CSE 4739
Data Mining
Assignment - 02
Frequent Itemset Mining and Association Rules
Task - 05

Mohammad Sabik Irbaz
ID : 160041004
Section 1

March 18, 2020

Implementation Process :

Step - 1 :

Finding out minimum support
Minimum Support = (Size of Data-set) x (Relative Support)

Step - 2 :

Finding out frequency of distinct one element in data-set and omitting the ones having lower frequency than Minimum Support.
Used mapping implementations to store and check. Here, omitting the ones having lower frequency is the required **optimization** in the question which reduces transaction scanning in the following steps.

Step - 3 :

Forming distinct 2-element sets, finding the frequency of each set and omitting the ones having lower frequency than Minimum Support. Then considering the rest as candidate sets.
Used 2D-mapping implementation for storing and checking. This mapped storage is used in further calculations for frequency checking. The ones after omitting are counted as candidate sets which will be used for further Frequent Itemset Mining and choosing final representative sets.

Step - 4 :

Forming distinct 3 to maximum possible element sets sequentially considering only last one is variant, finding the frequency of each set and omitting the ones having lower frequency than Minimum Support and the ones having sub-sequence of itemset omitted earlier. Then considering the rest as candidate sets.
This is the Final Step before validation. For each types of element set we do two types of **optimization** to reduce transaction scanning in further iterations. These are :

- After finding out all possible itemsets of each type, we discard the ones having a sub-sequence that's equivalent to the ones omitted in earlier iterations. Because, they won't ever be able to have minimum support either-way.
- From the rest of them, we omit the ones having lower support then minimum support.

Step - 5 :

Forming representative sets according to support count of the candidate sets.
We start to iterate from the itemsets with maximum size to the itemsets with size 2. Size 1 is not a valid sequence that's why not considered. While iterating, if a smaller set which is a subsequence of bigger set already considered as representative, have same or lower support count it is not considered as a representative set, or else it is considered as a representative set.

Findings :

We used two types of optimization. They worked because :

- After finding out all possible itemsets of each type, we discard the ones having a sub-sequence that's equivalent to the ones omitted in earlier iterations. It works because of intersection rules. Chances of A, B and C happening altogether is less than or equal to A and B happening together.
- From the rest of them, we omit the ones having lower support then minimum support. It also works because of intersection rules aforementioned.

By implementing these optimization, we have to do at least 20-30% less iterations.