LAB

Manual

PART

A

(PART A : TO BE REFFERED BY

STUDENTS)

**Experiment**

**No.10**

**A.1**

**Aim:**

To study and implement Apriori Algorithm for Association Rule Mining.

**A.2**

**Prerequisite:**

Refer the Lab manual for the steps and any programming language.

**A.3**

**Outcome:**

**After successful completion of this experiment students will be able to**

1. To understand the concept of Data Mining by implementing some data mining algorithm.
2. To understand the Association Rules in Mining.
3. To understand how to create Association Rules by Apriori Algorithm.

**A.4 Theory**

**Association Rules**

Association rules are used to show the relationships between data items.

**Association Rule Definitions**



***Set of items:*** I={I1,I2,…,Im}

***Transactions:*** D={t1,t2, …, tn}, tjI



***Itemset:*** {Ii1,Ii2, …, Iik}I

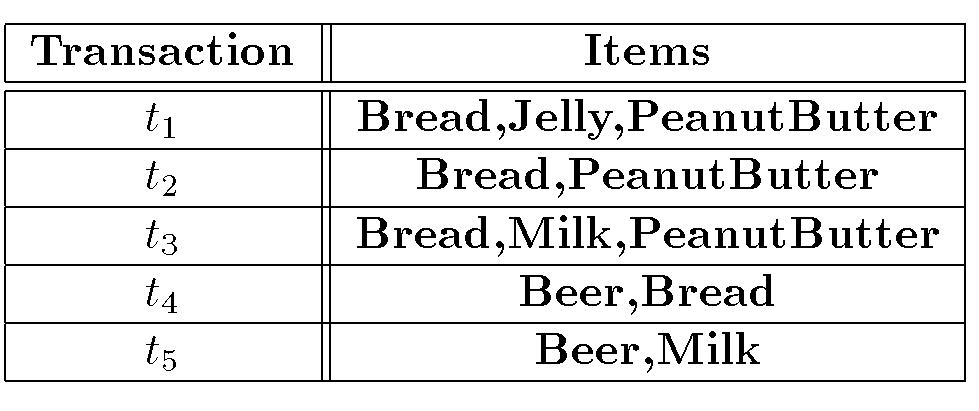


***Support of an itemset:*** Percentage of transactions which contain that itemset.



***Large (Frequent) itemset:*** Itemset whose number of occurrences is above a threshold.

**Association Rules Example**



I = { Beer, Bread, Jelly, Milk, PeanutButter}

Support of {Bread,PeanutButter} is 60%

**Association Rule Definitions**

***Association Rule (AR):*** implication XY where X,YI and X∩Y = Null;



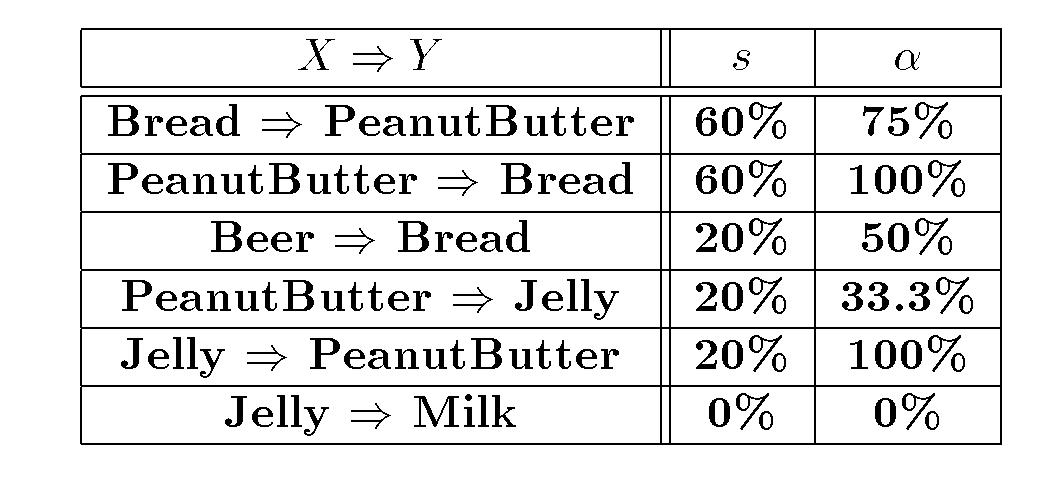
***Support of AR (s) X******Y:*** Percentage of transactions that contain X U Y



***Confidence of AR (a) X******Y:*** Ratio of number of transactions that contain X U Y to thenumber that contain X



**Association Rules Example**



**Apriori Algorithm**



***Large Item-set Property:*** Any subset of a large item-set is large.



**Contrapositive:** If an item-set is not large, none of its supersets are large.

**A.5 Procedure/Algorithm:**

**A.5.1 TASK 1:**

Join Step: Ck is generated by joining Lk-1with itself

•Prune Step: Any (k-1)-itemset that is not frequent can not be a subset of a frequent k-itemset

**Pseudo-code:**

*Ck*: Candidate itemset of size k

*Lk*: frequent itemset of size k

*L1*= {frequent items};

**for**(*k*= 1;*Lk*!=∅;*k*++) **do begin**

*Ck+1*= candidates generated from *Lk*;

**for each** transaction*t*in database do

increment the count of all candidates in *Ck+1* that are contained in *t* *Lk+1*= candidates in *Ck+1*with min\_support

**end**

**return** ∪*kLk*;

**Apriori Example**

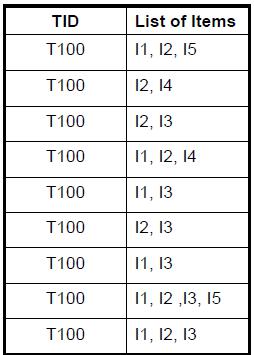
•Consider a database, D, consisting of 9 transactions.

•Suppose min. support count required is 2 (i.e. min\_sup = 2/9 = 22 % )

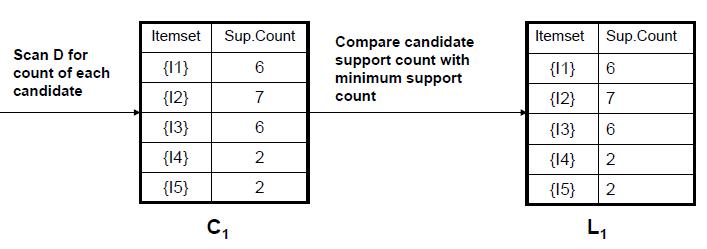
•Let minimum confidence required is 70%.

•We have to first find out the frequent itemset using Apriori algorithm.

•Then, Association rules will be generated using min. support & min. confidence.

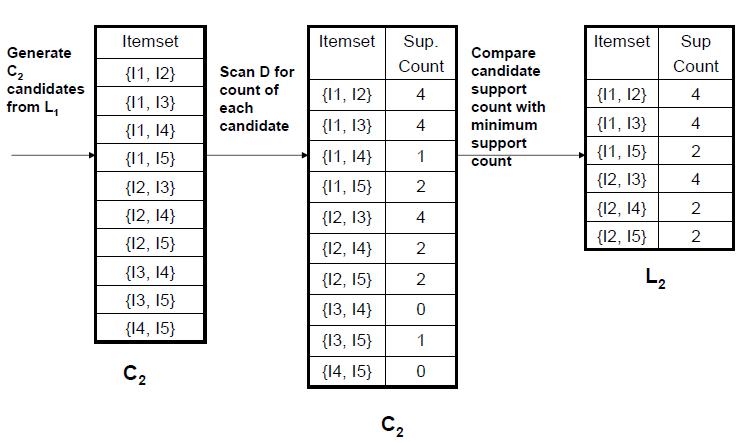


**Step 1**: Generating 1-itemset Frequent Pattern



The set of frequent 1-itemsets, L1, consists of the candidate 1-itemsets satisfying minimum support.

•In the first iteration of the algorithm, each item is a member of the set of candidate. **Step 2**: Generating 2-itemset Frequent Pattern

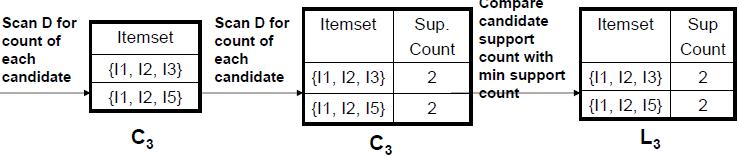


•To discover the set of frequent 2-itemsets, L2, the algorithm uses L1 *Join* L1to generate a candidate set of 2-itemsets, C2.

•Next, the transactions in D are scanned and the support count for each candidate itemset in C2is accumulated (as shown in the middle table).

•The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2having minimum support.

**Step 3**: Generating 3-itemset Frequent Pattern



The generation of the set of candidate 3-itemsets, C3, involves use of the Apriori Property.

•In order to find C3, we compute L2*Join*L2.

•C3= L2 *Join*L2 = {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}. •Now, Join steps complete and Prune step will be used to reduce the size of C3. Prune step helps to avoid heavy computation due to large Ck.

Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that four latter candidates cannot possibly be frequent. How?

•**For example**, let’s take {I1, I2, I3}.The 2-item subsets of it are {I1, I2}, {I1, I3} & {I2, I3}. Since all 2-item subsets of {I1, I2, I3} are members of L2, We will keep {I1, I2, I3} in C3. •Lets take another example of {I2, I3, I5} which shows how the pruning is performed. The 2-item subsets are {I2, I3}, {I2, I5} & {I3, I5}.

•BUT, {I3, I5} is not a member of L2and hence it is not frequent violating Apriori Property.

Thus We will have to remove {I2, I3, I5} from C3.

•Therefore, C3= {{I1, I2, I3}, {I1, I2, I5}} after checking for all members of result of Join operation for **Pruning**.

•Now, the transactions in D are scanned in order to determine L3, consisting of those candidates 3-itemsets in C3having minimum support.

**Step 4**: Generating 4-itemset Frequent Pattern

•The algorithm uses L3 *Join*L3to generate a candidate set of 4-itemsets, C4. Although the join results in {{I1, I2, I3, I5}}, this itemset is pruned since its subset {{I2, I3, I5}}is not frequent. •Thus, C4= φ, and algorithm terminates, having found all of the frequent items. This completes our Apriori Algorithm.

These frequent itemsets will be used to generate strong association rules( where strong association rules satisfy both minimum support & minimum confidence).

**Step 5:** Generating Association Rules from Frequent ItemsetsProcedure:

•For each frequent itemset ***“l”,***generate all nonempty subsets of ***l.***

•For every nonempty subset ***s***of ***l***, output the rule **“s** **(l-s)”**if

**support\_count(l) / support\_count(s) >= min\_conf**

where min\_conf is minimum confidence threshold.

•Back to Example:

We had L = {{I1}, {I2}, {I3}, {I4}, {I5}, {I1,I2}, {I1,I3}, {I1,I5}, {I2,I3}, {I2,I4}, {I2,I5}, {I1,I2,I3}, {I1,I2,I5}}.

–Let’s take ***l*** = {I1,I2,I5}.

–It’s all nonempty subsets are {I1,I2}, {I1,I5}, {I2,I5}, {I1}, {I2}, {I5}.

Let minimum confidence threshold is , say 70%.

•The resulting association rules are shown below, each listed with its confidence.

–R1: I1 ^ I2 I5

•Confidence = sc{I1,I2,I5}/sc{I1,I2} = 2/4 = 50%

•R1 is rejected.

–R2: I1 ^ I5 I2

•Confidence = sc{I1,I2,I5}/sc{I1,I5} = 2/2 = 100%

•R2 is selected.

–R3: I2 ^ I5 I1

•Confidence = sc{I1,I2,I5}/sc{I2,I5} = 2/2 = 100%

•R3 is selected.

–R4: I1 I2 ^ I5

•Confidence = sc{I1,I2,I5}/sc{I1} = 2/6 = 33%

•R4 is rejected.

–R5: I2 I1 ^ I5

•Confidence = sc{I1,I2,I5}/{I2} = 2/7 = 29%

•R5 is rejected.

–R6: I5 I1 ^ I2

•Confidence = sc{I1,I2,I5}/ {I5} = 2/2 = 100%

•R6 is selected.

In this way, we have found three strong association

rules.