CSE 601: Data Mining and Bioinformatics

Project-1

Project 1: Dimensionality Reduction & Association Analysis

Varsha Ravichandiran (50315099)

Charan reddy bodennagari (50338186)

Sri Charan Chintapenta (50313858)

CSE 601: Data Mining and Bioinformatics

Project-1

Objective:

To generate frequent itemsets for the given support threshold using Apriori Algorithm and generating the rules satisfying the confidence threshold for the given gene expression transactional datset.

We are given a data set with each transaction having a gene expression along with the respective medical condition. Our goal is to find patterns in the data.

This includes two parts:

Frequent set Item using apriori algorithm:

The goal is to generate item sets that are frequently occurring in the data, which satisfy a minimum support value. For that we need to find the support for all the item sets and prune the item sets which do not satisfy the support threshold provided.

Association rule generation with the frequent item sets:

The goal here is to generate association rules from the set of frequent items which satisfy the given confidence threshold. The rules generated should have item sets of length at least 2 which means there should a HEAD and a BODY for the rule.

Frequent item set generation using Apriori:

Apriori algorithm is an efficient algorithm in mining frequent item sets from a data set. Apriori works on the major principle that if an itemset is infrequent then all of its supersets are infrequent, which is called anti monotone property.

Apriori is done in two steps

Joining:

In this step we generate item sets of length (k+1) from item sets of length k.

Pruning:

This steps scans through the all samples and calculate the support for each of the itemset. If the candidate item does not meet the support requirement. From the antimonotone property we can conclude that if an item is not frequent then all of its subsets are infrequent, so we can drop the itemset from the frequent itemset list. By this way we can reduce the number of candidate item sets.

CSE 601: Data Mining and Bioinformatics

Project-1

Flow of Association rule algorithm:

As our data set value of each of the gene is only have two values Up and Down, in order to distinguish between different genes, we modify the given set by appending the gene id (G1, G2… respectively) to its value, so that each column has different gene id.

We maintain a Master dictionary which would contain all the item sets and their respective support.

With the modified filed values, we generate the set of all the different items from the dataset, that forms a candidate set of length i, where the value of i is 1 initially.

Then we Calculate the support for each item in the candidate item set, and prune the item sets that doesn't meet the minimum support criteria given.

Now we run the apriori algorithm generating item sets by joining and pruning them based on the support threshold.

We then take the filtered candidates list satisfying the support threshold to generate the frequent item sets of length i+1 .

Then we repeat the above process to prune the item sets generated to get the frequent item sets satisfying the support threshold.

This process is continued until all the frequent item sets have been generated.

With all the frequent items sets generated, we calculate the associate rules.

For every itemset in the frequent itemset we generate all the combinations of the association rules that can be generated from that itemset.

We calculate the confidence for each association rule generated. If the rule meets the min confidence threshold we keep the rule, if the confidence is below the threshold we drop the rule.

Now we have all the association rules that meet the support and confidence criteria.

CSE 601: Data Mining and Bioinformatics

Project-1

Results :

All the results are for the rules generated for support threshold 50 percent and confidence threshold 70 percent.

Frequent item sets generated for support threshold – 30 percent :

Number of frequent itemsets of length-1 : 196

Number of frequent itemsets of length-2 : 5340

Number of frequent itemsets of length-3 : 5287

Number of frequent itemsets of length-4 : 1518

Number of frequent itemsets of length-5 : 438

Number of frequent itemsets of length-6 : 88

Number of frequent itemsets of length-7 : 11

Number of frequent itemsets of length-8 : 1

Total Number of frequent itemsets : 12879

Frequent item sets generated for support threshold – 40 percent :

Number of frequent itemsets of length-1 : 167

Number of frequent itemsets of length-2 : 753

Number of frequent itemsets of length-3 : 149

Number of frequent itemsets of length-4 : 7

Number of frequent itemsets of length-5 : 1

Total Number of frequent itemsets : 1077

Frequent item sets generated for support threshold – 50 percent :

CSE 601: Data Mining and Bioinformatics

Project-1

Number of frequent itemsets of length-1 : 109

Number of frequent itemsets of length-2 : 63

Number of frequent itemsets of length-3 : 2

Total Number of frequent itemsets : 174

Frequent item sets generated for support threshold – 60 percent :

Number of frequent itemsets of length-1 : 34

Number of frequent itemsets of length-2 : 2

Total Number of frequent itemsets : 36

Frequent item sets generated for support threshold – 70 percent :

Number of frequent itemsets of length-1 : 7

Total Number of frequent itemsets : 7

Number of rules generated for support 30% and confidence 70% : 31759

Number of rules generated for support 40% and confidence 40% : 2528

Number of rules generated for support 50% and confidence 70% : 117

Number of rules generated for support 60% and confidence 60% : 4

For Template 1 :

(result11, cnt) = asso_rule.template1("RULE", "ANY", ['G59_UP']) : 26

(result12, cnt) = asso_rule.template1("RULE", "NONE", ['G59_UP']) : 91

(result13, cnt) = asso_rule.template1("RULE", 1, ['G59_UP', 'G10_Down']) : 39

(result14, cnt) = asso_rule.template1("HEAD", "ANY", ['G59_UP']) : 17

(result15, cnt) = asso_rule.template1("HEAD", "NONE", ['G59_UP']) : 100

(result16, cnt) = asso_rule.template1("HEAD", 1, ['G59_UP', 'G10_Down']) : 24

(result17, cnt) = asso_rule.template1("BODY", "ANY", ['G59_UP']) : 9

(result18, cnt) = asso_rule.template1("BODY", "NONE", ['G59_UP']) : 108

(result19, cnt) = asso_rule.template1("BODY", 1, ['G59_UP', 'G10_Down']) : 17

Template 2 :

(result21, cnt) = asso_rule.template2("RULE", 3) : 9

(result22, cnt) = asso_rule.template2("HEAD", 2) : 3

(result23, cnt) = asso_rule.template2("BODY", 1) : 117

Template 3 :

(result31, cnt) = asso_rule.template3("1or1", "HEAD", "ANY", ['G10_Down'], "BODY", 1, ['G59_UP']) : 16

(result32, cnt) = asso_rule.template3("1and1", "HEAD", "ANY", ['G10_Down'], "BODY", 1, ['G59_UP']) : 0

(result33, cnt) = asso_rule.template3("1or2", "HEAD", "ANY", ['G10_Down'], "BODY", 2)

(result34, cnt) = asso_rule.template3("1and2", "HEAD", "ANY", ['G10_Down'], "BODY", 2) : 13

(result35, cnt) = asso_rule.template3("2or2", "HEAD", 1, "BODY", 2) : 117

(result36, cnt) = asso_rule.template3("2and2", "HEAD", 1, "BODY", 2) : 6