**Documentation for Recommendation System**

Yuepeng Fan

**Abstract**

Recommendation System has attracted much interest in last decade. In project 2, we implemented a recommendation system based on the algorithm taught in the lecture. Baseline Algorithm, neighborhood method, and low rank matrix factorization are used in this system. The RMSE of proposed method on training set is 0.7809, and 0.9439 on testing set and validation set.

**1. Overview**

In the project, we have 80878 training samples in total. The number of users and items are 943 and 1682 respectively. Before starting the algorithm, we divided the training samples into two parts, training\_set and validation\_set. The ratio is 9:1.

This is the high level algorithm:

1) Prepare the training data in matrix form .

2) Predict the hidden ratings with baseline method. Calculate the difference between ground truth and prediction .

3) Use low rank matrix factorization method to calculate the . Firstly using SVD to get the such that . Then fix P and optimize Q, followed by optimizing P and fixing Q. Iterate until they converge. Finally get the and and . Then we get.

4) Use neighborhood method to get with the basis of . We implement two methods, user based neighborhood method and item based neighborhood method. It is shown from the experiment that item based method works better.

5) Combine the result of low rank matrix factorization method and neighborhood method with the factor ：

The details of each part will be explained in the following chapters.

The 2nd part is mainly coded by Menge Li, the 4th part is mainly coded by Weili Zhe, and the,3th is contributed by Yuepeng Fan.

**2. Baseline Method**

Generally, a baseline predictor can be used in such formular

Where and are the user and item baseline respectively. They can be get by the following method

This is the most common way for baseline calculator. We then use the ground truth and predicted value to calculate the difference .

**3. Low Rank Matrix Factorization**

We use Singular Value Decomposition (SVD) to get the low rank matrix. This kind of method finds the global feature of the matrix.

The number of rank is related with overfitting issue. In the project, we tried several numbers, like 15, 20, 22, 25, 30, 40. Finally, 20 was chosen because of its low RMSE.

Using SVD, we get U, S, VT. Then, let P = U\*S, Q = VT.

1) Using P and Q as the initial value for optimization.

2) Keep P unchanged, change Q to minimize using least square regression.

3) Then keep Q unchanged, changed P to minimize .

4) Repeat step 2 and 3 until P,Q converge.

Finally, we get the difference and then get prediction of this method.

**4. Neighborhood Method**

In neighborhood method, there are two kinds of algorithm. One is user based and the other is item based.

To generate predictions based on the difference , we need to calculate the similarity of different users. *cosine\_similarity* was called in this function. The output of *cosine\_similarity* is a (943 \* 943) matrix. If two users are very similar, this function will return a number close to 1, return a number close to 0 otherwise.

Each rating is calculated in the following method:

is the element in . is the similarity between users and .

Then predicted value

The item based method is similar. We can get

And get .

**5. Other Part**

In this project, we use RMSE to evaluate our model. The validation part is not used in any training model, so the result is trustful.

We use combination method to combine low rank method and neighbor method. We initialize the number , increase by step of 0.01 finally to 1. And choose the best combination result according to the RMSE. As seen from the result, the item based prediction works better than user based prediction. So item based method is applied in this combination model.

Also, we regularize the result of prediction to be between 1 and 5 and separated by 0.5.

**6. Findings**

1) During the optimization of P and Q in low rank matrix method. If we initialize P and Q randomly, the optimization iterations will be more than 500 round. It always reaches the maximum number of iterations we set. However, if P and Q are initialized using SVD, the optimization will stop no more than 4 iterations.

Our guess is that this might be the result of SVD.

2) During the combination part, the optimized is always 1. In other word, it works well when we only use low rank method without neighborhood method.

This might be the result of overfitting. Also another reason is that the matrix is very large but the known data set is not large enough.

**7. Conclusion**

We implemented an recommendation system. We implemented three models in the system. The RMSE for each one is

|  |  |  |
| --- | --- | --- |
|  | Training | Validation |
| Baseline method | 0.9242 | 0.9670 |
| Low rank method | 0.7809 | 0.9439 |
| User based neighborhood | 0.9210 | 0.9650 |
| Item based neighborhood | 0.9153 | 0.9600 |

However, since the time is limited, the optimization of this algorithm is not enough.

**References**:

[1] https://cambridgespark.com/content/tutorials/implementing-your-own-recommender-systems-in-Python/index.html

[2] https://github.com/srajangarg/movie-recommendation-algorithms

[3] http://scikit-learn.org

[4] Wen, Zaiwen, Wotao Yin, and Yin Zhang. "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm." *Mathematical Programming Computation* (2012): 1-29.

[5] https://www.analyticsvidhya.com/blog/2016/06/quick-guide-build-recommendation-engine-python/

[6] Lu, Yuan, and Jie Yang. "Notes on Low-rank Matrix Factorization." arXiv preprint arXiv:1507.00333 (2015).