National Conference On Emerging Trends in Computing Technologies (NCETCT-18) - 2018

RESEARCH ARTICLE OPEN ACCESS

Discovering Transitional Patterns and Their Significant Milestones in Transaction Databases

R.Sripriya

Assistant Professor, Dept.of Computer Science, Immaculate College for Women, Viriyur.

Abstract:

A transaction database usually consists of a set of time-stamped transactions. Mining frequent patterns in transaction databases has been studied extensively in data mining research. However, most of the existing frequent pattern mining algorithms (such as Apriori and FP-growth) do not consider the time stamps associated with the transactions. In this paper, we extend the existing frequent pattern mining framework to take into account the time stamp of each transaction and discover patterns whose frequency dramatically changes over time. We define a new type of patterns, called transitional patterns, to capture the dynamic behavior of frequent patterns in a transaction database. Transitional patterns include both positive and negative transitional patterns. Their frequencies increase/decrease dramatically at some time points of a transaction database. We introduce the concept of significant milestones for a transitional pattern, which are time points at which the frequency of the pattern changes most significantly. Moreover, we develop an algorithm to mine from a transaction database the set of transitional patterns along with their significant milestones. Our experimental studies on real-world databases illustrate that mining positive and negative transitional patterns is highly promising as a practical and useful approach for discovering novel and interesting knowledge from large databases.

Keywords:

I. INTRODUCTION

The problem of mining frequent itemsets is to find all the itemsets from a transaction database that satisfy a userspecified support threshold. It is one of the fundamental and essential operations in many data mining tasks, such as association rule mining, sequential pattern mining, structured pattern mining, correlation mining, and associative classification. Since it was first introduced by Agrawal et al. in 1993, the problem of frequent itemset mining has been studied extensively. As a result, a large number of algorithms have been developed in order to efficiently solve the problem, including the most well-known Apriori, FP-growth, and Eclat algorithms. In practice, the number of frequent patterns generated from a data set are often excessively large, and most of them are useless or simply redundant. Thus, there has been interest in discovering new types of patterns, including maximal frequent itemsets, closed frequent itemset, indirect associations, and emerging patterns. Mining for maximal or closed frequent itemsets greatly reduces the number of generated patterns by generating only the largest frequent itemsets with no frequent superset or no superset of higher frequency. Indirect associations are closely related to negative associations in that they both represent itemsets that do not have sufficiently high support. Indirect associations provide an effective way to detect interesting negative associations by discovering only “infrequent itempairs that are highly expected to be frequent” without using negative items or domain knowledge. Emerging patterns are defined as itemsets whose frequency increases significantly from one data set to another. They can capture emerging trends from one database to the other.

II.EXISTING SYSTEM:

- Apriori Algorithm for Transactional Databases without Timestamp
- The problem of frequent itemset mining has been studied extensively.
- As a result, a large number of algorithms have been developed in order to efficiently solve the problem.
- The number of frequent patterns generated from a data set are often excessively large, and most of them are useless or simply redundant.
- A common characteristic of the above learning methods is that they treat the transactions in a database equally and do not consider the time stamps associated with the transactions.

- Therefore, the dynamic behavior of the discovered frequent patterns cannot be revealed by these methods.

III.PROPOSED SYSTEM:

- Apriori Algorithm for Transactional Databases with Timestamp

- We extend the traditional frequent pattern mining framework to take into account the time stamp of each transaction, i.e., the time when the transaction occurs.
- We define a new type of patterns, called transitional patterns, to represent patterns whose frequency dramatically changes over time.
- Transitional patterns include both positive and negative transitional patterns.
- The frequency of a positive transitional pattern increases dramatically at some time point of a transaction database, while that of a negative transitional pattern decreases dramatically at some point of time.

IV.MODULES

1. Generating Transactional Database
2. Find Frequent Itemset (Apriori Algorithm)
3. Set Uniform Time points (1 - 100)
4. Find Positive and Negative Transitional Patterns
5. Generate Transitional Ratio

Module 1

In this module, first we have to create a transactioanl database for finding transitional patterns. The transactional databases are similarly very large databases. The items presented in the databases are the real time data.

Module 2

In second module we have to implement the Apriori algorithm for finding frequent itemset. These algorithm includes following steps:

- Generate Unique Itemset from transactional database
- Evaluate subsets for unique itemset.
- Calculate support value for one itemset, two and so on
- Check support value with support threshold. If support value ≥ support threshold then include it in frequent itemset.

Module 3

The next module is used to set uniform time points to the items presented in the transactional databases.

Module 4

The fourth module is used to implement the proposed method, which is finding positive and negative transitional patterens. We have to use the TB – mine algorithm for mining the set of positive and negative transitional patternsand their significant milestones with respect to a pattern support threshold and a transitional pattern threshold.

There are two major phases in this algorithm. During the first phase, all frequent itemsets along with their supports are initially derived using a standard frequent pattern generation algorithm, such as Apriori or FP-growth, with minimum support threshold. In the second phase, the algorithm finds all the transitional patterns and their significant milestones based on the set of frequent itemsets.

Module 5

We first introduced the concept of transitional patterns and an algorithm for mining transitional patterns and their significant milestones. In that algorithm, for each frequent itemset, we calculated two supports of the pattern and the transitional ratio (if the two supports satisfy the minimum support threshold) at each time point that corresponds to a time stamp in the transaction database, while in the new TP-mine algorithm, these values are calculated at each milestone that corresponds to the time point where the itemset occurs. In this module we generate the transitional ratio as follows, a transitional pattern was defined as a frequent pattern whose transitional ratio satisfies the transitional pattern threshold at atleast one of the time points.

V. CONCLUSIONS

A limitation of existing frequent itemset mining framework is that it does not consider the time stamps associated with the transactions in the database. As a result, dynamic behavior of frequent itemsets cannot be discovered. In this project, we introduced a novel type of patterns, positive and negative transitional patterns, to represent frequent patterns whose frequency of occurrences changes significantly at some points of time in a transaction database. We also defined the concepts of significant frequency-ascending milestones and significant frequency-descending milestones to capture the time points at which the frequency of patterns changes most significantly. To discover transitional patterns, we proposed the TP-mine algorithm to mine the set of positive and negative transitional patterns with respect to a pattern support threshold and a transitional pattern threshold. Our algorithms takes one database scan after mining frequent patterns to find the transitional patterns and their significant milestones. Our experimental results showed that the proposed algorithm is highly scalable.

VI. REFERENCES

[1] Chieh-Jen Cheng, Chao-Ching Wang, Wei-Chun Ku, Tien-Fu Chen , and Jinn-Shyan Wang, "Scalable High-Performance Virus Detection Processor Against a Large Pattern Set for Embedded Network Security" Commun. vol. 51, pp. 62–70,2011.

[2] O. Villa, D. P. Scarpazza, and F. Petrini, "Accelerating real-time string searching with multicore processors," Computer, vol. 41, pp. 42–50,2008.

[3] D. P. Scarpazza, O. Villa, and F. Petrini, "High-speed string searching against large dictionaries on the Cell/B.E. processor," in Proc. IEEE Int. Symp. Parallel Distrib. Process., 2008, pp. 1–8.

[4] D. P. Scarpazza, O. Villa, and F. Petrini, "Peak-performance DFA based string matching on the Cell processor," in Proc. IEEE Int. Symp. Parallel Distrib. Process., 2007, pp. 1–8.

[5] L. Tan and T. Sherwood, "A high throughput string matching architecture for intrusion detection and prevention,"in Proc. 32nd Annu. Int. Symp. Comput. Arch., 2005, pp. 112–122.

[6] S. Dharmapurikar, P. Krishnamurthy, and T. S. Sproull, "Deep packet inspection using parallel bloom filters," IEEE Micro, vol. 24, no. 1, pp.52–61, Jan. 2004.

[7] R.-T. Liu, N.-F. Huang, C.-N. Kao, and C.-H. Chen, "A fast string matching algorithm for network processor-based intrusion detection system," ACMTrans. Embed. Comput. Syst., vol. 3, pp. 614–633, 2004.

[8] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern matching using TCAM," in Proc. 12th IEEE Int. Conf. Netw. Protocols, 2004, pp. 174–178.intrusion detection system," ACMTrans. Embed. Comput. Syst., vol. 3, pp. 614–633, 2004.

[9] R. S. Boyer and J. S. Moore, "A fast string searching algorithm,"Commun. ACM, vol. 20, pp. 762–772, 1977.

[10] V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliographic search," Commun. ACM, vol. 18, pp. 333–340, 1975