Survey on Weighted Frequent Pattern Mining

Ambily Mohan¹, Visakh R²
¹Department of Computer Science & Engineering, Rajagiri School of Engineering and Technology, India
²Department of Computer Science & Engineering, Rajagiri School of Engineering and Technology, India

ABSTRACT: Data mining is the collection of techniques for the resourceful, automatic discovery of previously unknown, suitable, novel, helpful and understandable patterns in large databases. Frequent pattern mining has emerged as a vital task in data mining. Frequent patterns are those that occur frequently in a data set. In traditional frequent pattern mining, patterns and items within the patterns are treated uniformly with the assumption that all items have some significance. However, real items have different importance as some of the items are more beneficial in terms of profit. This thought resulted in weighted frequent pattern mining in which benefits are taken into account. In this paper, a comprehensive survey of the techniques used in mining weighted frequent patterns is being discussed.

Keywords - Data mining, frequent patterns, weights, weighted frequent patterns, weighted support.

1. INTRODUCTION

Data mining [1] is known for discovering previously unknown, valid, novel, useful and understandable patterns in large databases. Due to the availability of huge amount of data and the need to transform such data into useful information and knowledge, data mining has grown to be the most widely used technique in the society as a whole. Thus data mining can be used for applications ranging from market analysis, fraud detection and customer retention, to production control and science exploration.

Frequent pattern mining has become an important task in data mining. Frequent patterns are those patterns that occur frequently in a data set. For example, a set of items, such as milk and bread usually appears frequently together in a transactional database. So it is treated as a frequent itemset. Associations, correlations, and many other interesting relationships among data can be mined using frequent pattern mining.

Market basket analysis is one such application of frequent pattern mining. In this process, customer’s buying habits in terms of purchased items are analyzed to find out the items that are frequently bought together. Based on this analysis, the retailers can develop several marketing strategies that will help them in improving their sales as they are aware of the frequently purchased items. For instance, if customers are buying milk, how likely they will also buy bread on the same trip to the market? Extracting such information can lead to increased sales.

Frequent pattern mining algorithms like Apriori [2], FP-growth [3] are simply based on the count of the occurrences of itemsets in a transactional dataset. The support for an itemset is the number of transactions that contain the itemset. A pattern is said to be frequent if its support is above a minimum support threshold. Above all it treats all the items uniformly regardless of their significance in real application. Real data have different importance as some of the items are more beneficial in terms of the profit they yield per unit sale. For example, sale of egg may incur a profit of 20 cents, while a bottle of milk might incur a profit of 40 cents. Even if latter holds a low support value, it will be of more benefit than the other in terms of the profit. Traditional frequent pattern mining algorithms ignored this difference. With this thought, the idea of weighted frequent pattern mining came into existence.

In weighted frequent pattern mining, a weight factor is assigned to each of the items present in the transactions based on their benefits. The concept of weighted support is used instead of support used in traditional pattern mining which was simply the count of occurrence of itemsets in each transaction. Weighted support calculated by making use of the weights of items resulted in the selection of important patterns. An itemset is significant if its weighted support is above a pre-defined minimum weighted support.

The remainder of the paper is organized as follows. In section 2, preliminaries of weighted frequent pattern mining are discussed, followed by the brief idea of weighted support concept in section 3. Section 4 discusses the various techniques that are used for mining weighted frequent itemsets from transactional databases. Finally, conclusions are presented in section 5.

2. PRELIMINARIES

Let \( I = i_1, i_2, \ldots, i_n \) be a set of items. Let \( D \) be a transactional database which denotes a set of database transactions where each transaction \( T \) is a set of items such that \( T \subseteq I \). Each transaction is associated with an identifier, called TID. Let \( X \) be
an itemset such that \( X \subseteq \mathcal{I} \). A transaction \( T \) is said to contain \( X \) if \( X \subseteq T \). A set of items is referred to as an itemset. An itemset that contains \( k \) items in it is referred to as a \( k \)-itemset. The set \{computer, antivirus _software\} is a 2-itemset. Let \( W \) be a set of non negative real numbers. A pair \((x, w)\) is called a weighted item and \( w \) is the weight associated with \( x \).

Item weight [4] is a weight value bound to an item based on its significance. It is denoted as \( w(i) \). Weighting attributes are used for the calculation of weights. These are variables that range from item’s price in a supermarket to visitor page dwelling time in a web log mining domain. Therefore, item weight is a function of its weighting attribute denoted as \( w(i) = f(a) \).

3. **WEIGHTED SUPPORT**

Weighted support [5] computation involves the multiplication of two measures; supports and weights. If supports and weights are considered separately, then mining would result in itemsets that should have both the sufficient weight and as well as the support count. However this could result in loss of interesting knowledge. For example, a product which is under promotion and has its sale close to zero will be pruned as it does not have sufficient support.

Multiplication however provides a balance between the weight and the support. Weights can be regarded as the adjusting factor of the support. This is done with the help of multiplication. If the weight is very low, it will lower the weighted support, and it will keep the original support value, if the weight value is 1.

4. **WEIGHTED FREQUENT PATTERN MINING TECHNIQUES**

Zi-guo Huai and Ming-he Huang in [6] proposed an effective algorithm for mining weighted frequent itemsets. It was based on the hash table data structure. This technique could deal with the problems that arise when weight value of an item is changed and also when the database is changed. It requires only a single scan of the database. Weighted support calculation for the itemset \( X \) uses (1).

\[
: W(\text{Sup}(X)) = W(X) \times \text{Sup}(X) \quad (1)
\]

\( W(X) \) denotes the weight of itemset \( X \). \( \text{Sup}(X) \) is the support counts of itemset \( X \).

: \( W(X) = \text{Max } w(x) \) \text{ where, } \( x \in X \) (2)

The algorithm involves the construction of weighted hash table. Scan the first transaction and obtain all its non empty subsets. Put the subsets into the corresponding hash table according to the number of items in the subsets. Using the hash function, find its hash address. More than one sub set will be hashed to same address. It’s stored as a linked list of nodes. Each node contains three domains: \( \text{Item, Item_Count, Item_Weight and Next_Item} \). If this linked list already has the subset contained it, then increment \( \text{Item_Count} \) by 1, else add the subset to the tail of this linked list. Calculate \( W(\text{Sup}(X)) \) and store it into \( \text{Item_Weight} \). This is repeated for the entire transactions. Here mining operation is a parallel process since each hash table is independent and this improves the efficiency. When the weight value associated with the item is changed, weighted support of itemset in \( \text{Hash}[i] \) needs to be recalculated. When database is increased (db), each subset of every transaction of db will be stored in the corresponding hash table just done as before.

Approximate weighted frequent pattern mining was proposed by Unil Yun and Keun Ho Ryu in [7]. They proposed this concept thinking that real applications would contain huge amount of data, and inherent noise in the data would result in its inaccurate processing. Noisy data due to uncertainty of processing data and measurement errors can have negative effects on mining results. Result sets will be affected greatly even if there is only a small change in item’s weight or item’s support in noisy environment. As a result, an approximate factor was proposed that relaxed the requirement of weighted supports being exactly matched with the minimum threshold. This led to the discovery of approximate weighted frequent patterns. WAF (approximate weighted frequent pattern mining) algorithm is used to extract robust important patterns that are not affected by noise.

Mining process is done with pattern growth method [3, 8]. Weighted support of a pattern \( X \) is calculated by multiplying patterns support with the weight of the pattern. Weight of the pattern is the average value of weight of items present in the pattern. In noisy environment, weighted support of a pattern may be increased or decreased. If weighted support is increased by noisy data, the approximate weighted support is set to \( w_i(X) + \varepsilon \) whereas; if the weighted support is decreased by a missing data \( w_i(X) - \varepsilon \) is computed as approximate weighted support.

With fault tolerant factor, weighted frequent pattern may become weighted infrequent if the varied weighted support \( \text{WS}(P) - \varepsilon \) is less than
WFIM mines frequent itemsets from the FP-tree in a bottom up manner. Form conditional databases for all remaining items and compute the local weighted frequent items using the pruning conditions. WFIM is scalable and efficient algorithm for mining weighted frequent patterns.

In [11], Li Tong Yan and Chen Chao proposed an algorithm for mining weighted frequent itemsets with a maximum length. Let MLWF denote the set of all maximum length weighted frequent itemsets.

If W-support(X) ≥ wminsup, the pattern X is a weighted frequent pattern. Also the count of X must satisfy (4).

\[ w \cdot \text{min\sup} \cdot n \cdot N \leq \text{count}(X) \geq \sum_{i,j \in X, j=1}^{n} w_{ij} \]  

Where N stands for the number of transactions in D and n is the size of the pattern. The maximum possible weight for any k-itemset containing Y is given by (5).

\[ \frac{1}{W_{mp}(Y,k)} = \left( \sum_{j \in A \cup P} w_{ij} \right) + \sum_{l=1}^{k-p} r_{l} \]  

Where Y is a p-itemset and X is a superset with k-itemset. If a superset of p-itemset Y is weighted frequent, its minimum support and the minimum support count is given by (6) and (7) respectively.

\[ R(Y,k) = \frac{w_{\text{minsup}}}{W_{mp}(Y,k)} \]  

\[ B(Y,k) = \left\lceil R(Y,k) \times N \right\rceil \]  

The algorithm is as follows. First find all maximal length weighted frequent itemset in the database. In the initial WFP-tree, an itemset with maximum length is a complete path from the root. If expected weight (all the items in the single path)*sup (largest itemset I in this path) is greater than or
equal to \( win\text{msup} \), insert \( I \) into maximum length weighted frequent itemset list. Else it’s a maximum length weighted infrequent itemset. For MLWII with size \( n \), divide it into the itemset with size \((n-1)\) until MLWFI is found. For MLWFI, find all weighted frequent subsets. For the remaining weighted infrequent itemsets in MLWII, mine it by generating conditional pattern trees similar to FP growth algorithm.

Preetham Kumar and Ananthnarayana, V. S in [12] proposed a parallel algorithm to discover all frequent itemsets based on the quantity of items bought in a transaction. Items that appear in few transactions may be purchased in large quantity leading to very high profit. This motivated them to propose an algorithm that mines all frequent weighted itemsets based on the quantity purchased in just one scan of the database. This algorithm is divided into two phases. First phase generates a data structure called Weighted Tree in just one scan of the database. In the second phase an ordered weighted tree is formed by reducing the weighted tree to contain only frequent items and all branches of the tree are sorted according to their increasing order of frequency. This is replicated among nodes and mined in parallel.

Weighted tree has two different nodes. The first type of node is labeled with attribute name and it contains two pointers one pointing to the nodes containing transaction ids and weights and another is a child pointer pointing to the next attribute. The second type has 2 parts. First part represents the transaction id and second part represents the weight. Removal of infrequent attributes is based on the pruning condition. If sum (weights of all nodes) is less than weighted minimum support, then remove that branch from the tree. Frequent items thus obtained are arranged in increasing order of their weighted minimum support and the tree thus obtained is the Ordered Weighted Tree.

The ordered weighted tree is then replicated among all parallel nodes. Each processor receives one item and each node is responsible for generating all frequent patterns related to the items associated with that item. From the ordered weighted tree, each node reads the sub transaction for each frequent item associated with it and then builds independent and relatively small trees called the item tree. After mining the item trees, they will be discarded. Finally all the frequent patterns generated at each node are gathered into master node to produce full set of frequent patterns. This algorithm is space and time efficient and requires only one scan of the database.

Bay Vo, F Coenen and Bac Le in [13] came up with an algorithm for fast mining of frequent weighted itemsets from weighted transaction databases. This algorithm was based on WIT-trees (Weighted Itemset-Tidset tree) data structure. In this paper, calculation of transaction weight and the weighted support of itemset use (8) and (9) respectively.

\[
\text{tw}(t_k) = \sum_{j \in t_k} \frac{w_j}{|t_k|} \quad (8)
\]

\[
\text{ws}(X) = \frac{\sum_{t_k \in t(X)} \text{tw}(t_k)}{\sum_{t_k \in T} \text{tw}(t_k)} \quad (9)
\]

The WIT-tree data is comprised of itemset TID lists that help in the fast computation of weighted support values. Each node in the WIT-tree has 3 fields associated with it:

- \( X \): an itemset.
- \( t(X) \): the set of transactions containing \( X \).
- \( ws \): the weighted support of \( X \).

All 1-itemset nodes belong to the root node of WIT-tree. They also belong to the same equivalence class with prefix \{\}. Each node using its item as the prefix in level 1 will become a new equivalence class and it will join with all other nodes following it to create a new equivalence class. The process is repeated to create new equivalence classes in higher levels of the WIT-tree. It requires only single scan of the database.

Diffset strategy [14] is used for the efficient computation of Weighted Support of the itemsets. Diffset computes the difference set between two Tidsets that belongs to the same equivalence class. It results in fast computation of the weighted support values and also reduces the storage required for the Tidsets.

5. Conclusion

Various techniques involved in the discovery of weighted frequent patterns have been surveyed. Weighted frequent pattern mining is used for extracting weighted frequent itemsets as real items have different importance. The merits and demerits of various techniques have been explained. Some may require only one scan of the database, some methods enable parallel processing, and different methodologies requires various data structures like hash table data structure, wit-trees data structures for efficient mining.
REFERENCES


