Statistical Supports for Frequent Itemsets on Data Streams

Pierre-Alain Laur[1], Jean-Emile Symphor[1], Richard Nock[1], and Pascal Poncelet[2]

[1] Grimaag-Dépt Scientifique Interfacultaire,
Université Antilles-Guyane, Campus de Schoelcher,
B.P. 7209, 97275 Schoelcher Cedex, Martinique, France
{palaur,je.symphor,rnock}@martinique.univ-ag.fr
[2] Lg2ip-Ecole des Mines d'Alès,
Site EERIE, parc scientifique Georges Besse,
30035 Nîmes Cedex, France
pascal.poncelet@ema.fr

Abstract. When we mine information for knowledge on a whole data streams it's necessary to cope with uncertainty as only a part of the stream is available. We introduce a stastistical technique, independant from the used algorithm, for estimating the frequent itemset on a stream. This statistical support allows to maximize either the precision or the recall as choosen by the user, while it doesn't damage the other.Experiments with various association rules databases demonstrate the potential of such technique.

1 Introduction

A growing body of works arising from researchers in Databases and Data Mining deals with data arriving in the form of continuous potentially infinite streams. Many emerging and real applications generate data streams: trend analysis, fraud detection, intrusion detection, click stream, among others. In fraud detection, data miners try to detect suspicious changes in user behavior [5]. Trend analysis is an important problem that commercial applications have to deal with [8]. Security of network systems is becoming increasingly important as more and more sensitive informations are being stored. Intrusion detection has thus become a critical approach to protect systems [7].

From now on, we consider *items* to be the unit information, and *itemsets* to be sets of items. An itemset is *θ-frequent* if it occurs in at least a fraction θ of the stream (called its support), where θ is a user-specified parameter. As the item flow is fast and represent a huge amount of information, it prevents its exact storage. Out of the uncertainty it generates, the problem becomes to store the information so as to keep valid its most crucial contents. One example of such a content is the list of the most frequent items of itemsets encountered, a crucial issue in Data Mining that has recently attracted significant attention [6, 10, 12, 7].

When the database is subject to be updated regularly, maintaining frequent itemsets has been successfully addressed by various incremental algorithms [2, 19]. But due to the high frequency and potentially huge information carried out in a timely fashion by data streams, these incremental approaches cannot easily handle them, unless they take the risk to make errors [18] and/or fail at estimating supports, one of the two essential components of association rules algorithms. This is where our paper takes place.

More precisely, we address the following questions:

(a) is it possible to set up a method which replaces the *exact* support by a *statistical* support ensuring some desirable properties on support computations, and frequency estimations ? Ideally, we would like the resulting support to hold regardless of the algorithm used to build or maintain frequent items/itemsets (see *e.g.* [2, 19]), and rely on mild theoretical assumptions so as to be reliably implementable.
(b) how good is this statistical support, both from the theoretical and experimental standpoints ?

The rest of this paper is organized as follows. Section 2 goes deeper into presenting the problems of dealing with uncertainty in data streams, and gives an extensive statement of our problem. Section 3 presents our solution to this problem, and its properties. Section 4 presents experimental results, and Section 5 concludes the paper with future avenues for research.

2 Problem Statement

The huge size of data streams for real-world domains compared to the limited amounts of resources to mine them makes it necessary to cope with uncertainty to achieve reasonable processing time and/or space. A significant body of previous works has addressed the accurate storing of the data stream history [1, 3, 10].

Our setting is a bit more downstream, as we question the forecasting on the data stream future. Ideally, this information is sought to be accurate not only on the data stored, but also on the whole data stream itself. For example, it's not enough to observe some item as frequent in the data stored; it is much more important to *predict* if it is really frequent in the whole data stream. Similarly, it's not enough to observe that some itemsets doesn't meet the observed frequency requirements to argue that it is *really* not frequent on the whole data stream.

From the estimation standpoint, there are two sources of error:

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

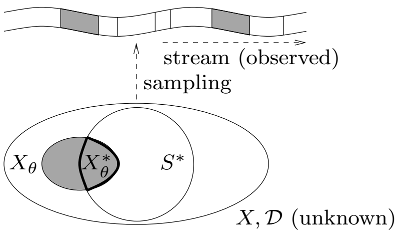


Fig. 1. Problem statement.

Should it rely on frequencies estimations, any loss due to the imperfection of the information stored is incurred by at least one of these sources of error. The point is that it is statistically hard to nullify both of them [17]. It is also generally impossible to capture the missing informations from the data stream to make a fully accurate prediction. Our paper is aimed at obtaining a solution to the following problem, which is a convenient relaxation of this unsatisfiable goal:

(a) the user chooses a source of error, and fixes some related parameters;
(b) the source of error chosen is nullified with high probability, while the other one incurs a limited loss.

It turns out that in many domains [18, 16], the relative importance of the two sources of error is not the same, and one may be much more important to control than the other one. For these domains, our approach may be a very convenient way to cope with uncertainty in finding frequent itemsets.

Now, let us skip to a slightly more formal presentation. The data stream is supposed to be obtained from the repetitive sampling of a potentially huge *domain* X which contains all possible itemsets, see Figure 1. Each itemset is sampled independently through a distribution $\mathcal{D}$ for which we make absolutely *no* assumption, except that it remains fixed (no drift). The reader may find relevant empirical studies on concept drift for supervised mining in [5, 20]. The user specifies a real θ, the *theoretical* support, and wishes to recover all the *true* θ-frequent patterns of X. This set is called X_θ in Figure 1.

Definition 1.

$$\forall 0 \leq \theta \leq 1, X_\theta = \{T \in X : \rho_X(T) \geq \theta\} \ , \tag{1}$$

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

The recovery of X_θ faces two problems. Apart from our statistical estimation problem, there is a combinatorial problem which comes from the fact that X is typically huge, even when finite. The set of observed itemsets which we have sampled from X, hereafter called S, has a size $|S| = m$ ($|S| << |X|$). In our

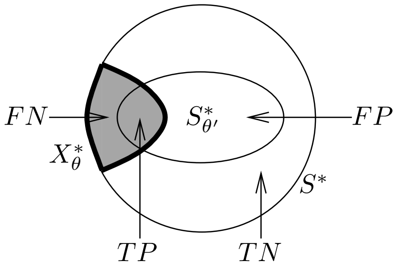


Fig. 2. The error estimation.

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 [13]. This is not the purpose of this paper to cover this combinatorial problem; the key point is that S^* is usually still not large enough to cover X_θ, regardless of the way it is built (see Figure 1), so that the pregnancy of our statistical estimation problem remains the same.

Our statistical estimation problem can be formalized as follows:

- approximate as best as possible the following set:

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

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

Remark that $\forall T \in S^*$, we cannot compute exactly $\rho_X(T)$, since we do not know X and $\mathcal{D}$. Rather, we have access to its best unbiased estimator $\rho_S(T)$:

$$\forall T \in S^*, \rho_S(T) = \sum_{T' \in S : T \leq_t T'} w(T') \ , \tag{3}$$

with $w(T')$ the weight (observed frequency) of T' in S. We adopt the following approach to solve our problem:

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

$$S^*_{\theta'} = \{T \in S^* : \rho_S(T) \geq \theta'\} \ . \tag{4}$$

Before computing θ', we first turn to the formal criteria appreciating the goodness-of-fit of $S^*_{\theta'}$ The two sources of error, committed with respect to X^*_θ, come from the two subsets of the symmetric difference with $S^*_{\theta'}$, as presented in Figure 2. To quantify them, let us define:

$$TP = \sum_{T \in S^*_{\theta'} \cap X^*_\theta} \mathcal{D}(T) \tag{5}$$

$$FP = \sum_{T \in S^*_{\theta'} \setminus X^*_\theta} \mathcal{D}(T) \tag{6}$$

$$FN = \sum_{T \in X^*_\theta \setminus S^*_{\theta'}} \mathcal{D}(T) \tag{7}$$

$$TN = \sum_{T \in S^* \setminus (S^*_{\theta'} \cup X^*_\theta)} \mathcal{D}(T) \tag{8}$$

The *precision* allows to quantify the proportion of estimated θ-frequent that are in fact not true θ-frequents, out of $S^*_{\theta'}$:

$$\mathtt{P} = TP/(TP + FP) \ . \tag{9}$$

Maximizing $\mathtt{P}$ leads to minimize our first source of error. Symmetrically, the *recall* allows to quantify the proportion of true θ-frequent that are missed in $S^*_{\theta'}$:

$$\mathtt{R} = TP/(TP + FN) \ . \tag{10}$$

Maximizing $\mathtt{R}$ leads to minimize our second source of error. We also make use of a 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:

$$\mathtt{F}_\beta = (1 + \beta^2)\mathtt{PR}/(\mathtt{R} + \beta^2\mathtt{P}) \ , \tag{11}$$

A naive approach to approximate X^*_θ would typically be to fix $\theta' = \theta$. Unfortunately, the main and only interesting property of $S^*_{\theta'}$ is that it converges with probability 1 to X^*_θ as $m \to \infty$ from the Borel-Cantelli Lemma [4]. Glivenko-Cantelli's Theorem gives a rate of convergence as a function of m, but this is only useful to yield the maximization of $\mathtt{P}$ and $\mathtt{R}$ in the limit.

3 Choosing θ'

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' would do the job, but they would yield very poor values for F_β, and also be completely useless for data mining. 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, and we would also keep all elements of S^* as true θ-frequents, 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. These extremal examples show the principle of our approach. Should we want to maximize the precision, we would pick a θ' larger than θ to guarantee with high probability that $\mathtt{P} = 1$, yet while keeping large enough values for $\mathtt{R}$ (or F_β), and a set $S^*_{\theta'}$ not too small to contain significant informations. There is obviously a statistical barrier which prevents θ' to be too close to θ to keep the constraint $\mathtt{P} = 1$ (*Cf* Section 2, last §). The objective is to be the closest to this barrier, which statistically guarantees the largest recall values under the constraint.The same principle holds for the maximization of the recall.

The following Theorem states explicitly our bound for the maximal $\mathtt{P}$. Its key feature is that it holds regardless of the domain, the distribution of the itemsets, the size of S^*, or the user-fixed parameters (support, statistical risk). It relies *only* on a rather mild assumption for sampling the itemsets out of the stream.

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. 3. Range of parameters for the experiments in the form $[a, b]/c$, where a is the starting value, c is the increment, and b is the last value.

Theorem 1. $\forall X, \forall \mathcal{D}, \forall m > 0, \forall 0 \leq \theta \leq 1, \forall 0 < \delta \leq 1$, *we pick* ε *satisfying:*

$$\varepsilon \geq \sqrt{\frac{1}{2m} \ln \frac{|S^*|}{\delta}} \ .$$

If we fix $\theta' = \theta + \varepsilon$ *in eq. (4), then* $\mathtt{P} = 1$ *with probability at least* $1 - \delta$.

Theorem 2. $\forall X, \forall \mathcal{D}, \forall m > 0, \forall 0 \leq \theta \leq 1, \forall 0 < \delta \leq 1$, *we pick* ε *satisfying:*

$$\varepsilon \geq \sqrt{\frac{1}{2m} \ln \frac{|S^*|}{\delta}} \ .$$

If we fix $\theta' = \theta - \varepsilon$ *in eq. (4), then* $\mathtt{R} = 1$ *with probability at least* $1 - \delta$.

These Theorems are proven using standard tools on concentration inequalities [14]; due to the lack of space, we skip their proofs. The main point is that the values of θ' seem to be very close to the statistical barriers [17, 11] that still guarantee the maximal values for the precision or recall.

4 Experiments

We focus on evaluating how our statistical support can be helpful to mine frequent itemsets on a data stream, given a fragment of this stream. For this purpose, we use the previously defined measures: $\mathtt{P}$ (9), $\mathtt{R}$ (10) and F_β (11).

We have chosen three real life databases from the Frequent itemsets Mining Dataset Repository [9] and an association rule mining algorithm, `kdci` [15]. The first dataset, named “Accidents” (34k transactions), holds form for each traffic accident that occurs with injured or deadly wounded casualties on a public road. The second data set, named “Retail” (88k transactions), holds customers basket from a retail supermarket store. The last dataset, named “Kosarak” (990k transactions), holds anonymized click-stream data of an on-line news portal.

To analyze the correctness of our statistical supports, we need to evaluate as many situation as possible, that is, we need to use our method with a range as large as possible for each of the free parameters. These parameters that vary during our experiments are described in Fig. 3.

Better than using a real data stream, we have chosen to simulate data streams assuming the complete knowledge of the domains, thus allowing to compute exact values for the performance measurements. More precisely, we simulate data streams by sampling each database into fragments. For example, we could consider that data arrive in a timely manner from the "Accidents" database, and that only 20% of the data is stored. So we pick 20% of the transactions of this database, we consider that it is the data stored. We have chosen to sample the database on a broad range of percentages using two scales. The first allows a fine sampling of the database, for values ranging from 1% to 10% by steps of 1% ("sampling1" in Fig. 3), and typically gives an idea of what may happens 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% ("sampling2") which gives an idea of the average and limit behaviors of our method.

Finally, δ has been chosen to range through an interval of values for common statistical risks, *i.e.* from 1% to 11% by steps of 2% (see Fig. 3). Due to the very large number of experiments and the lack of space to report them all, we have put all resulting plots into web pages[3].

Figure 4 shows result from experiments on the Accidents and Retail databases with $\delta = .05$. Each plot describes for one database and one support value, either P or R of the three methods which consist in keeping $S^*_{\theta-\varepsilon}$, S^*_{θ}, and $S^*_{\theta+\varepsilon}$.

A first glance at these plots reveals that their behavior is almost always the same. Namely:

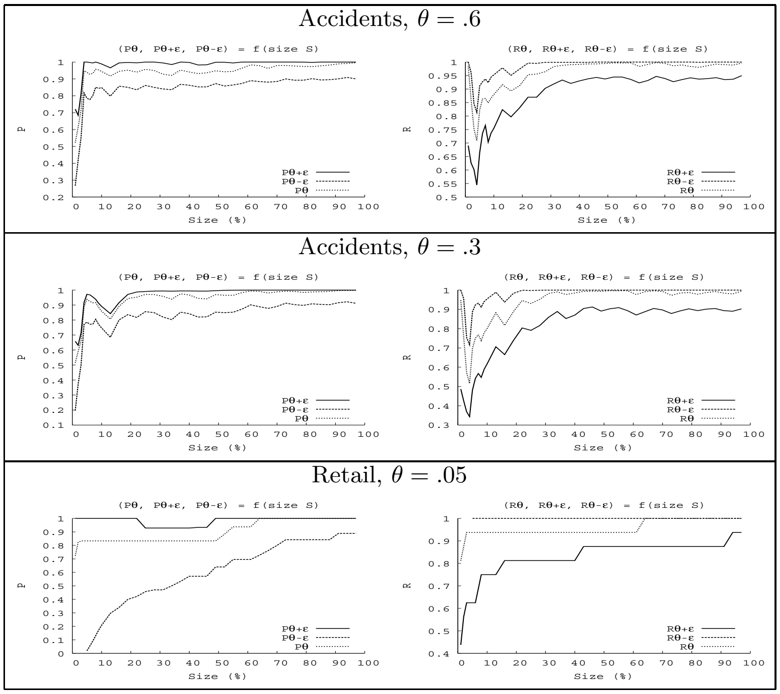
- the P increases with θ' (eq. 4), while the R decreases with θ',
- the P equals or approaches 1 for mostly storing sizes when $\theta' = \theta + \varepsilon$,
- the R equals or approaches 1 for mostly storing sizes when $\theta' = \theta + \varepsilon$.

These observations are in accordance with the theoretical results of Section 3. There is another phenomenon we may observe: the R associated to $\theta' = \theta + \varepsilon$ is not that far from the R of $\theta' = \theta$. Similarly, the P associated to $\theta' = \theta - \varepsilon$ is not that far from the P of $\theta' = \theta$. This shows that the maximization of P or R is obtained at a reduced degradation of the other parameter. We also remark that the P plots tend to be better than the R plots. This is not really surprising, as advocated in Section 3, since the range of values for P is smaller than that of R.

A close look at small storing sizes of the streams (before 10%) also reveals a more erratic behavior without convergence to maximal P or R. This behavior is not linked to the statistical support, but to the databases used. Indeed, small databases lead to even smaller storing sizes, and frequent itemsets kept out of small databases are in fact trickier to predict than for bigger ones. This point is important as, from a real-world standpoint, we tend to store very large databases, so we may expect this phenomenon to be reduced.

On the smallest databases, such as Retail and Kosarak, another phenomenon seems to appear. First of all, because of the small values for θ, some tests have not be performed because $\theta - \varepsilon$ was < 0. Furthermore, the greater difference observed between the curves seems to stem out from the different sizes of databases.

[3] http://www.univ-ag.fr/grimaag/statisticalsupports/



Accidents, $\theta = .6$

Accidents, $\theta = .3$

Retail, $\theta = .05$

Fig. 4. 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 $S^*_{\theta-\varepsilon}, S^*_{\theta}, S^*_{\theta+\varepsilon}$.

For example, the Retail database is smaller than the Accidents database by a factor 3. In addition, the number of frequent itemsets found in this database is smaller than a hundred. For the sake of comparison, the Accidents database for the smallest θ gives hundreds of thousands frequent itemsets. This, we think, explains the greater differences between the curves: they are mostly a small database phenomenon, and may not be expected from larger databases.

In Figure 5, 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 but not too small or too large to yield a reasonable prominence of one criterion (.2 and 1.8, see Figure 5). In each plot, the F_β value displays the advantage of choosing $\theta' = \theta \pm \varepsilon$ against the choice $\theta' = \theta$. Moreover, `R` that this is obtained while statistically guaranteeing the *maximal* value for whichever of `P` or `R` criterion, as chosen by the user.

5 Conclusion

One very promising research direction would be to integrate our approach with the approaches that consisting in somehow reducing the size of the data stored

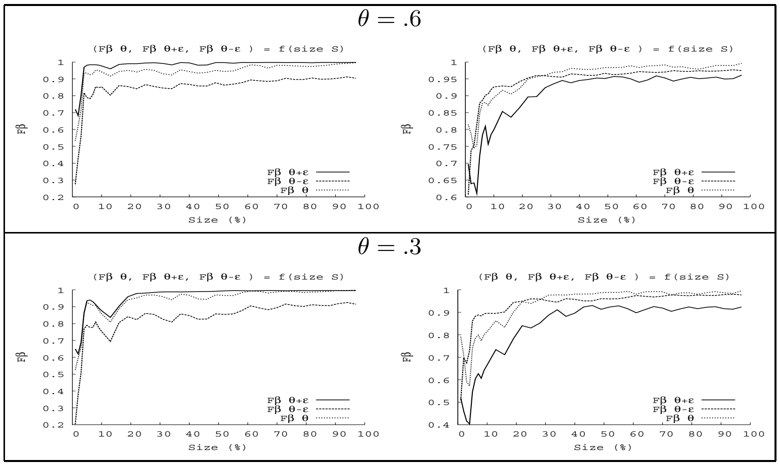


Fig. 5. Two sets of plots of the F_β value from the Accidents database, with $\beta = .2$ for the left plots and $\beta = 1.8$ for the right plots.

out of the database, so as to keep the property that itemsets *observed* as frequent still remain frequent with high probability [10]. In the framework of data streams, where we feel that such approaches take all their importance, it would be much more efficient from a statistical standpoint to keep the itemsets that are *truly* frequent (better than simply observed as frequent). This would basically boil down to mixing our approach with them, so as to keep maximal recall (this can straightforwardly be replaced by the constraint to keep maximal precision). Because of the technical machinery used in these papers (*e.g.* Blum filters [10]), mixing the approaches into a global technique for reducing the error in maintaining frequent itemsets from data streams may be more than simply interesting: it seems to be very natural.

References

1. M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *Proc. of the 29 th International Colloquium on Automata, Languages, and Programming*, pages 693–703, 2002.
2. D. Cheung, J. Han, V. Ng, and C. Wong. Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. In *Proc. of the* 12^{th} *International Conference on Data Engineering*, pages 106–114, New Orleans, Louisiana, February 1996.
3. G. Cormode and S. Muthukrishnan. What's hot and what's not: Tracking most frequent items dynamically. In *Proc. of the* 22^{nd} *ACM Symposium on the Principle of Database Systems*, pages 296–306. ACM Press, 2003.
4. L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, 1996.

5. W. Fan, Y.-A. Huang, H. Wang, and P.-S. Yu. Active mining of data streams. In *Proc. of the* 4^{th} *SIAM International Conference on Data Mining*, pages 457–461, 2004.
6. C. Giannella, J. Han, J. Pei, X. Yan, and P.-S. Yu. *Mining Frequent Patterns in Data Streams at Multiple Time Granularities*, chapter 6. Data Mining: Next Generation Challenges and Future Directions. H. Karguta, A. Joshi, K. Sivakumar and Y. Yesha (Eds.). MIT/AAAI Press, 2004.
7. L. Golab and M. Tamer Ozsu. Issues in Data Stream Management. *ACM SIGMOD Record*, 2(2):5–14, June 2003.
8. S. Gollapudi and D. Sivakumar. Framework and Algorithms for Trend Analysis in Massive Temporal Data Sets. In *Proc. of the 13* th *International Conference on Information and Knowledge Management*, pages 168–177, 2004.
9. Frequent itemset mining dataset repository — http://fimi.cs.helsinki.fi/data, 2005.
10. C. Jin, W. Qian, C. Sha, J.-X. Yu, and A. Zhou. Dynamically maintaining frequent items over a data stream. In *Proc. of the 12* th *International Conference on Information and Knowledge Management*, pages 287–294. ACM Press, 2003.
11. M. J. Kearns and Y. Mansour. A Fast, Bottom-up Decision Tree Pruning algorithm with Near-Optimal generalization. In *Proc. of the 15* th *International Conference on Machine Learning*, pages 269–277, 1998.
12. G. Manku and R. Motwani. Approximate Frequency Counts over Data Streams. In *Proc. of the 28*th *International Conference on Very Large Databases*, pages 346–357, Hong Kong, China, 2002.
13. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
14. R. Nock and F. Nielsen. Statistical Region Merging. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458, 2004.
15. S. Orlando, P. Palmerini, R. Perego, C. Silvestri, and F. Silvestri. kDCI: a multistrategy algorithm for mining frequent sets. In *Proc. of the Workshop on Frequent Itemset Mining Implementations, in conjunction with ICDM 2003*, 2003.
16. S.-J. Rizvi and J.-R. Haritsa. Maintaining Data Privacy in Association Rule Mining. In *Proc. of the 28*th *International Conference on Very Large Databases*, pages 682–693, 2002.
17. V. Vapnik. *Statistical Learning Theory.* John Wiley, 1998.
18. A. Veloso, B. Gusmao, W. Meira, M. Carvalho, S. Parthasarathy, and M.-J. Zaki. Efficiently Mining Approximate Models of Associations in Evolving Databases. In *Proc. of the 6*th *European Conference on the Principles and Practice of Knowledge Discovery in Databases*, pages 435–448, 2002.
19. A. Veloso, W. Meira, M. Carvalho, B. Possas, S. Parthasarathy, and M.-J. Zaki. Mining Frequent Itemsets in Evolving Databases. In *Proc. of the* 2^{nd} *SIAM International Conference on Data Mining*, pages 31–41, Arlington, April 2002.
20. H. Wang, W. Fan, P.-S. Yu, and J. Han. Mining concept-drifting data streams with ensemble classifiers. In *Proc. of the 9*th *International Conference on Knowledge Discovery in Databases*, pages 226–235, 2003.