Project 1

ESE 545, Data Mining: Learning from Massive Datasets

September 7, 2018

Due at 11:59PM on **October 1, 2018**

These related problems consist of several parts. You are required to solve them using Python, Matlab or C and turn in your code. You are allowed to work in groups of at most two members. You should submit your code with a brief report containing responses to each part. Turn in one report and script per group. Upload a zipped file containing the report and the code on Canvas.

Problem 1. Download the Netflix Dataset from Canvas. This file contains ratings of movies by real Netflix users. The data is organized by `MovieID` and then `UserID,Rating,Date`, in the form

```
MovieID:
UserID,Rating,Date
UserID,Rating,Date
⋮
```

Your first task is to make a function which will read in this data from a TXT file to a $M \times N$ matrix, where M is the number of movies and N is the number of users (who rated less than 20 movies), which should look like

$$M \text{ movies} \left\{ \overbrace{\begin{bmatrix} 0 & 1 & 0 & 1 & \cdots & 1 \\ 1 & 1 & 0 & 0 & \cdots & 0 \\ & \vdots & & & \ddots & \\ 1 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}}^{N \text{ users}} \right.$$

However, we only care about movies that users liked, so you should only save a movie for a user (put a 1 in the spot `(MovieID,UserID)`) when the rating is 3 or higher. To reduce the computational burden, you should only keep users that have rated *less than* 20 movies. We will not use the `Date`. Do not assume that `MovieIDs` or `UserIDs` are sequential, or that none are missing. **10 points**

Problem 2. To get a sense for the data, we will analyze a random sample. Pick 10,000 pairs at random and compute the average Jaccard distance of the pairs, as well as the lowest distance among them. Plot a histogram of the pairwise Jaccard distances, and include all results in your report. **10 points**

Hint: The Jaccard distance between sets A and B is

$$d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}.$$

Think about how you could interpret a user's liked movies as a set based on the matrix you made.

Problem 3. Our final goal will be to find the approximate nearest neighbor of a queried user. However, the current matrix representation is not the most efficient way to store the data for this purpose. Find a more efficient way to store the data that you may use for the later Problems. **10 points**
Hint: think of the data structures introduced in class, and how they might allow you to quickly find candidates for the nearest neighbor of a user.

Problem 4. Using your data structure from Problem 3, detect all pairs of users that are *close* to one another. We define two users as *close* if their Jaccard distance is below 0.35. You should do this as efficiently as possible, both in terms of storage and time complexity, using the entire dataset. If your solution requires you to pick any parameters, you should justify your choice with plots or data in your report, whichever is appropriate. **45 points**

Problem 5. Create a function that accepts a queried user and returns their approximate nearest neighbor. This should be done as efficiently as possible. **25 points**