Project 2 - 25 points

In this project we will be working with the movielens dataset to implement a few algorithms. In the following you are expected to implement everything **from scratch** in the programming language of your choice. In other words, you cannot use libraries that perform the requested functionality (i.e., you have to really implement k-means instead of using kmeans from scikit-learn for example).

You can download the dataset from here: https://grouplens.org/datasets/movielens/25m/

1. K-Means (12 points)

Implement a disk-based k-means flavor (whichever you prefer from the ones discussed in the clustering lecture from the youtube link you have in your email already). Please note that it is important for your implementation to be disk based, i.e., you cannot load the whole dataset in memory and process it from there. You can however keep the clusters and their metadata (i.e. centroid/clustroid and membership) in memory. The format of the files that you will prepare is up to you, but please include the code of preparing the necessary input files in your zip. You should not include neither the generated nor the original data from the grouplens web site.

Implement K-means clustering program for the movies. Your program should take as input:
- the number of clusters k
- the distance function to use (see below)
- any input file(s) that it needs

Its output should be a list of movies and the cluster they belong to. It should also print the centroids/clustroids.

Implement the following distance functions for the movies:
a) d1: jaccard similarity based on the genres of the movies
b) d2: jaccard similarity based on the tags of the movies
c) d3: cosine similarity based on the ratings of the movies
d) d4 = 0.3*d1 + 0.25*d2 + 0.45*d3

2. Collaborative Filtering (13 points)

Implement a disk-based program that does item-based and user-based collaborative filtering as discussed in the recommender systems lecture from the youtube link that you have in your email already. Please note that it is important for your implementation to be disk based, i.e., you cannot load the whole dataset in memory and process it from there. You can use secondary data structures (e.g. indexes) if you like on disk. If you do so, please include the code and instructions on how to generate those data structures on disk. You should not include neither the generated nor the original data from the grouplens web site.

Your program should take as input a parameter telling it whether we are interested in user-based, item-based or a combination of the two as well as any input files that it needs. Then it should wait for a user id as input from the keyboard and it should present the top-20 movie recommendations for that user together with the similarities that you have computed.

- If user-based was selected, then the selection is done performing user-user similarities using the ratings

- If item-based was selected, then the selection is done performing item-item similarities using the ratings
- If combination was selected, then we compute the top-20 from user-based, the top-20 from item-based, we merge the lists using the two different rankings, i.e. from user-based and from item-based, and compute the top-20 in this way.

Deliverables

- All files for your project need to be zipped in **one single zip file.** The file name should be in the following format: Lastname_Firstname_Project_X.zip (change X with the respective project number i.e. 1 for project 1, 2 for the second project and so on).
- Please email the zip file to the instructor by the deadline. **There are absolutely no exceptions and no extensions.**
- Inside the zip file, please include a) your source code b) any results that are part of the project c) a README text file that explains how to compile (if needed) and run your code and d) a short and concise report (preferably pdf - no more than 10 pages) describing your work and your approach to solving the project. **Please do not include the original or any derived data in the zip file.**

Important Notes

- Please check the class Web site: http://www.di.uoa.gr/~antoulas/m111 regularly for announcements and/or clarifications to the project. Announcements will show up there and will also be sent to the mailing list.
- You are free to make any assumptions you may need to along the way as long as you document them in your report.
- The project is meant to be worked on by the student submitting and only that student. In the event that there is a submission not worked on by the student, then the student fails the class.
- Although students are expected (and encouraged) to chat among them on potential solutions and approaches, sharing code and solutions is strictly not allowed. In the event that two or more students provide submissions that have common pieces, everyone involved (regardless of who is at fault or who copied from whom) fails the class.