Homework 4

CS 6720

Due 31st March midnight.

1. This question is about getting yourself familiar with frequent subgraph mining tools. You will use the dataset of molecules tested against AIDS . The format is the following:
   #graphID
   # of nodes
   Series of Node Labels
   # of edges
   Series of "Source node, Destination Node, Edge label"

Run gSpan, FSG (also known as PAFI), and Gaston (you should be able to find it using google) against frequency threshold in the aids dataset (you may need to write a script to change the format of the dataset) at minSup=5%, 10%, 25%, 50% and 95%. Plot the running times and explain the trend observed in the running times. Specifically comment on the growth rates and why one technique is faster than the others. You are free to consult the respective papers. (30 points)

Team Assignments:

2. Consider the same AIDS graph dataset. We now provide you a label for each graph: the active molecules against HIV virus and the inactive molecules. Design a technique to classify graphs by using frequent subgraphs as features.
   a. We will evaluate you on another dataset for molecules tested against cancer using the F-score measure. (60 points).

   Your grades will be assigned as follows.

   - Top 10% of all Fscores= 60 points
   - Top 20% of all Fscores= 50 points
   - Top 30% of Fscores = 40 points
   - Top 50% of all Fscores or Fscore > 0.5 = 30 points
   - Top 90% of all F-scores=20 points
   - Fscore > 0=10 points

   For automated testing, you must submit a shell script titled classify.sh. It should support the following operation.

"sh classify.sh <trainset filename containing graphs> <active graph IDs filename> <inactive graph IDs filename> <testset filename containing graphs>"

The output should be a file titled "output.txt" where each line contains either 1 or 0. If test set contains 100 graphs, output.txt should contain 100 lines. 1 indicates active class label and 0 indicates inactive.

Your shell script must complete with 20 minutes on dataset containing up to 10000 graphs. This includes both the training and testing time.

b. In addition, you should also submit a description of your algorithm in the report. (10 points)

3. a. Implement the algorithm to construct the canonical label of a graph using gSpan's smallest DFS code. You should not copy code from any online resource. You will be reported to DISCO if you plagiarize. (60 points).

   The input will be a file containing graphs and output should be a file called "output.txt" containing the canonical labels of each graph. Each line should correspond to a code for a graph. The edge labels should be formatted as follows. The codes should be delimited as follows:
   <edge1><edge2>…<edge n>
   Here each edge is a tuple <I,j,l(i),l(I,j),l(j)>. l(i) is the label of ith vertex.

For automated testing you must submit a shell script of the following format:

"sh generateDFSCode.sh <graphDataset filename>"

You will be graded as follows

- Correct code gives you 40 points
- Top 10% fastest of all submissions: +20
- Top 20% fastest of all submissions: +15
- Top 50% fastest of all submissions: +10
- Top 75% fastest of all submissions: +5

b. In addition, you should also submit a description of your algorithm in the report. (10 points)