

Association Report

Team

Siddharth Pandey

Abhijeet Jadhav

Anirudh Dalmia

Apriori Algorithm:

Apriori algorithm is used for frequent item set mining and creating association rules. The algorithm is mainly used with the relational databases.

The algorithm firstly identifies the frequent items and merge them to create larger and larger itemsets. The process of creating the larger itemsets continues till the algorithm creates an empty itemset i.e. it is impossible to create a larger itemset from the existing itemsets.

Apriori algorithm uses the property according to which all the subsets of the frequent itemset are always frequent. For infrequent itemset all the supersets will also be infrequent.

Implementation of Algorithm:

The implementation of the algorithm firstly reads the input data text file. For each data point the value is changed from Up or Down to G 'Column Number' _Up or G 'Column Number' _Down. For each of these values count is calculated so as to calculate the support for each term. This is the first level and after calculating the support we just keep the items which as support more than that if the support specified by the user.

These items are stored into a list. These items with length 1 are used to create the itemset with length2. The length 2 itemsets are created by taking the union of 1 length itemsets. Similarly K-length itemset is created by taking union of K-1 length itemsets.

These itemsets are used to create association rules.

Association Part 1:

1. Support is 30%

Number of 1-length itemsets: 196

Number of 2-length itemsets: 5352

Number of 3-length itemsets: 5251

Number of 4-length itemsets: 1463

Number of 5-length itemsets: 288

Number of 6-length itemsets: 61

Number of 7-length itemsets: 3

2. Support is 40%

Number of 1-length itemsets: 167

Number of 2-length itemsets: 753

Number of 3-length itemsets: 149

Number of 4-length itemsets: 7

Number of 5-length itemsets: 1

3. Support is 50%

Number of 1-length itemsets: 109

Number of 2-length itemsets: 63

Number of 3-length itemsets: 2

4. Support is 60%

Number of 1-length itemsets: 34

Number of 2-length itemsets: 2

5. Support is 70%

Number of 1-length itemsets: 7

Association Part 2:

Template 1:

1. (result11, cnt) = asso_rule.template1("RULE", "ANY", ['G59_UP'])
Number of rules: 0

2. (result12, cnt) = asso_rule.template1("RULE", "NONE", ['G59_UP'])
Number of rules: 117

3. (result13, cnt) = asso_rule.template1("RULE", 1, ['G59_UP', 'G10_Down'])

Number of rules: 15

4. (result14, cnt) = asso_rule.template1("HEAD", "ANY", ['G59_UP'])
Number of rules: 0
5. (result15, cnt) = asso_rule.template1("HEAD", "NONE", ['G59_UP'])
Number of rules: 117
6. (result16, cnt) = asso_rule.template1("HEAD", 1, ['G59_UP', 'G10_Down'])
Number of rules: 7
7. (result17, cnt) = asso_rule.template1("BODY", "ANY", ['G59_UP'])
Number of rules: 0
8. (result18, cnt) = asso_rule.template1("BODY", "NONE", ['G59_UP'])
Number of rules: 117
9. (result19, cnt) = asso_rule.template1("BODY", 1, ['G59_UP', 'G10_Down'])
Number of rules: 8

Template 2:

1. (result21, cnt) = asso_rule.template2("RULE", 3)
Number of rules: 117
2. (result22, cnt) = asso_rule.template2("HEAD", 2)
Number of rules: 117
3. (result23, cnt) = asso_rule.template2("BODY", 1)
Number of rules: 117

Template 3:

1. (result31, cnt) = asso_rule.template3("1or1", "HEAD", "ANY",
['G10_Down'], "BODY", 1, ['G59_UP'])
Number of rules: 7

2. (result32, cnt) = asso_rule.template3("1and1", "HEAD", "ANY",
['G10_Down'], "BODY", 1, ['G59_UP'])
Number of rules: 0
3. (result33, cnt) = asso_rule.template3("1or2", "HEAD", "ANY",
['G10_Down'], "BODY", 2)
Number of rules: 117
4. (result34, cnt) = asso_rule.template3("1and2", "HEAD", "ANY",
['G10_Down'], "BODY", 2)
Number of rules: 117
5. (result35, cnt) = asso_rule.template3("2or2", "HEAD", 1, "BODY", 2)
Number of rules: 117
6. (result36, cnt) = asso_rule.template3("2and2", "HEAD", 1, "BODY", 2)
Number of rules: 117