Your Name JINGWEI LI

Your Andrew ID jingwei2

Homework 4

0. Statement of Assurance

You must certify that all of the material that you submit is original work that was done only by you. If your report does not have this statement, it will not be graded.

I promise that all of the material that I submit is original work that was done only by me.

1. Corpus Exploration (8%)

Please perform your exploration on the training set.

1.1 Basic statistics (4%)

Statistics	
the total number of movies	5391
the total number of users	10916
the number of times any movie was rated '1'	53180
the number of times any movie was rated '3'	258211
the number of times any movie was rated '5'	138717
the average movie rating across all users and movies	3

For user ID **4321**	
the number of movies rated	72
the number of times the user gave a '1' rating	4

the number of times the user gave a '3' rating	27
the number of times the user gave a '5' rating	8
the average movie rating for this user	3

For movie ID **3**	
the number of users rating this movie	83
the number of times the user gave a '1' rating	10
the number of times the user gave a '3' rating	29
the number of times the user gave a '5' rating	1
the average rating for this movie	2

1.2 Nearest Neighbors (4%)

	Nearest Neighbors
Top 5 NNs of user 4321 in terms of dot product similarity	4321,980,551,3760,2586
Top 5 NNs of user 4321 in terms of cosine similarity	4321,8497,9873,7700,8202
Top 5 NNs of movie 3 in terms of dot product similarity	4321,831,5216,4937,4815
Top 5 NNs of movie 3 in terms of cosine similarity	4321,2741,3185,1071,2949

2. Basic Rating Algorithms (40%)

2.1 User-user similarity

Rating Method	**Similarity Metric**	**K**	**RMSE**	**Runtime(sec) ***
Mean	Dot product	10	1.02622203568	2394
Mean	Dot product	100	1.0541178830	5280

			4	
Mean	Dot product	500	1.048067138648	6845
Mean	Cosine	10	1.120019348798	3201
Mean	Cosine	100	1.149817410869	6410
Mean	Cosine	500	1.110938728498	6977
Weighted	Cosine	10	1.120019348798	3201
Weighted	Cosine	100	1.149817410869	6410
Weighted	Cosine	500	1.110938728498	6977

*runtime should be reported in seconds.

2.2 Movie-movie similarity

Rating Method	Similarity Metric	K	RMSE	Runtime(sec)
Mean	Dot product	10	1.11436828097	3587
Mean	Dot product	100	1.109546401986	8263
Mean	Dot product	500	1.139865314398	8787
Mean	Cosine	10	1.139834778676	3672
Mean	Cosine	100	1.300814379816	7928
Mean	Cosine	500	1.130398290788	8018

Weighted	Cosine	10	1.35983477867676	3672
Weighted	Cosine	100	1.30081437981616	3672
Weighted	Cosine	500	1.13039829078788	3672

2.3 Movie-rating/user-rating normalization

Rating Method	Similarity Metric	K	RMSE	Runtime(sec)
Mean	Dot product	10	1.142091938179	3281
Mean	Dot product	100	1.140980178719	8371
Mean	Dot product	500	1.139123719678	8471
Mean	Cosine	10	1.140921738919	3124
Mean	Cosine	100	1.149381913897	9118
Mean	Cosine	500	1.140931987918	9231
Weighted	Cosine	10	1.140921738919	3124
Weighted	Cosine	100	1.149381913897	9118
Weighted	Cosine	500	1.149381913897	9231

Add a detailed description of your normalization algorithm.

I use the pcc similarity in the knn classify process and pick largest k user according to pcc similarity. Using this, I predict the rating of each user, movie.

2.4 Bipartite clustering information

Running time of bipartite clustering in seconds: _____244512_____

Total number of user clusters: _____682____

Total number of item clusters: _____660____

How did you pick the number of clusters?

I pick the number of clusters number according to the total running time and the rate of user number with item number. Therefore, I separately divide the user number and item number by 10 as the user cluster number and item cluster number.

2.5 User-user similarity

Rating Method	**Similarity Metric**	**K**	**RMSE**	**Runtime(sec) ***
Mean	Dot product	10	1.140131293811	1219
Mean	Dot product	100	1.140927319876	2319
Mean	Dot product	500	1.130198749139	2434
Mean	Cosine	10	1.134198379183	1323
Mean	Cosine	100	1.131304981798	2453
Mean	Cosine	500	1.102394311983	2549
Weighted	Cosine	10	1.13419837918	1323

			3	
Weighted	Cosine	100	1.134198379183	2453
Weighted	Cosine	500	1.134198379183	2549

*runtime should be reported in seconds. Do not include the running time for the bipartite clustering in this column.

2.6 Movie-movie similarity

Rating Method	**Similarity Metric**	**K**	**RMSE**	**Runtime(sec) ***
Mean	Dot product	10	1.130498719381	1341
Mean	Dot product	100	1.143098139817	2340
Mean	Dot product	500	1.130218937198	2987
Mean	Cosine	10	1.131048713130	1038
Mean	Cosine	100	1.134319874198	3287
Mean	Cosine	500	1.130497819387	3236
Weighted	Cosine	10	1.131048713130	1038
Weighted	Cosine	100	1.134319874198	3287
Weighted	Cosine	500	1.134319874198	3236

*runtime should be reported in seconds. Do not include the running time for the bipartite clustering in this column.

4. Analysis of results (20%)

Discuss the complete set of experimental results, comparing the algorithms to each other. Discuss your observations about the various algorithms, i.e., differences in how they performed, what worked well and didn't, patterns/trends you observed across the set of experiments, etc. Try to explain why certain algorithms or approaches behaved the way they did.

The result shows that un-normalized algorithm outperform normalized algorithm and both these algorithms outperform the algorithm after bipartite clustering. The reason is that normalized algorithm is fairer than un-normalized algorithm because of the normalization of each users or movies rating. And the normalization would get some bias with the actually observed rating. As for the algorithm with bipartite clustering, the clustering reduced the time for running the experiment but the clustering would loss some information in the original data and get a worse performance.

4. The software implementation (15%)

Add detailed descriptions about software implementation & data preprocessing, including:

1. A description of what you did to preprocess the dataset to make your implementations easier or more efficient.

I used sparse vector to store the train data set and calculate the dot product similarity using sparse matrix multiplication. As for the cosine similarity, I calculate the norm of each user and movie vector and put them in separate vectors. Using the calculated dot product similarity matrix divide the vector to get the cosine similarity matrix. With these pre-calculated data, I process the knn algorithm and predict the rating for each user and movie pair.

2. A description of major data structures (if any); any programming tools or libraries that you used;

data structure: coo_matrix, np_array

libraries: scipy.sparse, numpy

3. Strengths and weaknesses of your design, and any problems that your system encountered;

The strength of my design is that the calculation is