ISSN: 2321-7782 (Online)

Volume 4, Issue 1, January 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Discovering Acyclic Patterns

Fokrul Alom Mazarbhuiya
College of Computer Science & IT
Albaha University,
Kingdom of Saudi Arabia (KSA)

Abstract: Discovering various patterns from supermarket datasets is an important data mining problem. One of such patterns is to find locally frequent patterns. Mahanta et al proposes an algorithm for extracting such local patterns where each pattern is associated with a sequence time intervals in which it is frequent. The sequence of time intervals associated with a pattern may provide us some interesting results, for example, the pattern may be acyclic. In this paper we propose a method of finding such acyclic patterns. The efficacy of the method is established through experimental results.

Key words: Itemsets; Frequent Itemsets; Local Frequent Itemsets; Local Association rules; Sequence of time intervals; Cyclic Frequent Itemsets.

I. INTRODUCTION

Agrawal *et al* [1] first formulated the problem of association rules mining for the application in large super markets. The supermarket transaction is normally temporal. Mining temporal dataset is also an important data mining problem. Ale and Rossi [2] propose a method of extracting frequent sets from such datasets. In [3, 4]. author proposes an algorithm to find locally frequent itemsets. The method proposed by them finds all frequent itemsets where each frequent itemset is associated with a sequence of time intervals. The sequence of time intervals associated with a frequent itemsets can be used to find some interesting results. For example the frequent itemsets maybe cyclic or acyclic. In this paper we propose to study the problem of acyclic nature of the sequence of time intervals associated with a frequent itemset and propose a method to extract all frequent itemsets which are acyclic. In such case, we define the equality between time intervals associated with a locally frequent itemsets as follows: two time intervals are said to be almost equal if their lengths are equal up to a small variation otherwise they are unequal. A frequent itemset is said to be acyclic if the lengths of time intervals associated with it is almost equal but at least any two of their time gaps is unequal.

The paper is organized as follows. In section II we give a brief discussion on the related work from the literature. The problem definition is described in section III. In section IV, we describe the proposed method for mining acyclic frequent sets. The experimental results and findings are described in section V. The conclusion and suggestions for future work are given in section VI.

II. RELATED WORK

The association rules mining problem is first formulated by Agrawal *et al* [1]. In [5], author proposed a method for extracting association rules called *A priori algorithm*. Mining Temporal Data is an extension of conventional data mining. An example of this is temporal association rule mining. In temporal association rules each rule has associated with it a time interval in which the rule holds. The problem of temporal data mining is addressed extensively by different researchers [2, 6, 7, 8]. In [3], authors proposed a method of finding locally frequent sets where each itemset is associated with a list of time intervals where the itemset is frequent. In [9], authors proposed a method of discovering calendar-based association rules. In [10], authors proposed a method of discovering cyclic itemsets where the cyclicity is defined in terms equality among the intervals of frequency associated with an itemset as well as that of the time gaps. Here time gap is the gap between two consecutive time intervals of frequency associated with the frequent itemset. The work done in [10] is an extension of work in [3]. Similar works were done in [11].

III. Problem Definition

Let $T = <t_o, t_1,..>$ be a sequence of time-stamps over which a linear ordering < is defined where $t_i < t_j$ means t_i denotes a time which is earlier than $t_{j..}$ Let I denote a finite set of items and the transaction dataset D is a collection of transactions where each transaction has a part which is a subset of the item set I and the other part is a time-stamp indicating the time in which the transaction had taken place. We assume that D is ordered in the ascending order of the time-stamps. For time intervals we always consider closed intervals of the form $[t_{1,} t_2]$ where t_1 and t_2 are time-stamps. We say that a transaction is in the time interval $[t_1, t_2]$ if the time-stamp of the transaction say t is such that $t_1 \le t \le t_{2.}$

We define the local support of an item set in a time interval $[t_1, t_2]$ as the ratio of the number of transactions in the time interval $[t_1, t_2]$ containing the item set to the total number of transactions in $[t_1, t_2]$ for the whole dataset D. We use the notation $Supp_{[t_1,t_2]}(X)$ to denote the support of the item set X in the time interval $[t_1, t_2]$. Given a threshold σ we say that an item set X is frequent in the time interval $[t_1, t_2]$ if $Supp_{[t_1,t_2]}(X) \ge (\sigma/100)*$ tc where tc denotes the total number of transactions in D that are in the time interval $[t_1, t_2]$. We say that an association rule $X \Rightarrow Y$, where X and Y are item sets holds in the time interval $[t_1, t_2]$ if and only if given threshold τ, $Supp_{[t_1,t_2]}(X \cup Y) / Supp_{[t_1,t_2]}(X) \ge \tau / 100.0$ and $X \cup Y$ is frequent in $[t_1, t_2]$. In this case we say that the confidence of the rule is τ.

A. Almost equal intervals

For each locally frequent item set extracted by algorithm [3, 4], a list of time intervals is kept in which the set is frequent where each interval is represented as [*start*, *end*] where *start* gives the starting time-stamp of the time interval and *end* gives the ending time-stamp of the time-interval. *end* – *start* gives the length of the time interval. Given two intervals [$start_1$, end_1] and [$start_2$, end_2] if the intervals are non-overlapping and $start_2 > end_1$ then $start_2 - end_1$ gives the distance between the time intervals. Similarly two intervals [$start_1$, end_1] and [$start_2$, end_2] are said to be *almost equal* in length if the length of the both intervals are equal up to a small variation say δ% i.e. (end_1- $start_1$)± δ% of (end_1- $start_1$) is equal to (end_2- $start_2$) or (end_2- $start_2$) ± δ% of (end_2- $start_2$) is equal to (end_1- $start_1$) where δ is specified by user. Otherwise they are said to be *unequal* in length. Similarly we define the equality of the time gaps of the consecutive intervals.

IV. Proposed Method

To extract acyclic frequent sets we find the time gap between any two consecutive frequent time intervals of the same set. If the time gaps between consecutive intervals are found to be *unequal* in length and also the lengths of the frequent intervals are found to be *almost equal* then we call these frequent sets as acyclic frequent sets. Now to find out such type of acyclicity for each frequent item set we proceed as follows. If the first frequent interval is *almost equal* in length with second frequent interval then we see whether the time gap between the first and the second time interval is *unequal* in length with the time gap between the second and third periods. If it is, then we check whether any of the two time gaps is *unequal* with the time gap between the third and the fourth periods. If the average length of the first two intervals of frequency is *almost equal* in length with the third interval of frequency, we proceed further or otherwise stop. In general if the average lengths of the first (n-1) frequent intervals is *almost equal* to the length of the n-th frequent interval and the any of the first (n-2) time gaps is *unequal* to the (n-1)th time gap, then the average of n frequent intervals is compared with (n+1)th frequent interval and that of any of the the first (n-1) time gaps is compared with the n-th time gap. This way we can extract acyclic patterns if such patterns exist. We can have another kind of acyclic patterns where the time intervals of frequency associated with an itemset as well as their time gap are unequal.

V. Results and Discussion

For experimental purpose we take a synthetic dataset generated through the program provided by the *Quest research group* at IBM Almaden. available on http://www.almaden.ibm.com. For experiment, we take first 20000 transactions. A program was written to incorporate temporal features in the dataset. The program takes as input a starting date and two values for the minimum and the maximum number of transactions per day. A number between these two limits are selected at random and that many consecutive transactions are labeled with the same date to reflect the fact that many transactions have taken place on that day. This process starts from the first transaction to the end by marking the transactions with consecutive dates (assuming that the market remains open on all week days). The process is repeated for first 40000, 60000, 80000, 100000 transactions to generate the datasets of different sizes. We have given the maximum number of transactions and minimum number of transactions in such a way that the lifetime of each size of dataset is almost one year. In the figure1 and figure 2 given below we describe the results obtained from the experiments in graphical form.

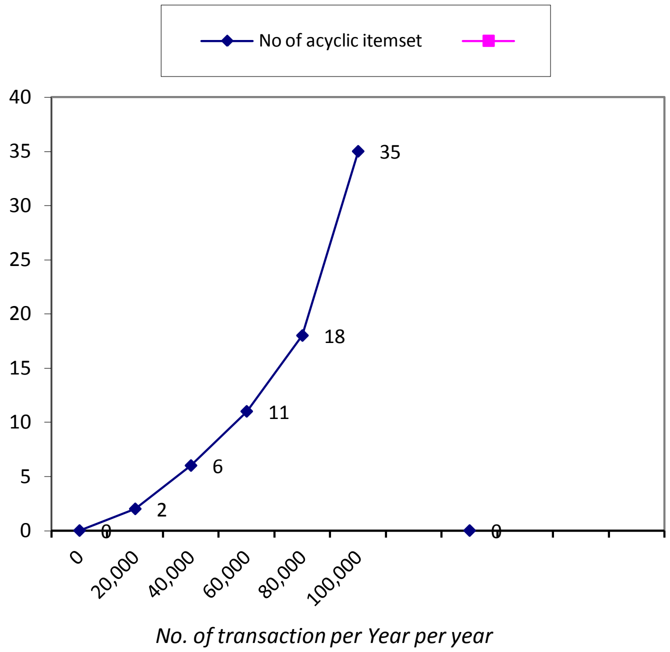


Fig.1: No. of transactions vs. No. of acyclic itemsets

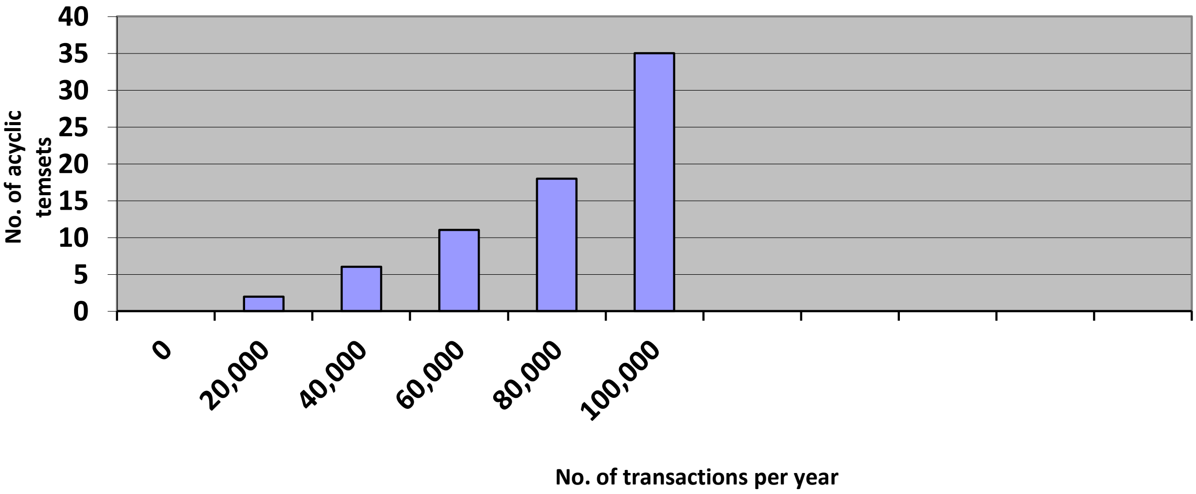


Fig.2: No. of transactions vs. No. of acyclic itemsets

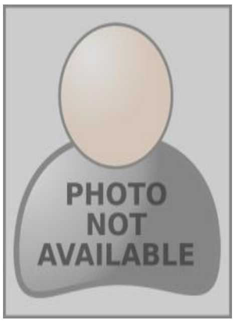
VI. Conclusions

An algorithm for finding acyclic frequent sets from temporal data is given in the paper. The algorithm discussed in [3, 4] gives all locally frequent itemsets where each frequent itemsets is associated with a list of time intervals where it is frequent. Our algorithm takes the input from the result of the algorithm proposed by [3, 4] and supplies all frequent sets which are acyclic in nature. Our resulting acyclic frequent itemsets are having at least two time gaps are *unequal* in length but the duration of the intervals of frequency are *almost equal*. In future, we may also modify our algorithm to get more accurate results. We would also like to find partially periodic patterns and other types of patterns which may exist in the datasets.

References

1. R. Agrawal, T. Imielinski and A. Swami; Mining association rules between sets of items in large databases; Proceedings of the ACM SIGMOD '93, Washington, USA, May 1993.
2. J. M. Ale, and G. H. Rossi; An approach to discovering temporal association rules; Proceedings of the 2000 ACM symposium on Applied Computing, March 2000.
3. A. K. Mahanta, F. A. Mazarbhuiya and H. K. Baruah; Finding Locally and Periodically Frequent Sets and Periodic Association Rules, Proceeding of 1st Int'l Conf on Pattern Recognition and Machine Intelligence (PreMI'05),LNCS 3776, 576-582, 2005.
4. F. A. Mazarbhuiya; An efficient implementation of an algorithm for mining locally frequent patterns, International Journal of Innovative Research in Engineering and Management (IJIREM), ISSN: 2350-0557, Vol. 3 (1), India, pp. 55-58, January, 2016.
5. R. Agrawal and R. Srikant; Fast algorithms for mining association rules, Proceedings of the 20th International Conference on Very Large Databases (VLDB '94), Santiago, Chile, June 1994.
6. X. Chen and I. Petrounias; A framework for Temporal Data Mining; Proceedings of the 9th International Conference on Databases and Expert Systems Applications, DEXA '98, Vienna, Austria. Springer-Verlag, Berlin; Lecture Notes in Computer Science 1460, 796-805, 1998.
7. X. Chen and I. Petrounias; Language support for Temporal Data Mining; Proceedings of 2nd European Symposium on Principles of Data Mining and Knowledge Discovery, PKDD '98, Springer Verlag, Berlin, 282-290, 1998a.
8. X. Chen, I. Petrounias and H. Healthfield; Discovering temporal Association rules in temporal databases; Proceedings of IADT'98 (International Workshop on Issues and Applications of Database Technology, 312-319, 1998.
9. 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 Reasonong, 2001.
10. F. A. Mazarbhuiya, M. Shenify, A. Khan, and A. Farook; Finding Cyclic Frequent Itemsets, International Journal of Computer Science Issues, Vol. 9. Issue 6 No. 1, pp. 229-236, 2012.
11. B. Ozden, S. Ramaswamy and A. Silberschatz; Cyclic Association Rules, Proc. of the 14th Int'l Conference on Data Engineering, USA, 412-421, 1998.

Author(s) Profile



Fokrul Alom Mazarbhuiya received B.Sc. degree in Mathematics from Assam University, India and M.Sc. degree in Mathematics from Aligarh Muslim University, India. After this he obtained the Ph.D. degree in Computer Science from Gauhati University, India. He worked as an Assistant Professor in College of Computer Science, King Khalid University, Saudi Arabia from 2008 to 2011. Curently he is an Assistant Professor in College of Computer Science & IT, Albaha University, Saudi Arabia. His research interest includes Data Mining, Information security, Fuzzy Mathematics and Fuzzy logic.