PART-2: ASSOCIATION ANALYSIS

Description of Apriori Algorithm

Apriori algorithm is used to generate association mining rules which helps correlate a set of related items in any transaction. It consists of mainly two parts:

Step 1: Generating frequent item sets having support >= minimum support

Step 2: Generating association rules having confidence >= minimum confidence

Step 1: Generating frequent item sets having support >= minimum support

This uses a bottom up approach where frequent item sets which are found are extended one step at a time. It generates a candidate item set and then checks the support of the generated candidate item set. If the support is equal or more than the minimum support, then it adds it to the list of frequent item sets.

Step 2: Generating association rules having confidence >= minimum confidence

Based on each frequent item set generated, for subsets of each frequent item set it verifies if the confidence is equal to or more than the minimum confidence, it generates and adds the association rule.

Workflow

- In our code, we first process the given dataset from the given file and append gene numbers appropriately and return the dataset as a list. We read the minimum support and confidence rules from the user input.
- We then generate the list of possible candidates – C1 by reading the loaded dataset.
- A dictionary is created where we the store the candidate as a key (Ex:'G1_Up') and a list of transaction ids as the value for all candidates.
- We then generate L1 by performing set intersection of list of transaction Ids by passing C1 and checking if the support of the particular candidate item set passes the minimum support threshold. If so, we add it to the frequent itemset list and also record the frequent itemset as a key and support value as value in dictionary.
- From the frequent itemset generated,we find all possible combinations to extend the frequent itemset length by 1 and the same process of generating Ck and Lk is followed until no more frequent item sets can be generated.As each Ck and Lk is generated it is added to main frequentitem set list and corresponding support values are recorded in a main dictionary.
- Once we have the frequent itemsets and the support value dictionary, for each frequent item set generated we generate all possible subsets of it. We start with frequent item sets of length 2 and greater. And for each pair of subset we calculate the confidence according to the confidence formula and if it crosses the minimum confidence, we generate the rule and add it to our main rule set. This processing stops once all the frequent itemsets have been checked for the confidence. The generated rules are printed.
- We have also written all 3 template methods as specified which queries the generated rules list.

1. Results obtained by different support values in the Apriori algorithm:

Support = 30%
Number of length-1 frequent itemsets: 196
Number of length-2 frequent itemsets: 5340
Number of length-3 frequent itemsets: 5287
Number of length-4 frequent itemsets: 1517
Number of length-5 frequent itemsets: 438
Number of length-6 frequent itemsets: 88
Number of length-7 frequent itemsets: 11
Number of length-8 frequent itemsets: 1
Number of all length frequent itemsets 12878

Support = 40%
Number of length-1 frequent itemsets: 167
Number of length-2 frequent itemsets: 753
Number of length-3 frequent itemsets: 149
Number of length-4 frequent itemsets: 7
Number of length-5 frequent itemsets: 1
Number of all length frequent itemsets: 1077

Support = 50%
Number of length-1 frequent itemsets: 109
Number of length-2 frequent itemsets: 63
Number of length-3 frequent itemsets: 2
Number of all length frequent itemsets: 174

Support = 60%
Number of length-1 frequent itemsets: 34
Number of length-2 frequent itemsets: 2
Number of all length frequent itemsets: 36

Support = 70%
Number of length-1 frequent itemsets: 7
Number of all length frequent itemsets: 7

2. Generating association rules based on the templates.

• Template 1: {RULE|HEAD|BODY} HAS ({ANY|NUMBER|NONE}) OF (ITEM1, ITEM2, ..., ITEMn)

(result11, cnt) = template1("RULE", "ANY", ['G59_Up'])
cnt = 26
(result12, cnt) = template1("RULE", "NONE", ['G59_Up'])
cnt = 91
(result13, cnt) = template1("RULE", 1, ['G59_Up', 'G10_Down'])
cnt = 39
(result14, cnt) = template1("HEAD", "ANY", ['G59_Up'])
cnt = 9
(result15, cnt) = template1("HEAD", "NONE", ['G59_Up'])

cnt = 108
(result16, cnt) = template1("HEAD", 1, ['G59_Up', 'G10_Down'])
cnt = 17
(result17, cnt) = template1("BODY", "ANY", ['G59_Up'])
cnt = 17
(result18, cnt) = template1("BODY", "NONE", ['G59_Up'])
cnt = 100
(result19, cnt) = template1("BODY", 1, ['G59_Up', 'G10_Down']
cnt = 24

• Template 2: SizeOf({HEAD|BODY|RULE}) ≥ NUMBER.

(result21, cnt) = template2("RULE", 3)
cnt = 9
(result22, cnt) = template2("HEAD", 2)
cnt = 6
(result23, cnt) = template2("BODY", 1)
cnt = 117

• Template 3: Any combined templates using AND or OR.

(result31, cnt) = template3("1or1", "HEAD", "ANY", ['G10_Down'],"BODY", 1,['G59_UP'])
cnt = 24
(result32, cnt) =template3("1and1", "HEAD", "ANY",['G10_Down'], "BODY", 1, ['G59_UP'])
cnt = 1
(result33, cnt) = template3("1or2", "HEAD", "ANY", ['G10_Down'],"BODY", 2)
cnt = 11
(result34, cnt) = template3("1and2", "HEAD", "ANY",['G10_Down'], "BODY", 2)
cnt = 0
(result35, cnt) = template3("2or2", "HEAD", 1, "BODY", 2)
cnt = 117
(result36, cnt) = template3("2and2", "HEAD", 1, "BODY", 2)
cnt =3

3)Demo – support = 60%
Confidence = 65%

Query: asso_rule.template3("1or2", "BODY", "ANY", ['G1_Down'], "HEAD", 2)
Result : 0