CS5304 Assignment 2: Recommender Systems

Wenyi Chu

April 15, 2020

Summary

In this project, we are implementing a system to recommend TV shows for users using different types of recommender systems. We used a dataset that includes 9985 users and 563 popular TV shows, for which a given user watched a given show over a 3 month period.

1 user-user recommender system

In order to implement user recommender system, we first compute the following for all Tv shows

$$r_{Alex,t} = \sum_{x \in users} cos - sim(x, Alex) * R_{xt}$$

where **cos - sim(x,Alex)** is the cosine similarity of other users with Alex. Therefore, those tv-shows that have higher similarity scores r_{xt} will be the shows that are rated high by the users similar to Alex.

Question From all the TV shows in S (first 100 shows), five shows that have the highest similarity scores for Alex are shown in the following table.

	Movies	**Similarity scores**
1	"FOX 28 News at 10pm"	908.48005348
2	"Family Guy"	861.1759992
3	"2009 NCAA Basketball Tournament"	827.60129547
4	"NBC 4 at Eleven"	784.7819589
5	"Two and a Half Men"	757.6011181

Table 1: User-user recommender system

2 item-item recommender system

In this section, we computed the following

$$r_{Alex,t} = \sum_{x \in items} R_{Alex,x} * cos - sim(x,t)$$

for all tv-shows, where **R** is the ratings matrix and **cos - sim(x,t)** is the cosine-similarity of each pair of TV shows.

Question From all the TV shows in S, five shows that have the highest similarity scores for Alex are shown in the following table:

	Movies	**Similarity scores**
1	”FOX 28 News at 10pm”	31.36470168
2	”Family Guy”	30.0011418
3	”NBC 4 at Eleven”	29.39679777
4	”2009 NCAA Basketball Tournament”	29.22700156
5	”Access Hollywood”	28.97127767

Table 2: item-item recommender system

3 Latent hidden model recommender system

In this section, we performed a singular value decomposition (SVD) that factors the **user ratings matrix R** into three matrices as follows:

$$R = U \sum V^T$$

where **U** is the *user* features matrix, and **V** is the *movie* features matrix. To get the lower rank approximation, we keep only the top k (k = 10) features as the k most important underlying taste and preference vectors. Therefore, the top 5 TV-shows for Alex in S are shown in the table below.

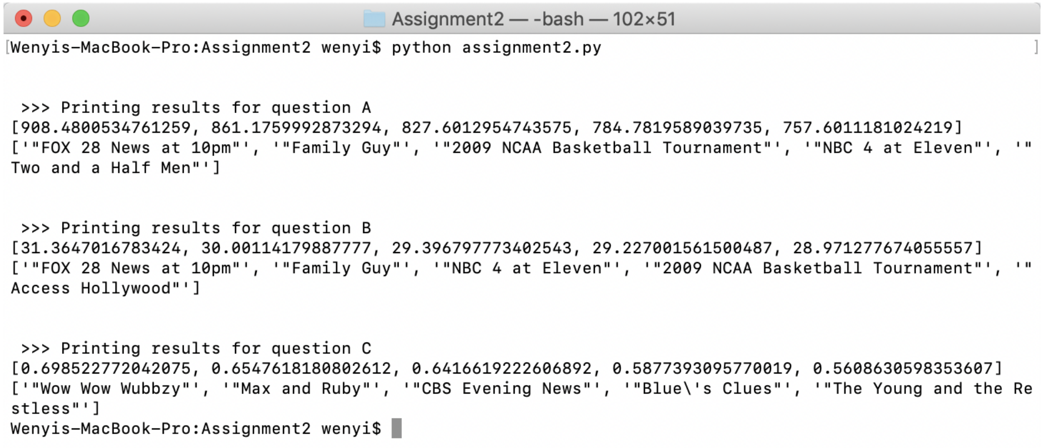
	Movies	**Similarity scores**
1	”Wow Wow Wubbzy”	0.698522772042075
2	”Max and Ruby”	0.6547618180802612
3	”CBS Evening News”	0.6416619222606892
4	”Blueś Clues”	0.5877393095770019
5	”The Young and the Restless”	0.5608630598353607

Table 3: Latent hidden model results

4 README and misc

Relevant code was included in the ***./assignment2.py*** file, and data was loaded from the ***./data*** folder. To run the code, simply put the command: *python assignment2.py*

Runtime output shown in the Figure 1.



```
Wenyis-MacBook-Pro:Assignment2 wenyi$ python assignment2.py


 >>> Printing results for question A
[908.4800534761259, 861.1759992873294, 827.6012954743575, 784.7819589039735, 757.601181024219]
['"FOX 28 News at 10pm"', '"Family Guy"', '"2009 NCAA Basketball Tournament"', '"NBC 4 at Eleven"', '"
Two and a Half Men"']


 >>> Printing results for question B
[31.3647016783424, 30.00114179887777, 29.396797773402543, 29.227001561500487, 28.971277674055557]
['"FOX 28 News at 10pm"', '"Family Guy"', '"NBC 4 at Eleven"', '"2009 NCAA Basketball Tournament"', '"
Access Hollywood"']


 >>> Printing results for question C
[0.698522772042075, 0.6547618180802612, 0.6416619222606892, 0.5877393095770019, 0.5608630598353607]
['"Wow Wow Wubbzy"', '"Max and Ruby"', '"CBS Evening News"', '"Blue\'s Clues"', '"The Young and the Re
stless"']
Wenyis-MacBook-Pro:Assignment2 wenyi$ 
```

Figure 1: Output