Association Mining in Gradually Changing Domains

Antonin Rozsypal
Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504-4330
axr8951@cacs.louisiana.edu

Miroslav Kubat
Department of Electrical and Computer Engineering
University of Miami
1251 Memorial Drive, Coral Gables, FL 33146
mkubat@miami.edu

Abstract

Association mining explores algorithms capable of detecting frequently co-occurring items in transactions. A transaction can be identified with a market basket—a list of items a customer pays for at the checkout desk. In this paper, we explore a framework for the detection of changes in the buying patterns, as affected by fashion, season, or the introduction of a new product. We present several versions of our algorithm and experimentally examine their behaviors in domains with *gradually* changing domains.

Introduction

Let a database consist of transactions $T_1, T_2, \ldots, T_N$ such that $\forall i, T_i \subseteq I$, where I is a set of *items*. Let an *itemset*, X, be defined as a group of items such that $\exists i, X \subseteq T_i$. A *support* of itemset X is the number of transactions, T_i, containing this itemset. Suppose a user of an association-mining algorithm submits a minimum support, θ. A popular research issue is how to detect all itemsets whose support is at least equal to θ. We will call them *high-support itemsets*.

In a classical application, the transactions are identified with lists of items a customer pays for at the checkout desk (market baskets). A supermarket then places associated items on neighboring shelves, advertises them in the same catalogues, and avoids discounts on more than one member of the same itemset. However, the paradigm extends well beyond the realm of supermarket data. In the Internet environment, a transaction may consist of hyperlinks pointing to a web page, and high-support itemsets then signal associations among web sites (Noel, Raghavan, & Chu 2001). In a medical application, each transaction can summarize a patient's history, and association mining may seek co-occurring symptoms or ailments.

Here, we are interested in domains where the list of high-support itemsets is subject to changes in time, for instance as a result of fashion or seasonal impacts (Cheung & Han 1996; Pitkow 1997; Raghavan & Hafez 2000). We will assume the framework of *block evolution* (Ganti, Gehrke, & Ramakrishnan 2000) where a block of market baskets is periodically added to an existing database as individual stores report their daily business. The task is to *adapt* to this change and to measure the accuracy of this adaptation. To this end, we recently developed a novel algorithm (Rozsypal & Kubat 2002) and investigated its behavior in domains where the change is abrupt. However, this may be too much of a simplification. More often than not, the "Winter" buying patterns only gradually replace the "Fall" patterns. This is the scenario we address. Somewhat surprisingly, it turns out that, under this circumstance, different operators for change detection and for knowledge update are useful. We report a series of experiments illustrating the point.

Related Work

The basic task of association mining is to find all itemsets with support at least equal to a user-set minimum, $\theta\%$. The seminal paper by Agrawal et al. (Agrawal, Imielinski, & Swami 1993) introduces the algorithm Apriori that recursively creates itemsets of size k from those of size $k-1$. Since the algorithm is known to be slow when applied to large real-world databases, most of the research has focused on how to expedite the attendant computations—see, e.g., (Aggarwal & Yu 2002; Chen, Park, & Yu 1996; Nag, Deshpande, & DeWitt 1999). Some scientists have addressed some practical aspects of realistic applications, such as the need to handle generalized items, synonyms, and similarities (Han & Fu 1999). Also techniques for direct coupling of association mining to relational database management systems have been studied. Yet another research strand has focused on processing specific types of queries.

In our own work, we have been concerned with the fact that knowledge can vary in time. While some itemsets have high support throughout the entire database, others are important only during specific time intervals. Let us introduce the informal notion of a *context*, marked by "local" high-support itemsets that are less frequent elsewhere. The authors of (Cheung & Han 1996) describe an incremental technique that can be used in such environments—after the arrival of a block of market baskets, the list of high-support itemsets is updated. However, the mechanism is slow to react to big changes in context and is essentially incapable of detecting the changes with reasonable precision. The approach advocated by (Raghavan & Hafez 2000) goes one step further: each block of time-ordered market baskets is analyzed using the results from previous blocks. Although the approach has been shown to detect locally important

itemsets, its weakness is that it expects the user to specify the borders of the individual contexts—performance plummets if the specification is imprecise. Other systems of this kind were independently developed by (Brin *et al.* 1997) and (Ramaswamy, Mahajan, & Silberschatz 1998).

In the research reported here, we take inspiration from the concept learning algorithms from the FLORA-family (Kubat 1989; Widmer & Kubat 1996) that apply induction techniques only to a "window" of time-ordered examples. Every now and then, new examples are added and older ones get deleted. Each time the window contents change, the current description of the induced concept is updated so as to reflect the new circumstances. Practical implementations differ in how they adjust the window size and how they update the concept description (Harries, Sammut, & Horn 1998; Matwin & Kubat 1997). Similar idea was independently developed for the needs of data mining by (Ganti, Gehrke, & Ramakrishnan 2000) who determine the size of the window using their own mathematical model for measuring differences between datasets (Ganti, Gehrke, & Ramakrishnan 1999). In our recent paper (Rozsypal & Kubat 2002), we reported an alternative approach (more closely following the ideas behind FLORA) and demonstrated its feasibility in simple domains with *abrupt* context changes. Later, we carried out a series of experiments with the somewhat more realistic domains where the set of high-support itemsets was changing only *gradually.* Surprisingly, the results and conclusions differed from those presented in (Rozsypal & Kubat 2002) to an extent that merits a separate paper.

Association Mining with a Moving Window

The idea of a window that moves along a stream of market baskets (arriving in blocks) is illustrated by Figure 1. We assume that the context changes after several blocks. If the knowledge is updated after each block, the system can at each moment return the list of itemsets that currently have high-support. Our algorithm uses heuristics to detect a significant change in the environment. If such change is detected, the system decides how many outdated blocks should be removed from the window. Each change in the window then triggers an update of the list of itemsets. Heuristics are used to keep the window large during periods of stability and to narrow it when a context change occurs. The minimum support, θ_{MIN}, remains fixed throughout the entire experimental run. If the user specifies some $\theta > \theta_{MIN}$, it is easy to choose from the list of high-support itemsets those that satisfy θ.

Detecting the change by simple counting

Let L_W denote the list of itemsets with high-support in the window and let L_B denote the list of itemsets with high-support in a new block. Itemsets from L_B that are not found in L_W are called "emerged" itemsets. As already mentioned, a series of blocks is presented to the system, one by one. The number of emerged itemsets in the i-th block is denoted by NE_i (number of emerged). Conversely, high-support itemsets from L_W that are not found in L_B are "vanished" itemsets and the number of vanished itemsets in the i-th block is denoted by NV_i (number of vanished).

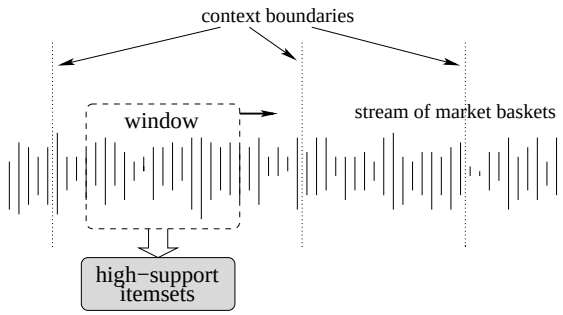

Figure 1: Association mining in time-varying domains. The system can "see" only the recent market baskets visible through the window.

In stable environments, NE_i and NV_i behave as random numbers drawn from normal distributions. To detect a change, the system has to determine whether the numbers of emerged and vanished itemsets in the new block differ *significantly* from what should be expected based on the underlying distribution. To merge the two distributions (emerged and vanished itemsets), our algorithm works with their geometric means $G_i = \sqrt{NE_i \times NV_i}$. The value of G_{new} for the newly added block is compared to the estimated parameters of G_i's distribution. The significance of the difference is evaluated using the one-tail t-test. The index i runs from 1 through M, where M is the number of blocks in the window.

Detecting the change by multivariate binomial distribution

Let f_I denote the relative frequency of itemset I. The probability that I is found in N market baskets is calculated from the binomial distribution that can be, for sufficiently large samples, approximated by the normal distribution with the mean $\mu_I = f_I$ and standard deviation $\sigma_I = \sqrt{\frac{f_I(1-f_I)}{n}}$. Let f_{IW} and f_{IB} denote the relative frequencies of itemset I in the lists L_W and L_B, respectively; and let n_W and n_B denote the numbers of market baskets in these two lists. If n_W and n_B are large, then $d_I = f_{IW} - f_{IB}$ can be approximated by the normal distribution with the following parameters.

$$\mu_I = f_{IW} - f_{IB}, \quad (1)$$

$$s_I = \sqrt{\frac{f_{IW}(1-f_{IW})}{n_W} + \frac{f_{IB}(1-f_{IB})}{n_B}} \quad (2)$$

If L_W and L_B come from the same source, then the next metric follows the normal distribution, $N(0,1)$, and we can use confidence intervals to decide whether the differences between the two frequencies are statistically significant:

$$z_I = \frac{f_{IW} - f_{IB}}{s_I} \quad (3)$$

Under the simplifying assumption that the occurrence of individual itemsets is pairwise independent, the last result can be generalized to the multiple-itemsets scenario,

where the distribution is characterized by the z_k-statistic (also called Penrose distance) defined by the following formula, where the index i refers to individual itemsets:

$$z_k = \sum_{i=1}^{k} \frac{(f_{iW} - f_{iB})^2}{\frac{f_{iW}(1-f_{iW})}{n_W} + \frac{f_{iB}(1-f_{iB})}{n_B}} \quad (4)$$

Note that this is, in effect, the sum of mean values divided by variances. The z_k-statistic has been shown to follow a χ^2 distribution with k degrees of freedom. This means that, for a set of k high-support itemsets, the fact that z_k satisfies the expression $z_k > \chi^2_{0.05;k}$ can be interpreted as evidence of a difference between the contexts that underly L_W and L_B.

Controlling the window size and updating the list of itemsets

If a context change is suspected, then the window is reduced by the removal of older, less relevant, blocks. We control the window size by the following heuristics:

1. `Harsh operator.` If a change is suspected, remove all blocks except for the latest block.
2. `Reluctant operator.` If a change is suspected, do nothing. Only if the change is detected in two consecutive blocks, remove all blocks except for the latest two blocks.
3. `Opportunistic operator.` If a change is suspected, remove 50% of the oldest blocks. If the change is confirmed after the next block addition, remove all the remaining blocks except for the latest two blocks.

Experiments

Experimental Setting

The technique has been tested on synthetic data produced by our own data generator. In the first step, it picks a random integer, N, to define the size of a market basket. Then, it generates N random integers to become the items in the basket. Random numbers are obtained from normal distributions with user-set parameters. Note that the experimenter can directly calculate the *theoretical support* of any itemset by taking the product of the relative frequencies of the items in the itemset.

Suppose the system returns a list of itemsets that have high support in the window. Let the theoretical support be denoted by f_{iT} and let f_{iR} denote the real support of the i-th itemset as measured by the association mining program. For a list of k itemsets, the error committed by the program is calculated by the following equation:

$$error = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(f_{iT} - f_{iR})^2} \quad (5)$$

The market baskets arrive at the system in blocks, each block consisting of 1,000 market baskets. We experimented with domains where pairs of contexts alternate: for 20 blocks, all market baskets come from the first context; for the next 20 blocks, the first context is gradually replaced with the second context; for the next 20 blocks, all market baskets come from the second context; for the next 20 blocks, the second context is gradually replaced with the first context; etc. By "gradual replacement" we mean that one block contains 5% market baskets from the new context and 95% from the old context; the next block will contain 10% market baskets from the new context and 90% from the old context; and so on, until all market baskets come from the new context.

Let f_{i1} and f_{i2} be the frequency of the i-th itemset in the first and second context, respectively; and let n_1 and n_2 be the respective numbers of market baskets in the two contexts. The difference between the two contexts is quantified by the z_k-statistics (Penrose distance):

$$d = \sum_{i=1}^{k} \frac{(f_{i1} - f_{i2})^2}{\frac{f_{i1}(1-f_{i1})}{n_1} + \frac{f_{i2}(1-f_{i2})}{n_2}} \quad (6)$$

We considered two domains: one with a relatively big difference between neighboring context ($d = 252$) and one with a relatively small difference ($d = 54$). For each of the domains, we experimented with two different heuristics for context change detection ("simple counting" and "multivariate binomial") and three different operators controlling window size (`harsh`, `opportunistic`, and `reluctant`). For each domain, we generated 10 random sequences of market baskets (taken from the alternating contexts as described above) and ran the algorithm on each of them. The results reported in the following subsection are obtained as average values from these 10 runs.

Experimental Results and Discussion

The results are summarized in Figure 2, where the horizontal axis represents the number of blocks presented to the system and the vertical axis represents the error as calculated by Equation 5. The upper row contains charts for the domain with the big difference between neighboring contexts and the lower row contains charts for the domain with the small difference between neighboring contexts. Each graph contains two curves: the dotted one plots the case where the change is detected by the simple counting (`SC`), and the solid curve plots the case where the multivariate binomial (`MB`) distribution is used. For each domain, we used three different operators to adjust the window size: `harsh`, `reluctant`, and `opportunistic`.

For any window-size operator, the error in the periods of change (between the 20th and 40th blocks, then between the 60th and 80th blocks, etc.) is lower when the `SC` change detector is used, no matter whether the difference between neighboring contexts is big or small. The `MB` detector outperforms `SC` only in relatively stable periods. Since the `MB` operator was better than `SC` in domains with abrupt change as observed in (Rozsypal & Kubat 2002), we have to ask about the cause of this phenomenon.

To provide some insight, Figure 3 shows how the window size evolves in time. The reader can see that the `SC` detector is nearly always better at recognizing a change at a very early stage—the window immediately shrinks. On the other hand, `MB` outperforms `SC` in stable periods because

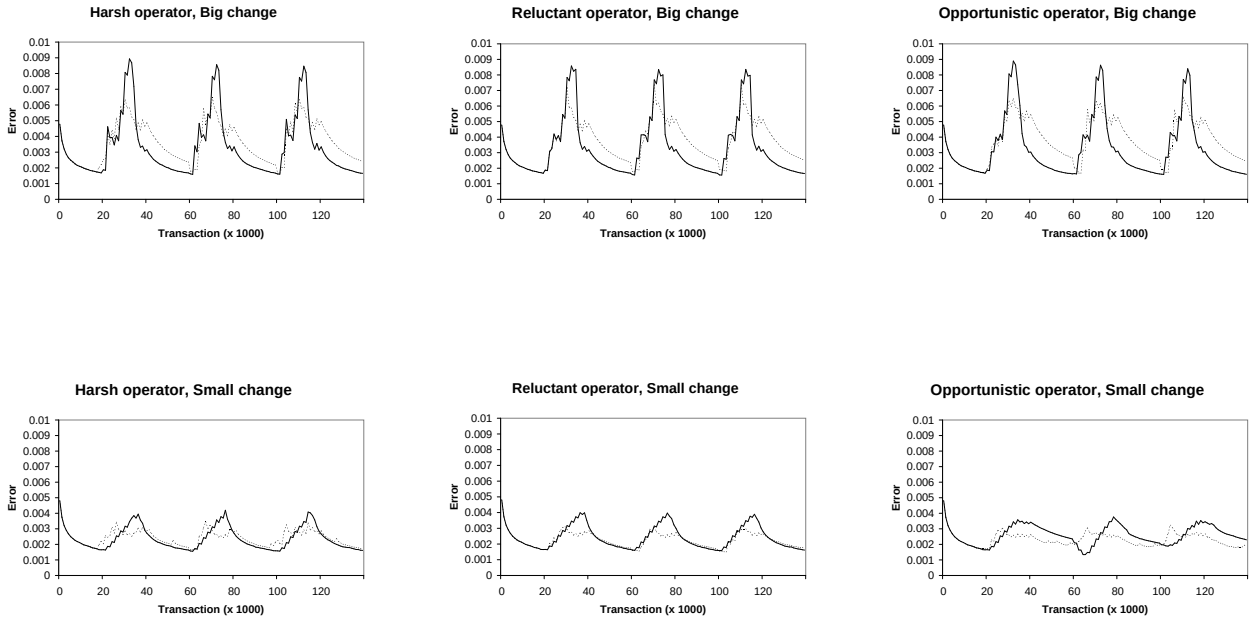

Figure 2: Alternating contexts. The dotted curves represent the error rate for the "simple count" heuristic, the solid curves represent the error rate for the "multivariate binomial" heuristic.

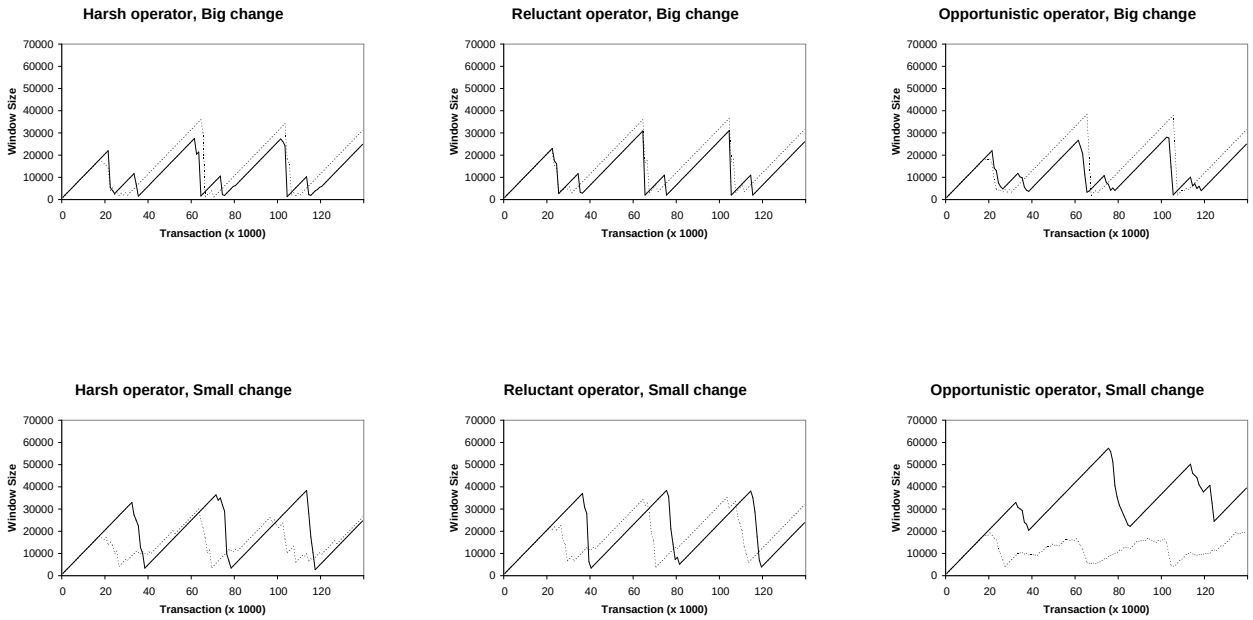

Figure 3: The changing size of the window in the two domains, as affected by the three window-size heuristics.

it helps remove the remnants of the transition period from a stable period. It appears that the MB detector sometimes treats the transition period as a specific sort of context and tends to ignore the change until it really becomes conspicuous. This is particularly pronounced in the small-difference change (right column), where the MB detector appears to ignore the change till the very last moment.

Among the window-size operators, no winner can be determined as long as the difference between neighboring contexts is big (only the harsh operator appears to have a slight edge). When the change is small, the opportunistic operator leads to more stable error with the SC detector, and also the window size changes less dramatically. The SC detector thus seems to be more sensitive to changes than the MB detector. Inevitably, this makes SC more sensitive to false alarms, which is why its behavior was less satisfactory in the experiments from (Rozsypal & Kubat 2002).

Conclusions

The paper reports our experience with FLORA-based association mining in domains with gradual changes. We investigated the behavior of two alternative operators for change detection (SC and MB) and the behavior of three operators for window-size reduction (harsh, opportunistic, and reluctant).

Contrary to the results from (Rozsypal & Kubat 2002), the change detector SC seems to be more appropriate here than MB because it reacts faster to changes. This, however, can be detrimental in noisy domains because the noise can be misinterpreted for a change. As for window-size control, all of the three operators led to about the same error in itemset detection. However, the opportunistic operator appears to have somewhat more stable behavior than the other two operators, especially in domains with a small change between neighboring contexts.

References

Aggarwal, C. C., and Yu, P. S. 2002. A new approach to online generation of association rules. *IEEE Transactions on Knowledge and Data Engineering* 13:527–540.

Agrawal, R.; Imielinski, T.; and Swami, A. 1993. Mining association rules between sets of items in large databases. In *Proceedings of ACM SIGMOD*, 207–216.

Brin, S.; Motwani, R.; Ullman, J. D.; and Tsur, S. 1997. Dynamic itemset counting and implication rules for market basket data. In Peckham, J., ed., *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*, 255–264. Tuscon, Arizona, USA: ACM Press.

Chen, M. S.; Park, J. S.; and Yu, P. S. 1996. Data mining for path traversal patterns in a web environment. In *Sixteenth International Conference on Distributed Computing Systems*, 385–392.

Cheung, D. W., and Han, J. 1996. Maintenance of discovered association rules in large databases: An incremental updating technique. In *Proceedings of the 12th International Conference on Data Engineering, New Orleans, Louisiana*.

Ganti, V.; Gehrke, J.; and Ramakrishnan, R. 1999. A framework for measuring changes in data characteristics. In *Eighteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 126–137.

Ganti, V.; Gehrke, J.; and Ramakrishnan, R. 2000. Demon: Mining and monitoring evolving data. In *Proceedings of the 16th International Conference on Data Engineering*. San Diego, California, USA: IEEE Computer Society.

Han, J., and Fu, Y. 1999. Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge Engineering* 11:798–805.

Harries, M. B.; Sammut, C.; and Horn, K. 1998. Extracting hidden context. *Machine Learning* 32:101–126.

Kubat, M. 1989. Floating approximation in time-varying knowledge bases. *Pattern Recognition Letters* 10:223–227.

Matwin, S., and Kubat, M. 1997. The role of context in concept learning. In *Workshop Notes of the Workshop on Learning in Context-Sensitive Domains*, 1–5.

Nag, B.; Deshpande, P. M.; and DeWitt, D. 1999. Using a knowledge cache for interactive discovery of association rules. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 244–253.

Noel, S.; Raghavan, V. V.; and Chu, C. H. 2001. Visualizing association mining results through hierarchical clusters. In *Proceedings of the 2001 International Conference on Data Mining (ICDM-01)*, 425–432.

Pitkow, P. 1997. In search of reliable usage of data on the www. In *Proceedings of the 6th International WWW Conference*.

Raghavan, V. V., and Hafez, A. 2000. Dynamic data mining. In *Proceedings of the 13th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE*, 220–229.

Ramaswamy, S.; Mahajan, S.; and Silberschatz, A. 1998. On the discovery of interesting patterns in association rules. In *The VLDB Journal*, 368–379.

Rozsypal, A., and Kubat, M. 2002. Association mining in time-varying domains. (manuscript).

Widmer, G., and Kubat, M. 1996. Learning in the presence of concept drift and hidden contexts. *Machine Learning* 23:69–101.