Your Name : Yuzhu Zhang

Your Andrew ID: yuzhuz

Homework 6

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.

Did you receive any help whatsoever from anyone in solving this assignment? Yes/ No. NO
Did you give any help whatsoever to anyone in solving this assignment? No
Did you consult any outside sources in completing this assignment? No

1. Training Set Construction (5 pts)

Construct the training set for LETOR as instructed and report the following statistics.

Statistics	
the total number of observed ratings in R	820367
the total number of training examples in T	3520432
the ratio of positive examples to negative examples in T	1：1
the number of training examples in T for user ID **1234**	196
the number of training examples in T for user ID **4321**	64

2. NDCG of Memory-based methods (10 pts)

2.1 User-user similarity (Do the same imputation first as in HW4)

Rating Method	**Similarity Metric**	**K**	**NDCG@20**	**Runtime(sec)***
Mean	Dot product	10	0.904370	28
Mean	Dot product	100	0.943115	31
Mean	Dot product	500	0.942084	61
Mean	Cosine	10	0.931614	23
Mean	Cosine	100	0.942304	25
Mean	Cosine	500	0.942802	39

*runtime should be reported in seconds.

2.2 Movie-movie similarity (Do the same imputation first as in HW4)

Rating Method	Similarity Metric	K	NDCG@20	Runtime(sec)
Mean	Dot product	10	0.934732	8
Mean	Dot product	100	0.941895	15
Mean	Dot product	500	0.940939	33
Mean	Cosine	10	0.938108	8
Mean	Cosine	100	0.945174	8
Mean	Cosine	500	0.942185	20

3. NDCG of PMF (6 pts)

Report the best results you can get on dev set from training PMF with different dimensions of latent factors.

Num of Latent Factors	NDCG@10	NDCG@20	NDCG@30	Runtime(sec)*
10	0.893326(0.938078)	0.901801(0.942448)	0.902821(0.942911)	656
20	0.916115	0.923118	0.923911	904
50	0.909735	0.917271	0.918151	1522
100	0.894474	0.902997	0.904042	2217

4. RankSVM (12 pts)

a. After the optimum w is learned by SVM on the training set T, given a testing user, how to generate the ranked list of unrated items and why? (Please be concise and clear; use formulas to express your idea when possible) (2 pts)

The score of a unranked movie i for user u can be obtained by multiplying the SVM w with the data that is generated from the PMF factorized matrix. That is S_{u,i} = w * [U_{u}.*V_{i}] where U and V are factorized user and movie matrix.

Then for a given user, the ranked list can be obtained by sorting all the score of the unrated movie for that user.

b. Report the best results you can get on dev set from training RankSVM with features from PMF (10 pts)

Num of Latent Factors from PMF	NDCG@10	NDCG@20	NDCG@30	Runtime(sec)*

10	0.935318	0.940148	0.940652	620.707979918
20	0.934385	0.939255	0.939722	867.338557959
50	0.930735	0.935601	0.936102	2168.71926904
100	0.921470	0.927190	0.927717	3307.12908006

*report the time of training SVM after PMF outputs the features. For each case, you might want to tune the regularization parameters of both PMF and SVM.

5. LR-LETOR (12 pts)

a. After the optimum w is learned by LR-LETOR on the training set T, given a testing user, how to generate the ranked list of unrated items and why? (Please be concise and clear; use formulas to express your idea when possible) (2 pts)

Similar for SVM case. The score of a unranked movie i for user u can be obtained by multiplying the LR w_{+1}-w_{-1} with the data that is generated from the PMF factorized matrix. That is S_{u,i} = (w_{+1}-w_{-1}) * [U_{u}.*V_{i}] where U and V are factorized user and movie matrix.

Then for a given user, the ranked list can be obtained by sorting all the score of the unrated movie for that user.

b. Report the best results you can get on dev set from training LR-LETOR with features from PMF (10 pts)

Num of Latent Factors from PMF	**NDCG@10**	**NDCG@20**	**NDCG@30**	**Runtime(sec)***
10	0.931820	0.936993	0.937517	419.545994043
20	0.934840	0.939623	0.940082	452.803373814
50	0.932427	0.937123	0.937649	501.881941412
100	0.927186	0.932389	0.932903	437.414888292

*report the time of training LR-LETOR after PMF outputs the features. For each case, you might want to tune the regularization parameters of both PMF and LR.

6. Analysis of results (20 pts)

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, different parameters, 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.

1. User-User VS Movie-Movie VS PMF_baseline:

The User-User and Movie-Movie CF approaches attain similar performance: 0.93 to 0.94 NDCG. The difference of the similarity metric of dot product VS cosine similarity is not significant. But K = 100 attains better NDCG than K = 10 and 500 for other settings. The performance of PMF is not as high as CF (0.89 to 0.92 NDCG). When the number of latent variables are 20 the NDCG is highest. But the running time increases significantly when number of latent variable increases.

The reason of poor performance of PMF is that it tries to minimize loss function which includes both high and low ratings. But for low ratings, this is not what we really care.

2 PMF_baseline VS RankSVM:

The performance of RankSVM is higher than PMF_baseline (0.93 to 0.94 NDCG), but when the number of latent variables increase, the NDCG does not increases. Too many hidden variables cannot represent the model very well. But the training data of SVM only distinguished the movie with rating 5 and rating 1. This classifier can better generate a ranked list which distinguished the high rating movie with low rating movie.

3 PMF_baseline VS LR_LETOR:

The performance of LR_LETOR is higher than PMF_baseline(0.93 to 0.94 NDCG). When the number of latent variable is 20, the NDCG is the highest. Similar as RankSVM, too many hidden variable cannot extract good feature which serves as the input of the classifier. Compared with RankSVM, NDCG does not change too much. The training time is shorter, and as latent variable number increases, training time does not varies a lot. This is because LR uses SGD as the optimization method and it can attain a faster convergence speed for large size of input.

7. The software implementation (15 pts)

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.

The PMF part is same as the in the HW4. No imputation is used. The training data set is constructed by using the PMF factorization matrix.

To speed up the experiment, the result of the PMF factorized matrix can be stored into a file, so for experiment 3, 4 and 5, we only need to do the PMF training once. The other two experiments can read from the factorized matrix file.

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

Library: sklearn, I use the LinearSVC for SVM experiment.

Data Structure: sparse matrix for storing training data, list for storing training data label

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

Weakness: since I use sklearn, all data structure are stored in the main memory before they are put into SVM or LR. The memory requirement is really huge. For 50 latent variables, the memory is 14GB to train the SVM classifier.

Strength: No need to do extra file IO for training classifier, save time.