CptS 315: Introduction to Data Mining Homework 2 (HW2)

Instructions

- Please use a word processing software (e.g. Microsoft word) to write your answers.
- You will need to submit your answers as a pdf file on Blackboard.
- This assignment is due by the date stated on Blackboard.
- All homeworks should be done individually.

Q1. (50 points) 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:

	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(User 1,User 2) = item2, item4, item5 / item1, item 2, item 3, item 4, item5, item6
= 3/6 **= 0.5**

Jaccard(User 1, User 3) = item1, item4 / item1, item2, item3, item4, item5, item6
= 2 / 6 = **0.33**

Jaccard(User 2, User 3) = item3, item4, item6 / item1, item2, item3, item4, item5, item6
= 3 / 6 = **0.5**

b) Treat missing values as 0. Compute the cosine similarity between each pair of users.

Cosine(User 1, User 2) = 3*5 + 3*5 + 1*1 / sqrt(4^2+5^2+5^2+1^2)*sqrt(3^2+4^2+3^2+1^2+2^2)
=31/sqrt(67)*sqrt(39) = 31/(8.18 * 6.24) = 31/51
= **0.606**

C(User 1, User 3) = (4*2 + 5*3) / sqrt(4^2+5^2+5^2+1^2)*sqrt(2^2+1^2+3^2+4^2) = 23/ sqrt(67) * sqrt(30)
= **0.513**

C(User 2, User 3) = (4*1+3*3+2*4) / sqrt(2^2+1^2+3^2+4^2) * sqrt(3^2+4^2+3^2+1^2+2^2) = 21 / sqrt(39) * sqrt(30)
= **0.614**

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

Avg(User1) = (4+5+5+1)/4= 3.75

Avg(User2) = (3+4+3+1+2)/5 = 2.6

Avg(User3) = (2+1+3+4)/4 = 2.5

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

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

Cosine(User 1, User 2) = 1.25*0.6 + 1.25*0.6 + -2.75*-1.6 / sqrt(0.25^2+1.25^2+1.25^2+-2.75^2)*sqrt(0.6^2+1.4^2+0.6^2+ -1.6^2+ -0.4^2) =0.75+0.75+4.4/sqrt(10.75)*sqrt(5.4) = 5.9/(8.18 * 6.24)
= **0.774**

C(User 1, User 3) = (.25*-0.5+ 1.25*-0.5) / sqrt(0.25^2+1.25^2+1.25^2+-2.75^2) * sqrt(-0.5^2 + -1.5^2 + -0.5^2 + 1.5^2) = -0.125 + -0.625/ sqrt(10.75) * sqrt(5)
= **-0.10**

C(User 2, User 3) = (1.4*1.5+0.6*-0.5+ -0.4*1.5) / sqrt(0.6^2+1.4^2+0.6^2+ -1.6^2+ -0.4^2) * sqrt(-0.5^2 + -1.5^2 + -0.5^2 + 1.5^2) = 1.2 / sqrt(5.4) * sqrt(5)
= **0.231**

Q2. (50 points) Please read the following two papers and write a brief summary of the main points in at most TWO pages.

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

Amazon was the very first company to implement an item based collaborative filtering system for millions of people at once. It was first implemented in 1998 and following its initial implementation has been the basis for many other websites that followed in its footsteps with similar algorithms. The main benefit of this recommendation system has been its simplicity, scalability, and ability to provide new and useful recommendations to Amazon's users. The basic idea behind the recommendation system is to find items that are often bought by a single person and build a related items table that can be used to lookup recommendations for users depending on what they have already bought.

As the more years pass since the original recommendation algorithm was implemented at Amazon some statistics have shown that up to 30% of Amazon's pageviews come from recommendations based off the relatable item table. Additionally, recommendations algorithms have found great use at Netflix who has reported that their recommendation systems accounts for up to 80% of their pageviews with the system being worth more than $1Billion per year. To build up these recommendation tables that the system is built on top of two products must come up with similarity or relatability values between each other. This is done by using the formula P(Y) = |Y buyers|/|all buyers|.

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

With the rise of Recommendation Algorithms there has also been an increase in the number of challenges that need to be solved to produce a successful implementation. For companies like amazon many of their problems stem from the following issues "How to generate recommendations in real time", "How to apply the system to a dataset with millions of datapoints", and "The age of user's account". The most basic approach to these problems is to look at the pool of users whose purchased and rated items overlap with the user the recommendation is being generated for. Using that information, the algorithm can find new items the user is most likely to enjoy and buy. This method of user comparison is usually called collaborative filtering or cluster models. A closer look at collaborative filtering reveals that it uses cosine similarity to find users who are very similar to base the recommendations off. This leads to several downsides though such as high computation time or low-quality recommendations with small data sets. Due to these issues companies like Amazon have had to find ways around the limitations of the above recommendation systems.

Ultimately when one recommendation algorithm cannot be used without encountering some of the issues mentioned above, companies with large datasets such as Amazon have found that multiple algorithms should be implemented and used. In that case

a combination of User to User recommendation, Cluster Models, Item to Item recommendation and Search based systems are used. With this combination companies have found that the best recommendation algorithms are ones that provide customers or users with a personalized experience, scale with large datasets, have low processing times, and quick reactions to users preferences.