# Assignment No. B10

## 1 Title

Apriori Algorithm.

## 2 Problem Definition

Implement Apriori approach for data mining to organize the data items on a shelf using following table of items purchased in a Mall

| Transaction ID | Item1 | Item2 | Item3 | Item4 | Item 5 | Item6 |
|---|---|---|---|---|---|---|
| T1 | Mango | Onion | Jar | Key-chain | Eggs | Chocolates |
| T2 | Nuts | Onion | Jar | Key-chain | Eggs | Chocolates |
| T3 | Mango | Apple | Key-chain | Eggs | - | - |
| T4 | Mango | Toothbrush | Corn | Key-chain | Chocolates | - |
| T5 | Corn | Onion | Onion | Key-chain | Knife | Eggs |

## 3 Objective

- To implement Apriori algorithm.
- To understand the concept of association.

## 4 Software and Hardware Requirements

1. 64 bit Machine i3/i5/i7
2. 64-bit open source Linux OS Fedora 20
3. gcc editor

# Theory

## Introduction:

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database:
Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent itemsets in a database.
2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data.
The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length from item sets of length . Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent -length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.The pseudo code for the algorithm is given below for a transaction database , and a support threshold of . Usual set theoretic notation is employed, though note that is a multiset is the candidate set for level. At each step, the algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma. accesses a field of the data structure that represents candidate set , which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

## Apriori Algorithm:

**Algorithm 6.2** Rule generation of the *Apriori* algorithm.

```
1: for each frequent k-itemset f_k, k ≥ 2 do
2:     H_1 = f_k          {1-item consequents of the rule.}
3:     call ap-genrules(f_k, H_1).
4: end for
```

**Algorithm 6.3** Procedure `ap-genrules`$(f_k, H_m)$.

```
1: for each h_m ∈ H_m do
2:     conf = σ(f_k)/σ(f_k − h_m).
3:     if conf ≥ minconf then
4:         output the rule (f_k − h_m) ⟶ h_m.
5:     else
6:         delete h_m from H_m.
7:     end if
8: end for
9: if k > m + 1 then
10:    H_{m+1} = apriori-gen(H_m).
11:    call ap-genrules(f_k, H_{m+1}).
12: end if
```

## Applications:

- Application of the Apriori algorithm for adverse drug reaction detection: The objective is to use the Apriori association analysis algorithm for the detection of adverse

drug reactions (ADR) in health care data. The Apriori algorithm is used to per- form association analysis on the characteristics of patients, the drugs they are taking, their primary diagnosis, co-morbid conditions, and the ADRs or adverse events (AE) they experience. This analysis produces association rules that indicate what combinations of medications and patient characteristics lead to ADRs.

- Application of Apriori Algorithm in Oracle Bone Ins cription Explication: Oracle Bone Inscription (OBI) is one of the oldest writing in the world, but of all 6000 words found t ill now there are only about 1500 words that can be exp licated explicitly. So explication for OBI is a key and open problem in this field. Exploring the correlation between the OBI words by Association Rules algorithm can aid in the research of explication for OBI. Firstly the OBI data extracted from the OBI corpus are preprocessed; with these processed data as input for Apriori algorithm we get the frequent itemset. And combined by the interestingness measurement the strong association rules between OBI words are produced. Experimental results on the OBI corpus demonstrate that this proposed method is feasible and effective in finding semantic correlation for OBI.

## 5 Mathematical Model :

Let S be the solution perspective of the system such that,
S ={$S_t$,$E_t$,X,Y,DD,NDD,$F_{me}$,Sc,Fc}
where,

$S_t$ = start state
such that Y={} $E_t$ = end state
such that Y={$I$} I=Frequent item set.
X=Set of Inputs X = {$X1, X2$}
X1= set of transactions
X2 = Min support.

Y = set of outputs
Y={$I$}
I=Frequent item set.

$F_{me}$=set of functions.
$F_{me}$={f1,f2,f3}
where,
f1= function to accept the transaction and min support.
f2= function to generate the candidate item set.
f3= function to generate frequent item set using apriori algorithm.

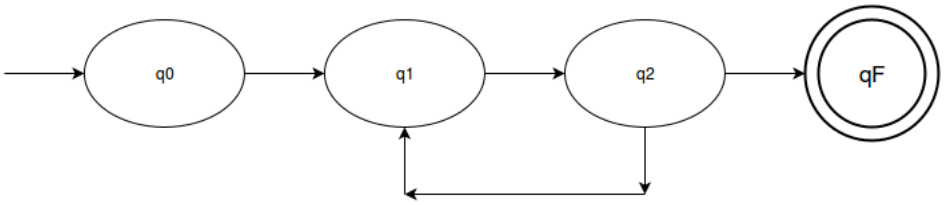
DD = Deterministic Data
= {$|X|$}

NDD (Non Deterministic Data)
= {$s1$}
s1=Number of candidate item sets after each iteration.

Sc = Implemented successfully

Fc = Unsuccessful

## 6 State Diagram :




q0 → input transactions state

q1 → self join state

q2 $\rightarrow$ pruning state

q3 $\rightarrow$ final state

## 7 Conclusion

Thus we successfully used Apriori algorithm to find the frequent item set from the given transaction..