Homework 4

COL 870

Due 5th Nov midnight.

TEAM ASSIGNMENT (2 members per team)

1. Consider the same AIDS graph dataset. We now provide you a label for a subset of the graphs: the active molecules against HIV virus and the inactive molecules. Molecules that do not have a class label should be ignored. Design a technique to classify graphs by using frequent subgraphs as features. More specifically, convert each graph into a binary feature vector where each dimension corresponds to the presence or absence of the corresponding subgraph. Ideally, you should not use all frequent subgraphs as features. Rather, you should only use the "discriminative" frequent subgraphs. We will classify the graphs (in the feature space) using the linear kernel of libsvm.

We will evaluate you on another dataset for molecules tested against AIDS 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 two files titled "train.txt" and "test.txt". In train.txt, the i_{th} line contains the class label of graph i followed by the feature vector representation of the i_{th} graph in the trainset. The label of an active graph is "1" and an inactive graph is "-1". Each line must be of the following format:

```
<label> <index1>:<value1> <index2>:<value2> ...
.
.
.
```

The test.txt file should of the same format as above, with the only exception being that you do not need to include the class label. That is, it should be of the form

```
<index1>:<value1> <index2>:<value2> ...
.
.
.
```

If test set contains 100 graphs, test.txt should contain 100 lines (same for train.txt as well). We will run libsvm on your files. **If you do not adhere to the above format, you will receive 0 points. More specifically, you should run libsvm on train.txt and ensure that libsvm is able to train on your data format.**

Your shell script must complete within 20 minutes. This includes both the training and testing time.

You may need to perform subgraph isomorphism for this task. To help you, a Java library with an example code is shared. To see an example of how subgraph isomorphism is performed, check the file toolbox/subgraphFrequencyCounter.java