Project 1: PCA & Apriori Algorithm

Code Submission Due: 10:30 am, Oct. 2

Hard Copy Report: Oct. 2

Demo time: 10:30 am, Oct. 2

General Introduction:

This project contains 2 parts. For the first part, you should implement PCA (Principle Components Analysis) algorithm, project the high-dimensional data to 2 dimensions, and plot the 2-dimensional data points. For the second part, you need to implement Apriori algorithm and rule generation algorithm.

Each team should submit codes and a hard copy report, and give a demo. Demo details will be released two days before the demo date on Piazza. You need to submit the hard copy report during demo, and submit codes to departmental server before 10:30 am, Oct 2.

Part 1: Dimensionality Reduction

Dataset Description:

In this part, you are expected to perform dimensionality reduction on three biomedical data files (*pca_a.txt*, *pca_b.txt*, *pca_c.txt*), which can be found on Piazza.

In each file, each row is the health record of a patient/sample; the last column is the disease name, and the rest columns are features. Note that your code should be able to handle the data with different number of rows/columns.

Requirements:

Please take the following steps:

1. You can use your preferred programming language(s), but you must implement PCA algorithm by yourself. Applying existing package(s) to conduct PCA directly **will not** receive any credit. If you are not sure about whether it is OK to use a certain function, please post your question on Piazza.
2. Implement PCA and then run it on three data files (*pca_a.txt*, *pca_b.txt*, *pca_c.txt*) to get the two-dimensional data points. For each dataset, draw the data points with a scatter plot, and color them according to their disease names.
3. Apply existing packages to run SVD and t-SNE algorithms (Do not need to implement them by yourself) and get the two-dimensional data points. Visualize

the data points of the two algorithms on the three datasets in the same way as PCA.

4. Prepare your submission. Create a folder named *PCA*, in the folder you should include:
    a. Report: A pdf file named as *PCA_report.pdf*. The report should contain:
        i. Nine scatter plots from three datasets and three algorithms. Label them properly by the dataset name and algorithm name in each plot.
        ii. Describe the workflow of your PCA implementation briefly, and discuss your results.
    b. A folder named *Code*, which contains all codes used in this part. Inside the folder, please have a file *README* to describe how to run your code.

Part 2: Association Analysis

Dataset Description:

The dataset is about gene expressions (*association-rule-test-data.txt*) and can be found on Piazza. 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.

Requirements:

1. Implement the Apriori algorithm to find all frequent itemsets. Report the number of frequent item sets for support of 30%, 40%, 50%, 60%, and 70%, respectively. Please see *Template.pdf* for details.
   You **cannot** directly call an existing function or package that implements Apriori. Apriori algorithm should be implemented by yourself. If you are not sure about whether it is OK to use a certain function, please post your question on Piazza.
2. Generate association rules based on the templates, which are listed as follows:
    - Template 1: {RULE|HEAD|BODY} HAS ({ANY|NUMBER|NONE}) OF (ITEM1, ITEM2, ..., ITEMn)
    - Template 2: SizeOf({HEAD|BODY|RULE}) ≥ NUMBER.
    - Template 3: Any combined templates using AND or OR. For example: BODY HAS (1) OF (Disease) AND HEAD HAS (NONE) OF (Disease)

For the above templates, assume we obtain a **RULE** {G1_Up, G3_Down} → {G4_Down, G34_Up}. {G1_Up, G3_Down} is **HEAD** and {G4_Down, G34_Up} is **BODY**.

For this part, if support >= 50% and confidence >= 70% are given, you need to generate **all the rules** satisfying the requirements. In your report, you are asked to just show the number of rules generated. However, in your code, **you need to keep support and confidence variables changeable to do other queries, and you may be asked to show the rules you generate for each query during the demo**. Please see *Template.pdf* for details.

3. Prepare your submission. Make a folder named *Association*, in the folder you should include:
    a. Report: A pdf file named as *Association_report.pdf*. The report should include:
        i. Describe Apriori algorithm with your own words and the workflow on how you mine association rules briefly.
        ii. The answers of the aforementioned queries in requirements 1&2, i.e., the number of generated rules based on the given support and confidence measures.
    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.

Project Submission:

1. Your final submission should be a zip file named as *project1.zip*. In the zip file, you should include aforementioned folder *PCA* and folder *Association*.
2. Log in any CSE department server and submit your zip file as follows:
   >> submit_cse601 project1.zip

Note that copying code/results/report from another group or source is not allowed and may result in an F in the grades of all the team members. Academic integrity policy can be found at https://engineering.buffalo.edu/computer-science-engineering/graduate/resources-for-current-students/academic-integrity.html.