CSE 601 : Data Mining and Bioinformatics
Project 1 Part 2: Association Analysis

NAGA VAISHNAVI PAKYALA (nagavais@buffalo.edu)
HRISHIKESH NITTURKAR (hnitturk@buffalo.edu)

Association Analysis

The aim of the project is to implement Apriori Algorithm to find frequent itemset and generate association rules. Given a set of transactions, the goal of association rule mining is to find all rules whose **support ≥ minsup threshold** and **confidence ≥ minconf threshold.**

This implementation is a two-step approach:

1. **Frequent Itemset Generation:** Generate all itemsets whose support ≥ minsup
2. **Rule Generation:** Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

IMPLEMENTATION OF APRIORI ALGORITHM

Language used: Python 3.6

Frequent Itemset Generation:

Dynamic Inputs: Minimum Support considered

1. Read the dataset from the text file imported and the gene expression from the data "DownDownDownUp...AML" is converted to the expected output which is "G1_DownG2_DownG3_DownG4_Up...AML".
2. Generate the frequent itemsets of length-1 by obtaining the unique items from the given dataset. Add the unique items to a list. Add the unique items and their count to a dictionary
3. From the unique items, generate all possible combinations of itemsets.
4. For every combination, prune the itemsets that have a support less than the Minimum Support. Add the frequent itemsets to a list and generate the number of frequent itemsets of increasing lengths. Repeat this process until this list becomes empty.

Results:

Input: Minimum Support considered= 50%

Output:

Number of length-1 frequent itemsets : 109

Number of length-2 frequent itemsets : 63

Number of length-3 frequent itemsets : 2

Number of all frequent itemsets : 174

Input: Minimum Support considered= 30%

Output:

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 : 1518

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 frequent itemsets : 12879

Input: Minimum Support considered= 40%

Output:

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 frequent itemsets : 1077

Input: Minimum Support considered= 60%

Output:

Number of length-1 frequent itemsets : 34

Number of length-2 frequent itemsets : 2

Number of all frequent itemsets : 36

Input: Minimum Support considered= 70%

Output:

Number of length-1 frequent itemsets : 7

Number of all frequent itemsets : 7

Association Rule Generation:

Dynamic Inputs: Minimum Support , Minimum Confidence, Template Query

1. Iterate over the frequent itemsets that are obtained above, generate rules that have a confidence greater than or equal to the Minimum confidence where each rule[head--> body] is a combination of head and body of each frequent itemset.
2. Once the rules are in hand, parse the template query from the input and produce the desired outcome for the query. The templates can be 1, 2 or 3.

   Template1: {RULE|HEAD|BODY}HAS({ANY|1|NONE})OF(ITEM1,ITEM2,...,ITEMn)

   Template2: SizeOf({HEAD|BODY|RULE})≥NUMBER

   Template3: Any combined templates using AND or OR. For example: BODY HAS (1) OF (Disease) AND HEAD HAS (NONE) OF (Disease)

3. For Template 1:

   ANY : Output all the frequent itemsets and the number of rows that contain the provided item(s) depending upon RULE, BODY AND HEAD.

   NONE: Output all the frequent itemsets and the number of rows that DO NOT contain the provided item(s) depending upon RULE, BODY AND HEAD.

   1 : Output all the frequent itemsets and the number of rows that contain the provided item(s) only once depending upon RULE, BODY AND HEAD.

4. For Template 2:

   Output all the frequent itemsets and the number of rows that have a total itemset count greater than or equal to the provided itemset count

5. For Template 3:

   Output all the frequent itemsets and the number of rows based on AND [Intersection] or OR [Union] operations on the templates specified.

Results:

Input: Minimum Support =50%, Minimum Confidence =70%

Total number of Rules generated: 117

TEMPLATE 1 QUERIES

Query 1 [(result11,cnt)=asso_rule.template1("RULE","ANY",['G59_UP'])]**:**

Output: Number of rows returned: **26**

Query 2 (result12,cnt)=asso_rule.template1("RULE","NONE",['G59_UP'])**:**

Output: Number of rows returned: **91**

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

Output: Number of rows returned: **39**

Query 4 (result14,cnt)=asso_rule.template1("HEAD","ANY",['G59_UP']):

Output: Number of rows returned: **17**

Query 5 (result15,cnt)=asso_rule.template1("HEAD","NONE",['G59_UP']):

Output: Number of rows returned: **100**

Query 6 (result16,cnt)=asso_rule.template1("HEAD",1,['G59_UP','G10_Down']):

Output: Number of rows returned: **24**

Query 7 (result17,cnt)=asso_rule.template1("BODY","ANY",['G59_UP']):

Output: Number of rows returned: **9**

Query 8 (result18,cnt)=asso_rule.template1("BODY","NONE",['G59_UP']):

Output: Number of rows returned: **108**

Query 9 (result19,cnt)=asso_rule.template1("BODY",1,['G59_UP','G10_Down']):

Output: Number of rows returned: **17**

TEMPLATE 2 QUERIES

Query 1 (result21,cnt)=asso_rule.template2("RULE",3)**:**

Output: Number of rows returned: **9**

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

Output: Number of rows returned: **3**

Query 3 (result23,cnt)=asso_rule.template2("BODY",1):

Output: Number of rows returned: **117**

TEMPLATE 3 QUERIES

Query 1
(result31,cnt)=asso_rule.template3("1or1","HEAD","ANY",['G10_Down'],"BODY",1,['G59_Up']):

Output: Number of rows returned: **16**

Query 2
(result32,cnt)=asso_rule.template3("1and1","HEAD","ANY",['G10_Down'],"BODY",1,['G59_UP'])

Output: Number of rows returned: **0**

Query 3 (result33,cnt)=asso_rule.template3("1or2","HEAD","ANY",['G10_Down'],"BODY",2):

Output: Number of rows returned: **13**

Query 4 (result34,cnt)=asso_rule.template3("1and2","HEAD","ANY",['G10_Down'],"BODY",2):

Output: Number of rows returned: **0**

Query 5 (result35,cnt)=asso_rule.template3("2or2","HEAD",1,"BODY",2):

Output: Number of rows returned: **117**

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

Output: Number of rows returned: **6**