Assignment NO - 8 Date : / /20

Title :- Case study (market Basket Analysis)

problem statement :- A mall has no. of items for sale. Build a required database to develope BA & I tool for considering one aspects of growth of business such as organization of products based on demand & patterns.

Input :- Transaction outbase & minimum support.

output :- frequent item sets, Association rules & graphical representation of rules as per confidences & lift.

Theory :-

By convention, the algorithm assume that items within a transaction or item set are sorted in lexicographic order. It employees an iterative approach known as a level-wise search, where item set are used to explore k itemset, first, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item & collecting those item that satisfy minimum support



Pooja

Date : / / 20

The resulting set is denoted as $L_1$. Next, $L_1$ is used to find $L_2$, the set of frequent 2 item set, which is then used to find $L_3$ & so on. until no more frequent k-itemset can be found.
To improve the efficiency of the level-wise generation of frequent itemsets, an important property called Apriori property is used to reduce the search space.
Apriori property - All nonempty subsets of a frequent itemset must also be frequent.
A two step process is used to find $L_4$ from $L_{4-1}$
for $K \geq 2$
1. The join step 1- To find $L_4$ is set of candidate k-itemset is generated by joining $L_{K-1}$ with itself. This set of candidate is denoted by $C_K$. Let $L_2$ be itemset in $L_{K-1}$ The notation $l_i[j]$ refer to jth item in $l_i$. Thus in $l_i$, the last item & the next to last item are given respectively by $l_i[k-1]$ & $l_i[k-2]$.
Any two itemset $L_{k-1}$ are joined if their first (k-2) items are in common. That is members '1' & '2 are joined if $(l_2[1] = L_2[1] \wedge (l_2[2] = l_2[J]) \wedge \dots \wedge (l_1(k-2) = L_2[k-2]) \wedge (l_L[k-1] < l_2[k-1])$

Date : / / 20

2. The prune step: set $C_k$ is a subset of $L_k$, because although all frequent k-itemset are included in $C_k$, its members may or may not frequent one could scan database to determine & eliminate any, itemset that does not meet the minimum support threshold. This would the given $L_k$ However $C_k$ can be huge & so this could be very time-consuming. To eliminate the infrequent itemsets the Apriori property is used as follows.

- An example of the use of Apriori Algorithm
We illustrate the use of Aprironi algorithm for finding frequent itemsets in our transaction database. count number of occurance of each item.

| CI itemset | support count | LI itemset | support count |
|---|---|---|---|
| {1} | 6 | {1} | 6 |
| {2} | 7 | {2} | 7 |
| {3} | 6 | {3} | 6 |
| {4} | 2 | {4} | 2 |
| {5} | 2 | {5} | 2 |
| {6} | 1 | | |

TO discover the set of frequent 2-itemset 1,2, the algorithm joins L with self w & generate candidate itemsets of 2-itemset.



Pooja

Date : / /20

Next $C_3$ is generated by joining $L_2$ itself
The result is $C_3 = \{\{1,2,3\}, \{1,2,5\}, \{1,3,5\}, \{2,3,4\}, \{3,3,5\}, \{2,4,5\}\}$ $C_3$ is pured using Apriori Property. All nonempty subsets of frequent itemset must also be frequent from way each candidate of $C_3$ is formed.
The candidate set
Since $\{2,3\}$ is a frequent itemset, we keep $\{1,2,3\}$ in $C_3$ since $\{2,5\}$ is a frequent itemset we keep $\{1,2,5\}$ in $C_3$ since $\{3,5\}$ is not frequent itemset, we remove $\{1,3,5\}$ from $C_3$. Since $\{3,4\}$ is not frequent itemset, we remove $\{2,3,4\}$ from $C_3$ since $\{3,5\}$ is not frequent itemset, we remove $\{2,3,5\}$ from $C_3$ since $\{4,5\}$ is not frequent itemset, we remove $\{2,4,5\}$ from $C_3$.
Therefore after pruning $C_3$ given by
$C_3$ itemset
$\{1,2,3\}$
$\{1,2,5\}$
The transaction in D are scanned to determine $L_3$ consisting of those candidates 3 itemsets in $C_3$ having at least minimum support.

| $C_3$ itemset | Support count |
|---|---|
| $\{1,2,3\}$ | 2 |
| $\{1,2,5\}$ | 2 |



Pooja

Date : / /20

Note that no candidate are removed from $C_2$ during the Pruning step.

| $C_2$ itemset |
|---|
| {1,2} |
| {1,3} |
| {1,4} |
| {1,5} |
| {2,3} |
| {2,4} |
| {2,5} |
| {3,4} |
| {3,5} |
| {4,5} |

| $C_2$ itemset | Support count |
|---|---|
| {1,2} | 4 |
| {1,3} | 4 |
| {1,4} | 1 |
| {1,5} | 2 |
| {2,3} | 4 |
| {2,4} | 2 |
| {2,5} | 2 |
| {3,4} | 0 |
| {3,5} | 1 |
| {4,5} | 0 |

Next the transaction in D are scaned & the support count of each candidate in $C_2$ is accumulated. consisting of those candidate 2-itemset in $C_2$ having minimum support.

| $L_2$ itemset | Support count |
|---|---|
| {1,2} | 4 |
| {1,3} | 4 |
| {1,4} | 2 |
| {2,3} | 4 |
| {2,4} | 2 |
| {3,4} | 2 |

Date : / / 20

| $L_3$ itemset | support count |
|---|---|
| {1, 2, 3} | 2 |
| {1, 2, 5} | 2 |

Finally $L_3$ joined with itself to generate a candidate set of 4-itemset $C_4$.
This result in a single itemset {1, 2, 3, 5}
However the itemset is pruned since its subset {3, 5} is not frequent. Thus $C_4 = \phi$ & algorithm terminate, having found all of the frequent itemsets.

Analysis - 1. Observe the graphs for generated rules with different support confidence & lift.
2. observe top rules & use this patterns for organization of products.

Conclusion :-
Thus the Groceries dataset is used to generate rules & applied rules for organization of products based on patterns & demand. Frequent itemset are found using apriori algorithm based on rule data mining technique. observations are recorded in terms of graph.

Pooja