International Journal of Computer Applications Technology and Research
Volume 4– Issue 5, 351 - 354, 2015, ISSN:- 2319–8656

Frequent Data Mining in Data Publishing for Privacy Preservation

Sheikh Nargis Nasir
Matoshri COERC,
Nashik, India

Swati A. Bhawsar
Matoshri COERC,
Nashik, India

Abstract: Weighted frequent pattern mining is suggested to find out more important frequent pattern by considering different weights of each item. Weighted Frequent Patterns are generated in weight ascending and frequency descending order by using prefix tree structure. These generated weighted frequent patterns are applied to maximal frequent item set mining algorithm. Maximal frequent pattern mining can reduces the number of frequent patterns and keep sufficient result information. In this paper, we proposed an efficient algorithm to mine maximal weighted frequent pattern mining over data streams. A new efficient data structure i.e. prefix tree and conditional tree structure is used to dynamically maintain the information of transactions. Here, three information mining strategies (i.e. Incremental, Interactive and Maximal) are presented. The detail of the algorithms is also discussed. Our study has submitted an application to the Electronic shop Market Basket Analysis. Experimental studies are performed to evaluate the good effectiveness of our algorithm..

Keywords: Data Mining, Incremental mining, Interactive mining, Maximal mining, Support;

1. INTRODUCTION

Nowadays, many commercial applications have their data presented in the form of continuously transmitted stream, namely data streams. In such environments, data is generated at some end nodes or remote sites and received by a local system (to be processed and stored) with continuous transmission. It is usually desirable for decision makers to find out valuable information hidden in the stream. Data-stream mining [1][2] is just a technique to continuously discover useful information or knowledge from a large amount of running data elements. Apart from traditional databases, evolving data set has some special properties: continuous, unbounded, coming with high speed and time varying data distribution. Therefore, discovering knowledge from data streams masquerades some limitations as follows. First, traditional multi-scan algorithms are no more allowed on infinite data as it can't be stored. Second, the algorithm must be as fast as possible because of high arrival rate of the data; otherwise, the accuracy of mining results will be decreased. Third, the data distribution within the data streams should be kept to avoid concept drifting problem. Fourth, it needs incremental processes to process the existing data as less as possible.

On the other hand, the main problem exists in this work is that the actual profits of items are not considered. In many applications, such as e-business, this factor is often one of the most important factors for the results. To triumph over this problem, frequent pattern mining [3] emerges as a new research issue for discovering the itemsets with high weights, i.e., high profits. To discover useful information from data streams, we need not only efficient one-pass algorithms but also effective data summary techniques.

The remainder of this paper is organized as follows. In section 2, we describe motivation. In section 3, we develop our proposed technique weighted frequent pattern mining over data stream. In section 4, our experimental results are presented and analyzed. Lastly, in section 5, conclusions are drawn.

2. MOTIVATION

In the very beginning some weighted frequent pattern mining algorithms MINWAL [4], WARM [5], WAR[6] have been developed based on the Apriori Algorithm [7]. There are two main problems exist in relevant studies: (1) The utilities (e.g., importance or profits or weights) of items are not considered. Actual weights of patterns cannot be reflected in frequent patterns. (2) Existing weighted frequent pattern mining methods produce too many patterns and this makes it difficult for the users to filter useful patterns among the huge set of patterns. In examination of this, in this paper we proposed a framework, to find maximal high utility patterns from data streams.

Motivated by these real world scenarios, in this paper, we propose a tree based technique to mine weighted frequent patterns over data streams. It can discover useful recent knowledge from a data stream by using a single scan. Our technique exploits a tree growth mining approach to avoid level-wise candidate generation and test problem. Besides retail market data, our technique can be well applied for the area of mining weighted patterns. By considering different importance values for different items, our algorithm can discover very important knowledge about weighted frequent items in real time using only one scan of data stream. Downward closure property is used to prune the infrequent patterns [7].

Main contributions of this paper are as follows: (1) This is the first approach on mining the compact form of high utility patterns from data streams; (2) the proposed framework is an effective single-pass framework which meets the requirements of data stream mining; (3) It also generates patterns which are not only high utility but also maximal. This provides compact and intuitive hidden information in the data streams. An itemset is called maximal if it is not a subset of any other patterns [8].

3. TECHNIQUE USED:

An evolving dataset may have infinite number of transactions. A weighted support of pattern P is calculated over by multiplying its support with its weight. Therefore, pattern P is weighted frequent pattern if its weighted support is greater than or equal to the minimum threshold. For example, if minimum threshold is 3.0,"ab" is a weighted frequent pattern. Its weighted support is 4*0.55=2.2, which is greater than the minimum threshold. Let, X= (X1, X2, X3,....,Xm) Where, X is

pattern of item set, X ϵ I and k ϵ [1,m]. The weight of pattern (WT), P[X1,X2,X3,...,Xk] is given by:

$$\text{Weight}(P) = \frac{\sum_{q=1}^{\text{length}(P)} \text{Weight}(x_q)}{\text{length}(P)}$$

Our proposed technique consist of some preliminary steps like Generation of header table, Construction of tree structure, Calculate weighted support, pattern pruning and evaluation of maximal frequent patterns.

Initially, the database performs transactions according to the user choice viz; get transaction or remove transaction. Transactions are read one by one from a transaction database and insert it into the tree according to any predefined order. The header table gets updated according to weighted ascending order and frequency descending order simultaneously. Each entry in a header table explicitly maintains item-id, frequency and weight information for each item. Then, WFP mining takes into an account that generated weighted frequent item sets. The generated weighted frequent patterns are appended for the maximal weighted frequent itemset mining. Then, Vertical bitmap is maintained to keep the record of candidate patterns. It performs AND-ing operation on the items of transactions. lastly, the final output is generated from weighted frequent patterns i.e. Maximal weighted frequent pattens mining. The proposed work is divided into four major modules:

1. **Database Transaction:**

The weight of the items has to be taken into consideration so that the algorithm can be more effective in real world applications. The weight of the pattern is the average of the weight of the itemsets that constitute the pattern if the weighted support of the pattern is greater than or equal to the minimum threshold. Header table is generated to handle the item weights and frequency.

2. **Calculate Weighted Support:**

The value achieved when multiplying the support of a pattern with the weight of the pattern is the weighted support of that pattern. That is, given pattern P, the weighted support is defined as WSupport(P) = Weight(P)*Support(P). A pattern is called a weighted frequent pattern if the weighted support of the pattern is no less than the minimum support threshold.

3. **Pattern Pruning:**

If the value of Weighted Support is greater than or equal to the threshold, then it is considered as frequent pattern else pattern is pruned.

4. **Maximal Mining:**

From the resultant weighted frequent patterns, maximal weighted frequent patterns are extracted.

4. ALGORITHMIC STRATEGY:

When new transactions are inserted or deleted, the incremental algorithm is processed to update the discovered most frequent itemsets. These transactions can be partitioned into four parts according to whether they are high transaction-weighted utilization itemsets or not in the original database. Each part is then processed in its own procedure. A simple way of finding possible frequent itemsets is to mine frequent patterns from every possible transaction, and then calculate weighted support of the occurrences of these patterns. The details of the proposed incremental mining algorithm are described below.

4.1.1 Algorithmic Strategy to Implement Incremental Data Mining:

The important problem is extracting frequent item sets from a large uncertain database. Frequent patterns are interpreted by calculating the weighted support for each pattern under the weight and frequency of the item. This issue is technically challenging for an uncertain database which contains an exponential number of possible patterns. By observing that the mining process can be modeled as a Poisson binomial distribution, we develop an approximate algorithm, which can efficiently and accurately discover frequent item sets in a large uncertain database. We also study the important issue of maintaining the mining result for a database that is evolving (e.g., by inserting a transaction). Specifically, we implement incremental mining algorithms, which enable Probabilistic Frequent Item set (PFI) results to be refreshed. This reduces the need of re executing the whole mining algorithm on the new database, which is often more expensive and unnecessary.

Downward Closure Property:

The downward closure property [1] is used to prune the infrequent patterns. This property says that if a pattern is infrequent, then all of its super patterns must be infrequent. We can maintain the downward closure property by transaction weighted utilization. In this method a data structure, called prefix Tree, is introduced to maintain frequent item sets in evolving databases. Another structure, called conditional, arranges tree nodes in an order that is affected by changes in weighted support for candidate pattern. The data structure is used to support mining on a changing database. To our best knowledge, maintaining frequent item sets in evolving uncertain databases has not been examined before. Here, Static Algorithms do not handle database changes. Hence, any change in the database necessitates a complete execution of these algorithms.

Following are the input and output requirements for implementing this incremental data mining algorithm.

Input:

1. Database,
2. Weight Table,
3. Updated database,
4. Minimum threshold

Output:

1. Weighted Frequent Patterns

4.1.2 Algorithmic Strategy to Implement Interactive Data Mining:

The data structures of the existing frequent pattern mining algorithms do not have the "build once mine many" property. As a consequence, they cannot use their previous data structures and mining results for the new mining threshold. This property means that by building the data structure only once, several mining operations can be done for interactive mining. For example, if the algorithms presented in the previous works want to calculate which patterns cover 40% of the total profit, then their internal data structures are designed in such a way that they can only calculate the asked amount. If the amount is changed from 40% to 30% of the total profit, then they have to do the whole calculation from the very beginning. They cannot take any advantage from their previous design. They have shown that incremental prefix-tree structures are quite possible and efficient using currently available memory in the gigabyte range. In our real world, however, the users need to repeatedly change the minimum threshold for useful information extraction according to their application requirements. Therefore, the "build once mine many" property is essentially needed to solve these interactive mining problems.

Motivated by these real world scenarios, in this project, we presented a tree structure, called frequent pattern tree (or high utility stream tree) and an algorithm, called high utility pattern mining over stream data, for incremental and interactive weighted frequent pattern mining over data streams. By exploiting a pattern growth approach, this algorithm can successfully mine all the resultant patterns. Therefore, it can avoid the level-wise candidate generation-and-test problem completely and reduces a large number of candidate patterns. As a consequence, it significantly reduces the execution time and memory usage for stream data processing.

Input:

1. Database,
2. Weight Table,
3. Updated database,
4. Minimum threshold

Output:

1. Weighted Frequent Patterns

4.1.3 Algorithmic Strategy to Implement Maximal Weighted Frequent Pattern Mining:

In this Maximal Miner algorithm, a descending support or frequency count order method is used. A divide-and-conquer traversal paradigm is used to mine weighted FP-tree for mining closed weighted patterns in bottom-up manner. The Maximal frequent itemset tree is used to store so far found (global) maximal weighted frequent patterns. After mining the traversal transaction, the set of real maximal weighted frequent Patterns is generated. This is because weighted Maximal Mining carries out the maximal frequent pattern mining with weight constraints. This proposed approach can reduces search space effectively. As compared to other methods, FPmax method only does the maximal frequent pattern mining without weight constraints, its search space is larger than that of our algorithm. Following diagram shows the location maximal frequent itemsets in frequent itemsets and closed patterns.

In this algorithm, we use divide-and-conquer paradigm with a bottom-up pattern-growth method and incorporates the closure property with weight constrain to reduce effectively search space. This also includes anti-monotone property. The reason for that is weighted Maximal Mining has a weight constraint to reduce the search space than FPmax [27] which has not weight constraint. To reduce the calculation time, they have used bit vectors and TID-lists for each distinct item. But these lists become very large and inefficient when the numbers of distinct items and/or transactions become large.

Anti-monotone Property:

The main focus in weighted frequent pattern mining is on satisfying the anti-monotone property[27] since this property is generally broken when different weights are applied to different items. Even if a pattern is weighted as infrequent, its super patterns can be weighted as frequent since super patterns of a low weight pattern can receive a high weight after other items with higher weight are added.

Input:

1. Weight Table,
2. Frequent Patterns,
3. Minimum Threshold.

Output:

1. Maximal Weighted Frequent Patterns.

4.1.4 Mining Process:

Here, we develop scalable algorithms for finding frequent item sets (i.e., sets of attribute values that appear together frequently in tuples) for uncertain databases. Our algorithms can be applied to tuple or transactions uncertainty models. Here, every tuple or transaction is associated with a probability to indicate whether it exists. The frequent item sets discovered from uncertain data are naturally probabilistic, in order to reflect the confidence placed on the mining results.

5. EXPERIMENTAL RESULTS:

5.1 Experimental Environment & Datasets:

Experimentation is carried out on Customer purchase behaviors-supermarket basket databases. This synthetic dataset contain statistical information for predicting what a customer will buy in the future. The weight value associated with each item represents the chance that a customer may buy that item in the near future. These probability values may be obtained by analyzing the users' browsing histories. For instance, if customer visited the marketplace 10 times in the previous week, out of which video products were clicked five times, the marketplace may conclude that customer has a 50 percent chance of buying videos. Conceptually, a database is viewed as a set of deterministic instances (called possible patterns), each of which contains a set of items. To implement and test this system, we have used a market basket analysis- synthetic dataset in which various transactions are performed on the items of the market.

5.1.1 Observations of the system on sparse dataset:

Impact of Parameter Minimum Threshold (delta)

Following graph in the figure 5.1 is drawn by taking different minimum threshold values to the 15 number of transactions.

Analysis:

1. Time required by itemsets in frequency descending order is less than the time consumed by itemsets in weight ascending order.
2. As the threshold value increases, the run time required is decreases.

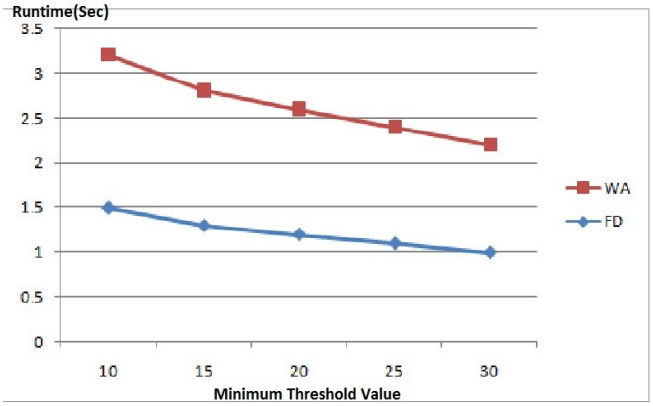

Figure 5.1 Impact of the No. of Transactions

Observations:

To observe the behavior of our proposed system under increased number of transactions, we applied a constant threshold value i.e. Threshold value = 40. Following graph is drawn from different number of transactions to the same input value, runtime is calculated. Figure 5.2: Impact of No. of transactions on efficiency

Analysis:

1. For any number of transactions, time required by the structure in weight ascending order is greater than the structure in frequency descending order.
2. As the number of transactions increases, the time required to execute transactions also increases.

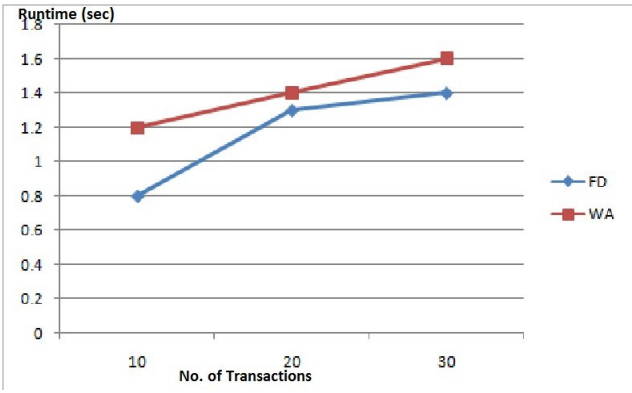

Figure 5.2 Impact of Modified Transactions

5.1.2 Comparison between Sorting Frequency in Descending Order and Weight in Ascending Order:

Analysis from Graph 1 and Graph 2:

1. IWFPTFD guarantees that non candidate item can't be passed in to the set of candidate patterns.
2. IWFPTFD creates tree structure from the candidate items generated from IWFPTWA. hence, speed up the tree creation process.
3. IWFPTED reduces memory space required to store items.
4. This speed up the overall time required to mine patterns.

Analysis:

1. When the newly discovered transactions are added to the existing dataset, existing tree structure in modified for the newly evolved transactions.
2. This reduces the processing overhead of the transactions.

Worst Case Scenario: When all the transactions are modified.

Best Case Scenario: When no transaction is modified.

6. CONCLUSION

The algorithm exploits two tree structures which employs weighted frequent pattern mining over data streams. The major objective of discovering recent weighted frequent patterns from uncertain database is fulfilled. By making use of efficient tree structure, our projected technique could capture newest data from a data stream. Since, it requires a single-pass of data stream for tree construction and mining operations. It is reasonably appropriate to apply maximal weighted frequent pattern mining algorithm to the operational database. The mining paradigm also prunes the unimportant patterns and reduces the size of the search space. We executed this work on synthetic dataset of Market Basket Analysis. The results show that our paradigm reduces the size required to search a frequently used patterns. Also, this could speed up the process to mine weighted frequent patterns.

7. REFERENCES

[1] Bifet, A., Holmes, G., Pfahringer, B., & Gavaldà, R. (2011). Mining frequent closed graphs on evolving data streams. In Proceedings of the 17th ACM SIGKDD conference on Knowledge Discovery and Data Mining (KDD 2011), San Diego, CA, USA (pp. 591–599).

[2] Cheng, J., Ke, Y., & Ng, W. (2008). A survey on algorithms for mining frequent itemsets over data streams. Knowledge and Information Systems, 16(1), 1–27.

[3] Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In Proceedings of the 20th international conference on very large data bases (pp. 487– 499).

[4] C. H. Cai, A.W. fu, C. H. Cheng and W. W. Kwong,"mining association rules with weighted items".Proc of int. database engineering and application symposiums, IDEAS 98,pp 68-7,Cardiff, Wales, UK, 1998.

[5] F. Tao,"Weighted association rules using weighted support and significant framework", Proc. Ninth ACM SIGKDD Int. conference on knowledge discovery and data mining,pp 661-666, 2003.

[6] W. Wang and J. Yang and P.S. Yu,"WAR: Weighted association rules for item intensities", Knowledge Information and Systems , vol 6, pp 203-229, 2004.

[7] Agrawal, R., Imielin´ ski, T., Swami, A. (1993). Mining association rules between sets of items in large databases. In Proceedings of the 12th ACM SIGMOD international conference on management of data, May (pp. 207–216).

[8] Gouda, K., & Zaki, M. J. (2001). Efficiently mining maximal frequent itemsets. In Proceedings of the IEEE international conference on data mining (ICDM), San Jose (pp. 163–170).