CS 412: Spring'21
Introduction To Data Mining

Assignment 3

(Due Monday, April 12, 11:59 pm)

General Instructions

- Feel free to talk to other students of the class while doing the homework. We are more concerned that you learn how to solve the problems than that you demonstrate that you solved it entirely on your own. You should, however, write down your solution yourself. Please try to keep the solution brief and clear.
- Please use Slack first if you have questions about the homework. You can also use CampusWire, send us e-mails, and/or come to our office hours. If you are sending us emails with questions on the homework, please cc all of us (Arindam, Carl, Jialong, and Qi) for faster response.
- The homework is due at 11:59 pm on the due date. We will be using both compass and **Gradescope** (details in the next section) for collecting the homework assignments. Please submit your code (net_id.py) via Compass (`http://compass2g.illinois.edu`). Please do NOT email a hard copy of your write-up. Contact the TAs if you are having technical difficulties in submitting the assignment. We will NOT accept late submissions!
- All programming will be in python 3.

Programming Assignment Instructions

- The homework will be graded using Gradescope. You should be automatically added to this course. You will be able to submit your code as many times as you want. There will be two sequence databases: validation and test. Each will contribute half of the possible points on this problem. The validation questions will provide feedback but the test questions will not.

Assignment. The focus of the programming assignment is to develop code for a frequent sequential pattern mining algorithm. Given an input sequence database S and a minimum support threshold (minsup), the algorithm should return the frequent sequential patterns. The sequential pattern is simplified as a **successive sequential pattern**. For example, given a sequence $\langle abc \rangle$, ab is regarded as a sequential pattern while ac is not.

The emphasis will be on correctness of the implementation, not computational efficiency. In particular, you can use any of the algorithms discussed in class for sequential pattern mining. Further, since we will be testing the code on relatively small sequence databases S, you can assume all operations will be doable in main memory, and can use suitably data-structures to hold intermediate results. In particular, you do not have to create physical projected databases, which is needed when intermediate results cannot be held in main memory.

You will not get credit if your code does not work.

Input Format. The input will be a plain text file with a sequence sequence database, with each line corresponding to a sequence. Each line will have a sequence id followed by the sequence. For example, for the sequence database given below:

Sequence_ID	Sequence
S_1	$\langle aabcacdcf \rangle$
S_2	$\langle adcbcae \rangle$
S_3	$\langle efabdfcb \rangle$
S_4	$\langle egafcbc \rangle$

the validation input text file will be:

```
s1, <aabcacdcf>
s2, <adcbcae>
s3, <efabdfcb>
s4, <egafcbc>
```

Your code will take two inputs: a plain text file, the sequence database, and an integer, the minimum support.

Output Format. Your code will implement a function called get_sequences. It will return a dictionary which will have keys as frequent patterns and values as the support of those patterns. The dictionary may look like,

$\{a : 4, ab : 2, ...\}$

What you have to submit. Your code homework3.py should be in Python 3 with a function get_sequences which takes two inputs:

1. Sequence database: A plain text file with the sequence database as shown in the example above. Each line will have a sequence Id and the sequence, and these two will be comma separated.
2. Minimum support: An integer indicating the minimum support for the freqeunt pattern

mining.

A call to the function would be like:
`get_sequences("seqDB.txt", 3)`

The output, i.e., frequent sequential patterns along with their support as shown in the example above, should be a dictionary.

Guidelines. The assignment needs you to both understand algorithms for sequential pattern mining (pick one you want to implement) as well as being able to implement the algorithm in python. Here are some guidelines to consider for the homework:

- You are getting more than 3 weeks for the assignment, and many of you may need this time. If you are planning to start a week before the deadline, it is less likely you will be able to do a satisfactory job.
- It is good idea to make early progress on the assignment, so you can assess how much it will take for you. (1) Start working on the assignment as soon as it is posted. Within the first week, you should have a sense of the parts which will be easier, and parts which will need extra effort from you; (2) Solve the example by hand as a warm-up to (a) decide the algorithm you will use, and (b) be comfortable with the steps which you will have to code.

While you will be writing your own code, there are existing code based which implement variants of these algorithms. We cannot vouch for their correctness, but some aspects of such code can help you in developing your own. Some sample code
`https://pypi.org/project/prefixspan/`
`https://github.com/chuanconggao/PrefixSpan-py/commit/441b04eca2174b3c92f6b6b2f50a30f1ffe4968c`