CptS 315: Introduction to Data Mining
Homework 2
(Due date: Mar 4, midnight PST)

Instructions

- Please use a word processing software (e.g., Microsoft word) to write your answers and submit a printed copy to me at the beginning of the class on Feb 8. The rationale is that it is sometimes hard to read and understand the hand-written answers.
- All homeworks should be done individually.

Analytical Part (40 points)

Q1. Consider the following ratings matrix with three users and six items. Ratings are on a 1-5 star scale. Compute the following from data of this matrix: (20 points)

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	4	5		5	1	
User 2		3	4	3	1	2
User 3	2		1	3		4

Table 1: Data of ratings from three users for six items.

a) Treat missing values as 0. Compute the Jaccard similarity between each pair of users.
Jaccard Similarity [User 1, User 2] =3/6 = 1/2
Jaccard Similarity [User 1, User 3]= 2/6 = 1/3
Jaccard Similarity [User 2, User 3]=3/6=1/2

b) Treat missing values as 0. Compute the cosine similarity between each pair of users.
Cosine Similarity [User 1, User 2] $= \frac{4*0+5*3+0*4+5*3+1*1+0*2}{sqrt(15+25+25+1)*sqrt(9+16+9+1+4)} = 0.6110$
Cosine Similarity [User 1, User 3]$= \frac{4*2+5*0+0*1+5*3+1*0+0*4}{sqrt(15+25+25+1)*sqrt(4+1+9+16)} = 0.5169$
Cosine Similarity [User 2, User 3]$= \frac{2*0+0*3+1*4+3*3+0*1+4*2}{sqrt(9+16+9+1+4)*sqrt(4+1+9+16)} = 0.6139$

c) Normalize the matrix by subtracting from each non-zero rating, the average value for its user. Show the normalized matrix.

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User 1	0.25	1.25		1.25	-2.75	
User 2		0.4	1.4	0.4	-1.6	-0.6
User 3	-0.5		-1.5	0.5		1.5

d) Compute the (centered) cosine similarity between each pair of users using the above normalized matrix.

Cosine Similarity [User 1, User 2] $= \frac{0.25*0+1.25*0.4+0*1.4+1.25*0.4-2.75*-1.6+0*-0.6}{sqrt(10.75)*sqrt(5.2)} = 0.7223$

Cosine Similarity [User 1, User 3]$= \frac{0.25*-0.5+1.25*0+0*-1.5+1.25*0.5-2.75*0+0*1.5}{sqrt(10.75)*sqrt(5)} = 0.0682$

Cosine Similarity [User 2, User 3]$= \frac{-0.5*0+0*0.4-1.5*1.4+0.5*0.4+0*-1.6+1.5*-0.6}{sqrt(5.2)*sqrt(5)} = -0.5491$

Q2. Please read the following two papers and write a brief summary of the main points in at most TWO pages. (20 points)

Brent Smith, Greg Linden: Two Decades of Recommender Systems at Amazon.com. IEEE Internet Computing 21(3): 12-18 (2017)
https://www.computer.org/csdl/mags/ic/2017/03/mic2017030012.pdf

Brent Smith and Greg Linden's paper "Two Decades of Recommender Systems at Amazon.com" opens with Amazon's first launch of its item-based collaborative filtering in 1998 and which has since been incorporated in many companies' recommendation systems. Microsoft research found that 30 precent of Amazon's page view were from recommendations. Companies such as YouTube and Netflix also utilize item-based collaborative filtering. A challenge of using item-based collaborative filtering is defining relation of items, in general it can be assumed that the number of customers who bought XY and the purchase of X are the probability same and Y probability is the estimate of those who bought XY. The challenge comes from the fact that for any pair of items there is a certain number of customers who bought X and will likely buy Y than everyone else. The erraticness of customer purchase history means all purchases of X must be accounted to estimate when Y will be bought. Of course, there are other ways to measure

relatedness and not one works best in every scenario. Machine Learning can help pick parameter for specific recommendations such as compatibility and timing of purchase, and timing of interest, and searches. The paper ends with the greatness of opportunity still left in the optimization and creation of recommendations systems, including item-based collaboration.

Greg Linden, Brent Smith, Jeremy York: Industry Report: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Distributed Systems Online 4(1) (2003) https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf

Greg Linden, Brent Smith, and Jeremy York's Industry Report: Amazon.com Recommendations: Item-to-Item Collaborative Filtering discuss e-commerce recommendation algorithms and the challenges that come such as data volume, the short time for data output, new customer cold starts, and immediate reaction to new information. The majority of algorithms find overlapping customers and suggest unpurchased and highly rated items. Traditional collaborative filtering uses vectors of items and users to find similar users and recommend items which is O(M+N), because there are scaling issues samples are taken to get around any limitations of data capability. Cluster models are focused on sorting customers together in groups of most similarity, these have much better scalability. Another method is search-based methods which find items user has purchased and rated to find items of the same actress, company, author, etc. Amazon utilizes item-based collaborative filtering matching user's purchases and rating to similar items, then combines those similar items into a recommendation list. Because the computation can be very extensive at is about O(NM) with worst case at $O(N^2M)$. Amazon is able to scale this algorithm quite well to its 29 million customers, and it is due to it's expensive similar-items table offline.

Programming and Experimental Part (60 points)

Movie Recommendations via Item-Item Collaborative Filtering. You are provided with real-data (Movie-Lens dataset) of user ratings for different movies. There is a *readme* file that describes the data format. In this project, you will implement the *item-item collaborative filtering* algorithm that we discussed in the class. The high-level steps are as follows:

a) Construct the profile of each item (i.e., movie). At the minimum, you should use the ratings given by each user for a given item (i.e., movie). Optionally, you can use other information (e.g., genre information for each movie and tag information given by user for each

movie) creatively. If you use this additional information, you should explain your methodology in the submitted report.

b) Compute similarity score for all item-item (i.e., movie-movie) pairs. You will employ the *centered cosine* similarity metric that we discussed in class.

c) Compute the neighborhood set *N* for each item (i.e. movie). You will select the movies that have highest similarity score for the given movie. Please employ a neigborhood of size 5. Break ties using lexicographic ordering over movie-ids.

d) Estimate the ratings of other users who didn't rate this item (i.e., movie) using the neighborhood set. Repeat for each item (i.e., movie).

e) Compute the recommended items (movies) for each user. Pick the top-5 movies with highest estimated ratings. Break ties using lexicographic ordering over movie-ids.

Your program should output top-5 recommendations for each user.

Instructions for Code Submission and Output Format.

Please follow the below instructions. It will help us in grading your programming part of the homework.

- Supported programming languages: Python, Java, C++
- Store all the relevant files in a folder and submit the corresponding zipfile named after your student-id, e.g., 114513209.zip
- This folder should have a script file named run_code.sh

  Executing this script should do all the necessary steps required for executing the code including compiling, linking, and execution
- Assume relative file paths in your code. Some examples:

  ''./filename.txt'' or ''../hw2/filename.txt''
- The output of your program should be dumped in a file named "output.txt" in the following format. One line for each user.

  User-id1 movie-id1 movie-id2 movie-id3 movie-id4 movie-id5
  User-id2 movie-id1 movie-id2 movie-id3 movie-id4 movie-id5
  …
  …

  Explanation.

  - Line 1 should have the first user-id followed by the movie-ids of recommended movies.
  - Line 2 should have the second user-id followed by the movie-ids of recommended movies.

- Make sure the output.txt file is dumped when you execute the script `run_code.sh`

- Zip the entire folder and submit it as

  `<student_id>.zip`