Mining Weighted Frequent Itemsets Algorithm

Ritesh Yadav

June 2020

1 Objective

Let I = {i_1,i_2,....,i_m } be a finite set of m different items. Let D = {T_1,T_2,....,T_n}, where each T_q represents a single transaction and is subset of I, with a unique identifier *TID*. A subset of I which consists of k distinct items, is called k-itemsets or itemset of length k. Assume that D is divided into k batches; each item in each batch is assigned a distinct weight, which is a nonnegative real number.
Given an uncertain database D, a user-specieed weight tables, and a user-specified minimum expected weighted-support threshold ϵ. The problem of mining weighted itemsets in an uncertain database D is to discover the set of expected weighted frequent k-itemsets while considering weights.

2 Terminology

- Adaptive weighted support of an itemset X, denoted as AWsupp(X), is denoted by:

$$\sum_{j=1}^{k} W(X,j) \times F(X,j) \tag{1}$$

  where W(X, j) is the weight of X in the j^{th} which is calculated by the average weight of the items in the batch belonging to X, F(X, j) is the support(or frequency) of X in the j^{th} batch.

- An itemset X is called adaptive weighted frequent itemset if the adaptive weighted support of X is greater or equal to the minimum threshold AWminsupp, that is:

$$AWsupp(X) \geq AWminsupp \tag{2}$$

- Given a transaction database DT consisting of K batches and an itemset X. Let MAXW(j)be the highest weight value of the items in the j^{th} batch, j = 1,...,K. Then the measure

$$MAXAWsupp(X) = \sum_{j=1}^{K} MAXW(j) \times F(X,j) \tag{3}$$

- Given a transaction database DT consisting of K batches and an itemset X. For a given threshold AWminsupp, Xis called a maximum adaptive weighted frequent itemset if

$$MAXAWsupp(X) \geq AWminsupp \tag{4}$$

- If X is a maximum adaptive weighted frequent itemset then all of its subsets are also maximum adaptive weighted frequent itemsets.
- Given a transaction database D and an itemset X. If X is an adaptive weighted frequent itemset then X is also a maximum adaptive weighted frequent itemset.

3 Algorithm

In order to mine adaptive weighted frequent itemsets,an algorithm AWFIMiner is used consisting of two steps, as following:

1. Find all maximum adaptive weighted frequent itemsets.
2. From the set of all maximum adaptive weighted frequent itemsets, by applying (1), determine the set of all adaptive weighted frequent itemsets.

3.1 Construction of AWFI-tree

- The structure of AWFI-tree is combined of two parts:
  - AWFI tree
  - Header
- AWFI-tree consists of one root node referred to as **null**(signs as {}), a set of item-prefix sub-trees as children of the root.
- Header table is maintained to keep items in lexicographical order and the related information of items. In each entry of a header table, the first value is the item identifier, after that the weight and frequency information of an item in batch by batch fashion, and a pointer pointing to the first node in the AWFI-tree carrying the item.
- Transactions in each batch of the database are inserted one by one into the tree in lexicographical order of items. Except the root, each node of the AWFI-tree contains item-identifier and its frequency information. To facilitate the tree traversals adjacent links are also maintained in AWFI-tree like FP-tree.
- **Tail Node :** Let T = $\{i_1,i_2,....,i_k\}$ be a transaction in a database with items sorted accoding to lexicographic order. If T is inserted into a SWFP-tree in this order, then the node of the tree that represents item i_k is defined as a tail-node for T.
- By using the notion of tail-node and ordinary node, we organize nodes in the AWFI-tree as follows: At each ordinary node, we only keep the total frequency of the node in the path from the root, but at each tail-node, we maintain a list of batch-by-batch frequency information.
- Whenever a new tail-node is created in the tree by inserting a transaction from the j^{th} batch of the database consisting of K batches, a list consisting of K frequency values in K batches will be created with value 1 at the j^{th}position, value 0 at all remaining positions.
- We can consider the AWFI-tree constructed above as an extended FP-tree. It inherits the compress advantage of FP-tree.

3.2 AWFIMiner Algorithm

AWFIMiner Algorithm mines adaptive weighted frequent itemsets from the AWFI-tree and gives set of all adaptive weighted frequent itemsets. Input to AWFIMiner would be a transaction database containing K batches, the weights of the items in each batch, the minimum support count thresold AWminsupp. Steps for applying algorithm is as follows :

1. Create AWFI-tree.
2. L = ϕ
3. Identify the set C_1 of candidate 1-itemsets, that are itemsets whose MAXAWsupp calculated by the formula(3) is no less than AWminsupp.
4. L = L $\cup$C$_1$
5. Prune the AWFI-tree: erase every node in the AWFI-tree that does not represent for the candidate items in C_1, (these nodes can not associate with other nodes to generate maximum adaptive weighted frequent item-sets).
6. For each item in the header table, in bottom-up order:
   (a) Construct its conditional tree.
   (b) Prune the conditional tree: erase every node which does not represent for the candidate items.
   (c) Mine pruned conditional tree to get maximum adaptive weighted frequent itemsets by pattern growth. Add the maximum adaptive weighted frequent itemsets into L.
   (d) From L, determine the adaptive weighted frequent itemsets, that are itemsets having AWsupp value no less than AWminsupp