Homework 1: Association Analysis

Code and report due: September 18, 2022, 11:59pm

General Introduction:
In this homework, you are asked to implement Apriori and association rule generation algorithms. Everyone should submit their codes and a report via ICON.

Recommended Programming Language:
Python

Dataset Description:
The dataset is about gene expressions (*association-rule-test-data.txt*) and can be found on ICON under Homework 1. Each row stands for a patient/sample. The last column is the disease name. For the rest columns, they are gene expressions with values Up or Down (Binary Value). For example, the row "Down Down Down Up … AML" can be interpreted as "G1_ Down G2_ Down G3_ Down G4_Up … AML", and AML is a disease name.

Required Tasks:

1. Implement the Apriori algorithm to find all frequent itemsets. Report the number of frequent itemsets for support of 30%, 40%, 50%, 60%, and 70%, respectively. Please see *Template* for details.
   You **should not** directly call any existing function or package that implements Apriori. Apriori algorithm should be implemented by yourself.
2. Generate association rules based on the templates. The following are templates:
   - Template 1: {RULE|BODY|HEAD} HAS ({ANY|NUMBER|NONE}) OF (ITEM1, ITEM2, ..., ITEMn)
   - Template 2: SizeOf({BODY|HEAD|RULE}) ≥ NUMBER.
   - Template 3: Any combined templates using AND or OR. For example: HEAD HAS (1) OF (Disease) AND BODY HAS (NONE) OF (Disease)

   Below is an example illustrating RULE, BODY and HEAD in the templates: Assume we obtain a **RULE** {G1_Up, G3_Down} → {G4_Down, G34_Up}. {G1_Up, G3_Down} is **BODY** and {G4_Down, G34_Up} is **HEAD**.

   If support = 50% and confidence = 70% are given, you need to generate **all the rules** satisfying these requirements. In your report, you are asked to show the number of rules generated. However, in your code, you need to make sure that support and confidence can be changed to other values in new queries, and show and count the resulting rules you generate for each query. Please see *Template* for details.

3. Prepare your submission. Make a zipped folder named "[HawkID]-*Association*.zip", where "[HawkID]" refers to your HawkID. In the folder, you should include:
   a. Report: The report should include:
      i. Describe Apriori algorithm and the flow of the association rule generation algorithm briefly.
      ii. The answers of aforementioned queries in required tasks 1&2 (Number of frequent itemsets or generated rules).
      iii. The filename for your report should be "report.pdf"

b. A folder named *Code*, which contains all codes used in this part. Inside the folder, please have a file *README* which describes how to run your code. The folder name for your code should be "code.zip".
c. An ICON assignment page have been created for homework1. Please submit your zipped folder there.

Note that copying code/results/report from another group or source is not allowed and may result in an F grade.

HW1 dataset explanation

In the data file, we have 100 rows, where each row stands for a patient/sample.

column 1 - column 100 correspond to 100 genes (let's give each gene an id from 1 to 100). Each gene can be up-regulated (correspondingly the value in that cell is "UP") or down-regulated (Corresponding, "Down"). **We treat each gene as two distinct items**, for example, gene1_up, gene1_down. Thus we have 200 gene items.

Column 101 (the last column) is disease name. In total, we have four different diseases (AML, ALL, Breast Cancer, Colon Cancer). Each disease is an item. Thus, we have **204 items**: 200 gene items (gene1_up, gene1_down, gene2_up, gene2_down, ...) + 4 disease items.

Your program should parse the file, process the information as indicated above, and implement association rule algorithm.

Your program should be able to output association rules based on various criteria. The format of the association rules should be something like this:

(gene1_up, gene2_down, gene_50_up) --> gene77_down

(gene1_up, gene2_down, gene_3_up) --> AML

...

Template

1. In Part 1 of required tasks for Apriori, please list the results obtained by different support values using the following format in your report:
   Support is set to be 50%

   number of length-1 frequent itemsets: ??

   number of length-2 frequent itemsets: ??

   number of length-3 frequent itemsets: ??

   ……….

   number of all lengths frequent itemsets: ??

2. In Part 2 of required tasks for Apriori, your program is expected to answer the template queries. The following are queries written by Python to give you an idea of what it should be like.

For template 1, we have 9 possible keywords combinations:

```
(result11, cnt) = asso_rule.template1("RULE", "ANY", ['G59_UP'])
(result12, cnt) = asso_rule.template1("RULE", "NONE", ['G59_UP'])
(result13, cnt) = asso_rule.template1("RULE", 1, ['G59_UP', 'G10_Down'])
(result14, cnt) = asso_rule.template1("BODY", "ANY", ['G59_UP'])
(result15, cnt) = asso_rule.template1("BODY", "NONE", ['G59_UP'])
(result16, cnt) = asso_rule.template1("BODY", 1, ['G59_UP', 'G10_Down'])
(result17, cnt) = asso_rule.template1("HEAD", "ANY", ['G59_UP'])
(result18, cnt) = asso_rule.template1("HEAD", "NONE", ['G59_UP'])
(result19, cnt) = asso_rule.template1("HEAD", 1, ['G59_UP', 'G10_Down'])
```

For template 2, we have 3 keywords choices:

```
(result21, cnt) = asso_rule.template2("RULE", 3)
(result22, cnt) = asso_rule.template2("BODY", 2)
(result23, cnt) = asso_rule.template2("HEAD", 1)
```

For template 3, you need to implement AND/OR logical operator to connect two parts which can be from either template 1 or template 2. For example, “1or1” means “a query of Template1 OR another query of Template1”.

```
(result31, cnt) = asso_rule.template3("1or1", "BODY", "ANY", ['G10_Down'], "HEAD", 1, ['G59_UP'])
(result32, cnt) = asso_rule.template3("1and1", "BODY", "ANY", ['G10_Down'], "HEAD", 1, ['G59_UP'])
(result33, cnt) = asso_rule.template3("1or2", "BODY", "ANY", ['G10_Down'], "HEAD", 2)
(result34, cnt) = asso_rule.template3("1and2", "BODY", "ANY", ['G10_Down'], "HEAD", 2)
(result35, cnt) = asso_rule.template3("2or2", "BODY", 1, "HEAD", 2)
(result36, cnt) = asso_rule.template3("2and2", "BODY", 1, "HEAD", 2)
```

Your program should be able to output the sample query results and the number of rules.

Set Support = 50%, Confidence = 70% for the above queries, and write down the corresponding number of rules in your report.