition, Springer, Berlin, 1996.

[18] M.J. Kearns, Y. Mansour, On the boosting ability of top-down decision tree learning algorithms, in: Proceedings of the 28th ACM Symposium on the Theory of Computing, 1996, pp. 459–468.

[19] D. McAllester, Some PAC-Bayesian theorems, Mach. Learning 37 (1999) 355–363.

[20] Frequent itemset mining dataset repository, 2005. ⟨http://fimi.cs.helsinki.fi/data⟩.

[21] W.-G. Teng, M.-S. Chen, P.S. Yu, A regression-based temporal patterns mining schema for data streams, in: Proceedings of the 29th International Conference on Very Large Databases, 2003, pp. 93–104.

[22] F. Masseglia, F. Cathala, P. Poncelet, The PSP approach for mining sequential patterns, in: Proceedings of the Second European Conference on Knowledge Discovery in Databases, Springer, Berlin, 1998, pp. 176–184.

[23] Y. Chi, H. Wang, P.S. Yu, R.R. Muntz, Moment: maintaining closed frequent itemsets over a stream sliding window, in: Proceedings of the Fourth IEEE International Conference on Data Mining, 2004, pp. 59–66.

[24] G. Giannella, J. Han, J. Pei, X. Yan, P. Yu, Mining frequent patterns in data streams at multiple time granularities, in: Next Generation Data Mining, MIT Press, 2003.

[25] J. Han, J. Pei, B. Mortazavi-asl, Q. Chen, U. Dayal, M. Hsu, Freespan: frequent pattern-projected sequential pattern mining, in: Proceedings of the Sixth International Conference on Knowledge Discovery in Databases, 2000, pp. 355–359.

[26] H. Toivonen, Sampling large databases for association rules, in: Proceedings of the 22th International Conference on Very Large Databases, 1996, pp. 134–145.

[27] C. Domingo, R. Gavaldà, O. Watanabe, Adaptive sampling methods for scaling up knowledge discovery algorithms, Data Mining and Knowledge Discovery 6 (2002) 131–152.

[28] T. Scheffer, S. Wrobel, Finding the most interesting patterns in a database quickly by using sequential sampling, Mach. Learning Res. J. 3 (2002) 833–862.

[29] J. Yang, W. Wang, P.-S. Yu, J. Han, Mining long sequential patterns in a noisy environment, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, Springer, Berlin, 2002, pp. 406–417.

[30] P.-B. Gibbons, Y. Matias, New sampling-based summary statistics for improving approximate query answers, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 1998, pp. 331–342.

[31] R. Nock, F. Nielsen, Statistical region merging, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 1452–1458.

[32] X. Cheng, X. Yan, J. Han, Incspan: incremental mining of sequential patterns in large database, in: Proceedings of the 10th International Conference on Knowledge Discovery in Databases, 2004, pp. 527–532.