An Algorithm to Discover Calendar-based Temporal Association Rules with Item's Lifespan Restriction

Geraldo Zimbrão
Jano Moreira de Souza
Victor Teixeira de Almeida
Wanderson Araújo da Silva

Computer Science Department, Graduate School of Engineering
Federal University of Rio de Janeiro
PO Box 68511, ZIP code: 21945-970 Rio de Janeiro - Brazil,
Email: {zimbrao,jano,valmeida,wanderson}@cos.ufrj.br

Abstract: *Mining association rules in a market basket database is a well stated problem, and there are a number of approaches to deal with this problem. However, there are many circumstances where the current techniques are not adequate. An interesting one is to mine rules about items that have seasonal selling rate, which is called calendar-based association rule mining. Another one is to mine association rules about items introduced into or removed from the database, that is, to consider the items' lifespan. Although there are some works about these problems, no one considers both problems together – but in real market basket databases it is crucial to consider both the items' lifespan and its seasonality. This work states this problem in detail, and we present an example where other algorithms fail to discover important (quite evident) association rules. Also, we present an extended version of the calendar-based algorithm to mine these kinds of association rules.*

1. Introduction

Mining association rules in a market basket database is a well stated problem, and there are a number of approaches to deal with it. Even though the main problem is well defined and lots of algorithms to solve it exist in the literature, in real databases, some particularities are not handled by these common algorithms. If these particularities were not handled, two big problems may occur: first, important association rules may not be considered by the algorithm; and second, the algorithm would be optimized taking into account these particularities, achieving a better performance.

One of these particularities are the items that are being sold according to some time cycle or any repetition pattern, like products that are better sold on summer, or even association rules that are important in one season and are not in any other. Besides that, some items, because of logistic or commercial purposes, have periods of suspended selling.

Finally, the databases tend to cover a very long time interval (specially large databases, the main focus of the association rule mining) that the association rules containing recent products will only be discovered where they surpass a specified threshold (minimum support). In an analogous way, association rules that involve products being no more commercialized will tend to disappear as results of the association rule mining algorithms.

This work proposes a solution to handle the temporal dimension in the association rule mining problem. We are especially interested in handling the seasonality but also the time intervals (lifespan) of the items. In this way, this work presents an approach to extend the existing calendar-based algorithms to deal with the items' lifespan.

This paper is organized as follows: section 2 defines precisely the problem for what we are proposing a solution. The section 3 presents the related work found in the literature. In section 4 the problems with the works in the literature are stated and our solution is proposed. Section 5 concludes the paper and proposes some future work.

2. Defining the Problem

The problem of discovering association rules was first presented in [AIS93] and then extended in [AS94] for solving the market basket data analysis problem. Let *c* be a set of data items. Let *D* be a transaction database. A transaction *t* in the database is a subset of *c*, i.e., *t Í c*. We say that a transaction *t* contains or satisfies *X*, a set of items in *c*, if *X Í t*. An association rule is an implication of the form *X Þ Y* where *X* and *Y* are disjoint sets of items in *c*, i.e., *X Í c*, *Y Í c* and X ∩ Y = ϕ. The confidence *c* of the rule means that *c%* of the transactions in *D* that satisfy *X* also satisfy *Y*. The support *s* of the rule means that *s%* of the transaction in *D* satisfy X ∪ Y. While confidence is a measure of the rule's strength, support corresponds to statistical significance. Let σ_X the number of transactions in *D* that satisfy *X*. The support and confidence of a rule can be stated as above:

$$s = \frac{s_{X \cup Y}}{|D|} \qquad c = \frac{s_{X \cup Y}}{s_X}$$

The problem of mining association rules in a market basket database is, for a given confidence and support thresholds *minsupport* and *minconfidence*, to find all the association rules that have confidence and support greater than these corresponding thresholds. This problem can be divided into two sub-problems: (a) find all sets of items (itemsets) that have support greater than *minsupport*, which are called large itemsets; (b) generate the association rules using the large itemsets calculated in (a) respecting the *minconfidence* threshold. The first step is much more expensive than the second and a good algorithm is presented in [AS94], therefore, the problem of mining association rules can be reduced to the problem of finding all large itemsets, i.e., find the set $L = \{X | X \subseteq c \wedge s_X \geq s \times |t|\}$.

This initial work of discovering association rules and several efforts on extend it ([BMUT97], [SON95], [HPY00], [ZPOL97], and more) did not take time into consideration. Several problems of omitting the time dimension of a transaction database may occur.

One example of such problems stated in [ORS98] is a database maintained by a supermarket, where an association rule might be of the form "*beer Þ chips* (support: *3%*, confidence:*87%*)", which means that *3%* of all database transactions contain the data items beer and chips, and *87%* of the transactions that have the item "beer" also have the item “chips” in them. It may be the case that beer and chips are sold together primarily between *6PM* and *9PM*. Therefore, if we segment the data over the intervals *7AM–6PM* and *6PM–9PM*, we may find that the support for the beer and chips rule jumps to *50%*. It is known that beer is better sold on the summer, so if we segment again the data according to its season, the support for the rule “*beer Þ chips*” may now jump to more than *55%* in the summer and at time interval *6PM–9PM*.

Another example is that items do not necessarily exist through all the time of the database. Items may have been discontinued and logically items may have just been recorded. These items have no chance to appear as a result of traditional algorithms to discover association rules because of their low support, even if they are important during their existence.

According to these examples, the algorithms to discover association rules that do not take the transaction time information into consideration may be omitting several important rules.

3. Related Work

Much work in the discovery of common sequences or patterns of events may be found in the literature ([AS95], [MTV97], [SA96], [BWJL98]). However, these algorithms were designed to find commonly occurring sequences rather than association rules.

In order to discover temporal association rules, the work in [ORS98] stated very clearly the problem of omitting the time dimension. It is assumed that the transactions in the database are timestamped and the time interval is specified by the user to divide the data into disjoint segments, like months, weeks, days, etc. The cyclic association rules are defined as association rules that hold the minimum confidence and support at specific regular time intervals. Using this definition, a rule may not have high support and confidence for the entire transactional database, but only for transactional data in a particular periodic time interval. Several algorithms and optimization were shown to be efficient through an experimental analysis. A disadvantage of these algorithms is the fact that it does not deal with multiple granularities of time intervals ([BJW00]). As an example of a calendar pattern that cannot be represented by cycles is the simple concept of the first working day of every month.

The problem stated below was perceived by the authors in [RMS98] and then extended. They introduced the notion of calendar algebra, based on the work in [A85], [LMF86] and the implementation reported in [CSS94], to describe phenomena of interest in association rules. The calendar algebra define a set of time interval that the algorithm considers to find the association rules, i.e., a calendar *C* is a set of (possibly interleaved) time intervals $\{(s_1, e_1), (s_2, e_2), \ldots, (s_k, e_k)\}$, where (s_i, e_i) is a time interval starting at time s_i and ending at time e_i. Such rule is called calendric if it has the minimum support and confidence during every time unit contained in the calendar. The problem is then to find calendric association rules. This work in [RMS98] is the first to propose the notion of finding fuzzy patterns in association rules, i.e., finding the patterns that approximately match the user-defined patterns. The only disadvantage of this approach is that the user must have prior knowledge about the temporal patterns in the transactional database to define the calendars.

Another work using calendars to find association rules may be found in [LNWJ01]. The main difference between both of the approaches is that this work requires less prior knowledge of the database by the user. It is stated in this paper that using the representation mechanisms proposed in [LMF86] or [BJW00] calendar schemas for both cyclic and user defined calendar patterns can be achieved. The user needs only to give a simple calendar-based pattern, or just calendar pattern for short, belonging to a calendar schema. As an example, found in [LNWJ01], a calendar schema may be $(year : \{1995, 1996, \ldots, 1999\}, month : \{1, 2, \ldots, 12\}, day : \{1, 2, \ldots, 31\})$. The user may want to find all association rules that is valid for the calendar pattern $\langle 1999, 12, * \rangle$ which means "every day of December, 1999". The wild card "*"may be replaced by the word "every". This paper presents two algorithms for finding calendar-based association rules, a precise match and a fuzzy match. Two optimization techniques are proposed and an experimental evaluation using both real and synthetic data shows the good performance of the algorithms proposed.

In [AR00], another problem of the lacking of temporal information for discovering association rules is stated. The problem is that the support must not consider the time when the items were not present in the database. Thus, it is proposed an approach that computes the association rules with the minimum support considering only the items' life time.

The main idea in this work is that every item in the transaction database has a lifespan *[t_1, t_2]* which it is active. Let $T = \{K ,t_0,t_1,t_2,K\}$ be a set of time instants, countable and infinite, a linear order $<_T$ is defined, where $t_1 <_T t_2$ means that t_1 occurs before than t_2. The lifespan of an item *X* is represented by a closed interval *[t_i, t_j]*, where $t_i <_T t_j$ is denoted by l_X. If *d* is the transaction database, then d_{lx} and |d_{lx}| are the set and the number of transactions whose timestamps $t_i \in l_X$ respectively.

Thus, the support of an item *X*, denoted by *s(X,d)*, is modified to take into consideration the items' lifespan. The denominator is no more the number of transactions of the entire database |d|, but the number of transactions executed where the item X were active, i.e., |d_{lX}|.

In order to filter the items with short life, the temporal support is then defined. It is the amplitude of the lifespan of an itemset. The lifespan of an itemset is the intersection of the items' lifespan of the itemset, i.e., if *Y* = *{I_1, I_2, …, I_n}* is an itemset, then l_Y=l_{I1}Ç l_{I2}Ç…Ç l_{In}. A threshold for the temporal support is also defined as a fraction of |l_d|, where |l_d| is the duration of the database *d*. Another concept to filter the items in the algorithm is the obsolescence. An item whose lifespan is *[t_1, t_2]* is obsolete at a specified time instant t_o if t_2<t_o.

An itemset is then considered relevant in a transactional database if it satisfies both the minimum temporal support and the minimum support thresholds. An algorithm based on the Apriori [AS94] is presented to discover these kind of temporal association rules.

Other works on finding temporal association rules may be found in the literature. These works include [CP98], [CP99] and [RR99], but can be considered complementary ones.

Finally, a bibliography of temporal data mining can be found in [RHS00] and a good survey of temporal data mining methods including temporal association rules in [AO01].

4. Our Approach

In the literature presented below, we can divide the problem of finding temporal association rules into two subcategories. The first one that tries to discover association rules during some cycle or temporal pattern like a calendar, for example. The works in [ORS98], [RMS98] and [LNWJ01] may be included in this category. The latter one is, at the best of our knowledge, the most complete work on this category. The other category tries to find association rules during the life time of the itemsets. The work in [AR00] may be included in this category.

In [LNWJ01], it is stated that the work in [AR00] is quite different since they are interested in association rules for calendar patterns instead of the life time of the items. In our point of view, these works not only substantially differ from each other, but may be viewed as complementary works, otherwise some important temporal association rules may be lost.

To clarify this idea, let us examine the KDD Cup 2000 [K+00] data set used in [LNWJ01] to find calendar-based association rules. The KDD Cup 2000 database contains clickstream and purchase data from the site Gazelle.com, a legwear and legcare web retailer that closed their online store on 8/18/2000. The clickstream database associates WWW users to the resources received from web servers. As said in [A+01], the data collection may be done at the application server layer (not web server) in order to support logging of data and metadata that is essential to the discovery process. As everybody knows, the resources in the WWW are very volatile. They appear and disappear with high frequencies.

The requests recorded in the clickstream data file from KDD Cup 2000 are from January 30, 2000 to March 31, 2000, which cover 8 weeks plus 6 days. Let us follow the calendar-based association rule mining in [LNWJ01], where they use timeOfDay to represent the calendar concept formed by partitioning each day into three parts: early morning (0am - 8am), daytime (8am - 4pm), and evening (4pm - 12pm). Let a HTML file H1 be created since Feb 6, 2000 (i.e. created in the second week of the clickstream recording), and be widely accessed together with an older one (H2) created before the clickstream recording. Imagine that H1 has a higher page views at morning, so the algorithm may return the temporal association rule "H1 $\Rightarrow$ H2 <Morning,*,*>".

The problem with the algorithm presented in [LNWJ01] is the creation of star calendar patterns given non-star ones (basic time intervals), once all of the non-star calendar patterns covered by the star one may be found in the database. Once the HTML pages H1 and H2 are not viewed together in the first week of the recording, the algorithm will not create the <Morning,*,*> but only <Morning,*,March> star calendar pattern.

In our proposal, we will include the lifespan information into the calendar-based association rule mining algorithm to solve this problem. Our proposed algorithm would return the rule "$H_1 \Rightarrow H_2$ <Morning,*,*> [02/06/2000, 03-31/2000]", where [02/06/2000, 03-31/2000] is the lifespan of the itemset $H=(H_1, H_2)$.

Therefore, we propose a new approach that will take into consideration the two categories of algorithms to discover association rules, i.e., a calendar-based algorithm with items' (or itemsets') lifespan restrictions. This approach intends to solve the problem stated above.

The proposal of the algorithm is to change the Lemma 2 and the Phase III of the algorithm in [LNWJ01] to discover calendar-based association rules, including the itemset's lifespan restriction, i.e., the Lemma 2 should now be rephrased: "given a star calendar pattern *e*, an itemset *X* is large for w.r.t. precise match only if it is large w.r.t. precise match for all 1-star calendar patterns covered by *e* that intersects the itemset's lifespan l_X".

Our propose is to change the Update procedure inside the Phases 0 and III. The original one updates the set $L_k(e)$ of large *k*-itemsets for the calendar pattern *e* by intersecting it with the set $L_k(e_0)$ large *k*-itemsets for the basic time interval e_0, i.e., $L_k(e)= L_k(e) \cap L_k(e_0)$. In the new version of this algorithm the update is done by $L_k(e)=L_k(e) \cap \{\forall X \in L_k(e_0) \mid l_X \cap e_0 \neq \emptyset\}$. This guarantees that the algorithm will only search for the non-star calendar patterns e_0 that belongs to the lifespan of the itemset l_X. This improvement can be seen in the Algorithm 1 below.

```
//Phase 0: Initial update for star calendar patterns of 1-itemsets
forall basic time intervals $e_0$ do
begin
   $L_1(e_0)$ = {large 1-itemsets in $T[e_0]$ with their lifespans
   forall star patterns e that cover $e_0$ do
      update $L_1(e)$ using $L_1(e_0)$ respecting the itemsets' lifespan
end

for(k=2; exists a star calendar pattern e such that $L_{k-1}(e)$ ? Ø; k++)
begin
   forall basic time intervals $e_0$ do
   begin
      //Phase I: generate candidates
      generate candidates $C_k(e_0)$

      //Phase II: scan the transactions
      forall transaction t in $T[e_0]$ do
            subset($C_k(e_0)$,t) // c.count++ if $c \in C_k(e_0)$ is contained in t
      $L_k(e_0)$ = { $c \in C_k(e_0)$ | c.count = minsupport

      //Phase III: update for star calendar patterns
      forall star patterns e that cover $e_0$ do
         update $L_k(e)$ using $L_k(e_0)$ respecting the itemsets' lifespan
   end
   Output <$L_k(e)$,e> for all star calendar pattern e
end
```

Algorithm 1: Algorithm to find the large itemsets

5. Conclusions and Future Work

This work presented a new approach to solve the association rule mining problem handling the temporal dimension, i.e., the temporal association rule mining problem. The problem was stated in detail and it was shown an example where the existing algorithms fail to mine quite evident temporal association rules, which justifies the need of a new approach.

We have proposed a new algorithm that finds the rules that other temporal association rule algorithms do not in the example given. The algorithm was presented in detail, and was shown how it mines the seasonal rules handling the items' lifespan.

As future work, we are planning to implement the algorithm and execute some tests to analyze the performance of the algorithm and also to show how it find the rules that the others do not.

Bibliography

[A85] J. F. Allen. "Maintaining Knowledge about Temporal Intervals". In "Readings in Knowledge Representation", pages 509–521. Morgan-Kaufman Publishers, Inc., 1985.

[AIS93] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In Proc. of the 1993 Int'l Conf. on Management of Data, pages 207–216, 1993.

[AO01] Cláudia M. Antunes, Arlindo L. Oliveira: Temporal Data Mining: an overview. Workshop on Temporal Data Mining - 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001.

[AR00] J.M. Ale and G.H. Rossi. An approach to discovering temporal association rules. In Proc. of the 2000 ACM Symposium on Applied Computing, pages 294–300, 2000.

[AS94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Proc. of the 1994 Int'l Conf. on Very Large Data Bases, pages 487–499, 1994.

[AS95] Agrawal, R., Srikant, R.: Mining Sequential Patterns. International Conference on Data Engineering (ICDE), March, Taipei, Taiwan (1995).

[A+01] S. Ansari, R. Kohavi, L. Mason, Z. Zheng, Integrating E-Commerce and Data Mining: Architecture and Challenges, Proceedings of the 2001 IEEE International Conference on Data Mining (ICDE), San Jose, California, USA, pp. 27-34, 2001

[BJW00] C. Bettini, S. Jajodia, and X.S. Wang. Time granularities in databases, data mining, and temporal reasoning. Springer-Verlag, 2000.

[BMUT97] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In Proc. of ACM SIGMOD Int'l Conf. on Management of Data, pages 255–264, 1997.

[BWJL98] Bettini, C-Wang, X. Jajodia, S. Lin, J.: Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences. IEEE TOKDE Vol. 10 N ° 2: 222-237. April 1998.

[CP98] X. Chen and I. Petrounias. A framework for temporal data mining. In Proc. of the 9th Int'l Conf. on Database and Expert Systems Applications, pages 796–805, 1998.

[CP99] X. Chen and I. Petrounias. Mining temporal features in association rules. In Proc. of the 3rd European Conf. on Principles and Practice on Knowledge Discovery in Databases, pages 295–300, 1999.

[CSS94] R. Chandra, A. Segev, and M. Stonebreaker. Implementing calendars and temporal rules in next generation databases. In International Conference on Data Engineering, Houston, Texas, USA, February (1994).

[HPY00] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. Naughton, and P. A. Bernstein, editors (2000).

[K+00] R. Kohavi, C. Brodley, B. Frasca, L. Mason, Z. Zheng, KDD-Cup 2000 Organizers' Report: Peeling the Onion, SIGKDD Explorations, v. 2, n. 2, pp 86-98, 2000. http://www.ecn.purdue.edu/KDDCUP

[LMF86] B. Leban, D. McDonald, and D. Foster. A representation for collections of temporal intervals. In Proc. of AAAI-1986 5th Int'l Conf. on Artifical Intelligence, pages 367–371, 1986.

[LNWJ01] Y. Li, P. Ning, X. S. Wang, and S. Jajodia. Discovering calendar-based temporal association rules. In Proc. of the 8th Int'l Symposium on Temporal Representation and Reasoning (2001).

[MTV97] Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. Report C-1997-15, University of Helsinki, Department of Computer Science, February (1997).

[ORS98] B. Özden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. In Proc. of the 14th Int’l Conf. on Data Engineering, pages 412–421, 1998.

[RHS00] John F. Roddick, Kathleen Hornsby, Myra Spiliopoulou: An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research. TSDM 2000: 147-164.

[RMS98] S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. In Proc. of the 1998 Int’l Conf. on Very Large Data Bases, pages 368–379, 1998.

[RR99] C. P. Rainsford and J. F. Roddick. Adding temporal semantics to association rules. In Proc. of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, pages 504--509. Springer, 1999.

[SA96] Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. Fifth International Conference on Extending Database Technology (EDBT), March, Avignon, France.(1996).

[SON95] A. Savasere, E. Omiecinski, and S. B. Navathe. An e#cient algorithm for mining association rules in large databases. In Proceedings of 21th International Conference on Very Large Data Bases, pages 432--444. VLDB, Sept. 11-15 (1995)

[ZPOL97] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In Intl. Conf. on Knowledge Discovery and Data Mining, August 1997.