SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Group Work Report

Topic 3.4

Recommender system using collaborative filtering

z3485805	Xiuye Yuan
z3485944	Zhengyu Guo
z5216787	Ying Qian
z5223541	Ruijun Zhou

Master of Information Technology

Submitted: August 2019

Table of Contents

1. Introduction

In the time during which knowledge and information develop quickly and widely, humanity service is a kind of popular pattern of services. Among the extensive online services, recommender systems play a vital role in alleviating information overload, including E-commerce, online news and social media sites (He et al., 2017). The recommender system is designated for selecting appropriate items among tons of information, which maximizes the user experience. To achieve the recommendation performance, two methods are widely used: content-based filtering and collaborative filtering (Pazzani 1999).

Content-based filtering predicts the action of users based on users' past actions and the entity of items (Basilico & Hofmann 2004), while collaborative filtering is an approach, through modelling preference of users on items based on the interactions of other users with similar acting pattern (Sarwar et al., 2001).

In the last decade, a variety of CF (collaborative filtering) algorithms have been proposed, two major classes of that are memory-based approaches and model-based approaches (Breese et al., 1998). Memory-based CF predict a user's ratings based on the ratings given by like-minded users, similar to nearest-neighbour scheme. In contrast, model-based CF first learns a descriptive model of user preferences and then uses it for predicting ratings (Yu et al., 2004).

Through the analysis of various collaborative filtering techniques, He and his group partners (2016) provided the matrix factorisation (MF) is the most suitable technique for movie recommendation (Koren 2008), which projects users and items into a shared latent space and determined the decisions of users based on those latent features (Mnih & Salakhutdinov 2008). The specific presentation and application of MF will be showed in the following method section.

In our experiment, our goal is to use collaborative filtering as our approach to develop a movie recommender system with accuracy at a relatively high level. This report will fully explore the application of matrix factorisation on movie recommender system.

The main contributions of this work are as follows.

1. According to the provided datasets, we find out the real effective features through the training sets that we have partitioned.
2. We perform extensive experiments and analyze the results of experiments to demonstrate the effectiveness of our approaches through the other test sets.
3. Deep learning and analyzing the application of collaborative filtering in recommender system in the real world.

2. Data preparation

The dataset we used in the collaborative filtering recommender systems is the MovieLens Dataset, which is one of the most common open source datasets for education and development. For the first approach, a small version dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from [MovieLens](http://movielens.org). It contains 100,000 ratings and 3600 tag applications applied to 9000 movies by 600 users.

In order to achieve the best performance in the experiment of matrix factorisation, we need pretreated dataset. The data are contained in the files ‘movie.csv’, ratings.csv’ and ‘tags.csv’. However, not all the data in those files are compulsory, we need to simplify the clauses. In this process, the columns with the title timestamp in ratings.csv and tags.csv files have been dropped. Furthermore, the title column in movie.csv and userId column in tags.csv also dropped respectively.

After that, the data need to be separated in two parts which will be used in training and testing later. Firstly, we used 0.8 as the separation rate that means the 80% of the entire data are concerned with the training part. In the implementation, train_test_split module in sklearn is used to perform the data separation. The training and testing part were sorted by userId and movieId.

In addition, the dataset separation will lead the asymmetry phenomenon, which means that the training set does not have enough training resource for some user. That may lead the test result has error. That can be discussed later.

3. Related work and Method

In recent year, collaborative filtering (CF) has been successfully used to provide users with personalised products and services. In this report, the CF (collaborative filtering) algorithms has been applied in the recommender system as the main method. As mentioned previously, the CF algorithms has two representative approaches: Memory-based CF and Model-based CF. In the next segmentations, the report will give specific explanations to each approach.

3.1 Memory-based CF

Memory-based collaborative filtering (CF) recommends a set of user preferences based on the project. The basic idea of this approach is that the interests of users are more likely to be consistent with the interests of users who share similar preferences with them (Jeong et al., 2010). The first CF system, Group-lens (Resnick et al., 1994) and Ringo (Shardanand & Maes 1995) belong to this category. In the literature, the term collaborative filtering is occasionally only used to refer to memory-based methods (Yu et al., 2004). The techniques are also known as nearest-neighbour or user-based CF, are more popular and widely used in practice. (Sarwar et al., 2001).

There are two main types of memory-based collaborative filtering algorithms (Resnick et al., 1994):

3.1.1 User-based

Resnick et al. (1994) provided a U-I rating matrix, which is a typical user-based CF method that can be used to predict a user's rating of a target item by aggregating a rating similar to that previously given by the user. In other words, the user will be recommended movies based on similarity and recommend movie from similar users.

3.1.2 Item-based approach

It is quite similar to user-based algorithm, but instead of user, it focuses on items that can quickly recommend a set of items and have been shown to produce recommendation results through using the pre-computed model (Karypis 2001). Normally, historical information is analyzed to identify relationships between items such that purchase of items often results in the purchase of another item (Breese et al., 1998).

3.1.3 Similarity metrics

Moreover, the choice and computation of a similarity measure between users is critical to rating items. (Yu et al., 2004). There are three distance similarity metrics that are usually used in collaborative filtering:

1. Jaccard Similarity (Strehl & Ghosh 2000)

The similarity is depended on the numbers of users who have rated item A and B divided by the number of users who rated one of A and B. It has been wildly used in such area where the number of rating does not provide but just the Boolean value. Such as the product has been sold out or the picture been liked.

2. Cosine Similarity (Ye 2011)

The cosine similarity is calculated by the angle between the 2 vectors of the variables A and B, in this report, we calculated the user's ratings. The more closer vector will get the more smaller angle which leads the cosine value larger. The cosine of the angle between the vectors is within the values between 0 and 1.

3. Pearson Similarity (Ahlgren et al., 2003)

It also is called Pearson's correlation coefficient and it is the test statistics that measures the statistical relationship, or association, between two continuous variables which is based on the method of covariance. The magnitude of the association, correlation and the direction of the relationship could be provided by this method.

3.1.3 Analysis

Table 1. The advantage and disadvantages in different base CF model (Drachsler et al., 2007)

	Advantages	**Disadvantages**
User-based CF	-No content analysis -Domain-independent -Quality improves -Bottom-up approach -Serendipity	-New user problem -New item problem -Popular taste -Scalability -Sparsity -Cold-start problem
Item-based CF	-No content analysis -Domain-independent -Quality improves -Bottom-up approach -Serendipity	-New user problem -Popular taste -Sparsity -Cold-start problem

3.2 Model-based CF

Based on prediction models, the approach of Model-based CF can train and use the U-I matrix in whole or in part as input (Adomavicius &Tuzhilin 2005). Then the trained prediction models can then be used to generate recommendations for individual users (Shi et al., 2014).

Nowadays, matrix decomposition (MF) technology has attracted considerable attention due to its advantages in scalability and accuracy, as evidenced by algorithms developed in the Netflix competition (Koren et al., 2009). In the following three segments, the report will introduce the MF and SVD (Singular Value Decomposition).

3.2.1 Matrix Factorization Model

Generally, MF models learn low-rank representations (also referred as latent factors) of users and items from the information in the U-I matrix, which are further used to predict new scores between users and items (Shi et al., 2014). Behind these models, the idea is that the user's attitude or preference is determined by some unobserved factors (Mnih&Salakhutdinov 2008).

In the following experimental implementation section, the report will specifically state the analysis of MF that apply in this project through combine the model with our code.

4. Experimental Implementation

4.1 Memory-based

There are two main types of memory-based collaborative filtering algorithm.

4.1.1 User-User Collaborative Filtering:

As we did in the data preparation, the dataset of training is used because it contains all information that we need for determining the similarity between users based on their ratings for given movies. In the dataset of training, there are sets of userID, movieID and ratings. The userID is placed in the head of row and the movieID is put in the head of column respectively in dataset.

The dataset matrix contains lots of empty value since there are huge number of movies and the users do not watch all of them. The method like matrix factorization could be used to deal with this situation that what we did in the model-based part. In this part, we use the user average rating over the row to represent the predicted rating for the movies that have not been watched by any user. On the other hand, the movie average over the column to represent the predicted rating for new users.

After the replacement operation, the dataset matric with rating for every user and all movie are generated. Then, we can calculate the similarity between each users by the previous given cosine similarity function.

user_similarity = 1 - pairwise_distances(self.train.astype('float16'), metric='cosine')

Therefore, the user-user similarity matrix within the range of 0 to 1 table was generated. The next step is predicting the ratings that were not provided by the dataset. Using the prediction results, we can compare them with the testing data which is a way to validate the accuracy of our recommender system later. In the prediction part, we will focus on the similarity between predicting user A and top 10 similar users. For example, if we want to predict the rating for movie A that user A has not rated, we will choose 10 most similar users with user A where the 10 users have watched movie A. Then, the predicted rating for user A is calculated by the weighted average of the 10 users' ratings. The formula is:

$$PredictRating = \frac{\sum(u_i \times sim_i)}{\sum sim_i}$$

where the u_i means the rating of user i and sim_i is the similarity between user A and user i. If it is a new movie or it has not been rated, the estimated rating will be the mean rating of user A.

4.1.2 Item-Item Collaborative Filtering

Item based process is very similar to the user-user collaborative filtering. After the data preparation, we can construct the item-item similarity matrix based on the user-item matrix. Here the similarity between two items we also calculate with cosine distance which is same as the one used in user-user similarity calculation. Afterwards, we can proceed to find the similar items and do prediction. Similar with the user-based, the item-base used top 15 similar movies to predict. In same way:

$$PredictRating = \frac{\sum(Movie_i \times sim_i)}{\sum sim_i}$$

Here the $movie_i$ means the rating of movie I and sim_i is the similarity between predicting movie and movie i.

4.2 Model-based

The approach of model-based collaborative filter is achieved by matrix factorisation (MF). Matrix factorisation provides an approach to learn user and item location. The basic method of MF is going to form the rating matrix which contains every user-item pair. In this matrix, there are lots of missing rating values since the dataset is sparse. The goal of MF is to fill in these missing values by learning during iterations. We can combine the MF and recommendation systems as we have prediction of every missing rating for users. So, we could do recommendation by choosing the highly rated movies for a particular user from the matrix.

Firstly, the dataset needs to be initialized into training set and testing set as user-item matrix. As previous declared, we split it in ratio of 8:2 for training and testing. Then, based on the main purpose of MF, the rating matrix U need to transform into two matrix multiplication. One matrix P_u is user-feature matrix that means the rating of feature k, another one Q_i is feature-movie matrix that means the degree of movie on k. Then the both users and items with latent need to joint together with latent feature space of dimension k. Thus, the combined MF model of predicted rating r_{ui} is:

$$r_{ui} = P_u^T Q_i = \sum_{k=1}^{k} P_{uk} Q_{ki}$$

Where k represents the number of latent features and k will not be greater than either number of u or i.

Furthermore, as the result of some users' ratings usually have subjective features which are independent to others may produce error. We also produce the matrix factorization model with biases to compare the rating result with the original unbiased MF model where the biases also can be viewed as baseline predictors. The biases function presented as:

$$b_{ui} = b_u + b_i + \mu$$

Here the parameter b_u is the bias between rating presented by user u with the mean rating, b_i represents the bias of rating of movie i with mean rating, μ stands for the average rating. Then, our biased MF model becomes:

$$r_{ui} = P_u^T Q_i + b_u + b_i + \mu$$

Finally, the way to reduce the overfitting is to introduce the regularization. This is implemented by adding parameter β and modify the squared error as follows:

$$e_{ij}^2 = (r_{ij} - \sum\nolimits_{k=1}^{K} p_{ik} q_{kj})^2 + \frac{\beta}{2} \sum\nolimits_{k=1}^{K} (||P_u||^2 + ||Q_i||^2)$$

The parameter β is used to control the magnitude of the user-feature and item-feature vectors such that P and Q would give a good approximation of R without having to contain large numbers, in our MF model, parameter β was set as 0.02.

In order to accelerate the training convergence speed, the Stochastic Gradient Descent (SGD) algorithm (Bottou 2010) was implemented. Instead of computing the gradient every node exactly, each iteration estimates the gradient on the basis of a single randomly selected sample z_t.

$$\omega_{t+1} = \omega_t - \gamma_t \nabla_\omega Q(z_t, \omega_t)$$

Where the ω_t is the stochastic step and $\gamma_t$is the learning rate, in our model, $\gamma_t = 0.001$.

Combined all of previous assumption and matrix factorization function, experienced a series training, we can get the final user-rating matrix about every movie. Which means we can predict any movie rating based on the result when the user and movie were provided.

4.3 Evaluation

There are many evaluation metrics but one of the most popular metrics used to evaluate accuracy of predicted ratings is Root Mean Squared Error (RMSE). It is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum(R_i - \overline{R_i})^2}$$

N is the total number of ratings in the test dataset, R_i is the provided real rating, $\overline{R_i}$ denotes the predicted ratings by model. The performance of the estimation will mainly depend on the value of RMSE rather than accuracy and the aim is to reduce the RMSE as much as possible.

5. Results

In this subsection, all the result about memory-base and model-base will be presented.

Table 2. Memory-based CF result within 100k dataset.

Memory-Based CF	User-Based RMSE	Movie-Based RMSE
1	1.1432	1.0583
2	1.1422	1.0513
3	1.1351	1.0469
4	1.1298	1.0457
5	1.1430	1.0633
Average	1.1387	1.0531

In the experiment, we run 5 tests with different random state for cross validation. The analysis was depending on the average result. From table 2, we can get that the movie-based CF model has more accuracy on the movie rating than user based, the percentage of movie-based result higher than the user-based result about:

$$\frac{1.1387 - 1.0531}{1.0531} \times 100 = 8.31\%$$

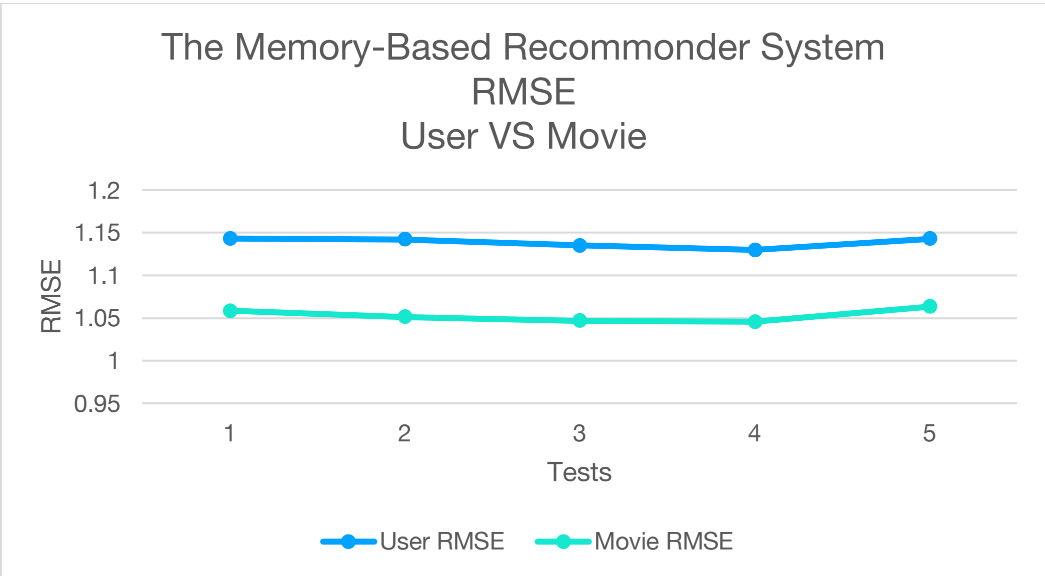

Figure 1. The RMSE result comparation between User-based and Movie-based Recommender system.

The previous figure 1 shows that the movie-based CF results are always better than the user-based.

Table 3. Memory-Based CF within 1m dataset

Memory-Based CF	User-Based RMSE	Movie-Based RMSE
1m	0.9837	0.9315

As the table 3 shown, it is clearly that 1m dataset is 15.76% more accurate than result of 100k dataset. The movie-based CF result also is better than the user-based one.

Table 4. The RMSE analysis with different similarity function.

Random State	Pearson's RMSE	Cosine's RMSE
23	1.1437	1.1423
43	1.1427	1.1432
63	1.1304	1.1299
83	1.1442	1.1431
103	1.1449	1.1445
AVG	1.1412	1.1406

Table 4 shows the different similarity calculation function does not have significant impact on RMSE results.

Table 5. The model-based CF result about 100k datasetwithin different latent features.

Feature	RMSE for biased MF	RMSE for unbiased MF
8	1.0694	1.0797
16	1.0944	1.1075
32	1.022	1.0619
64	1.0843	1.1186
avg	1.0676	1.0919

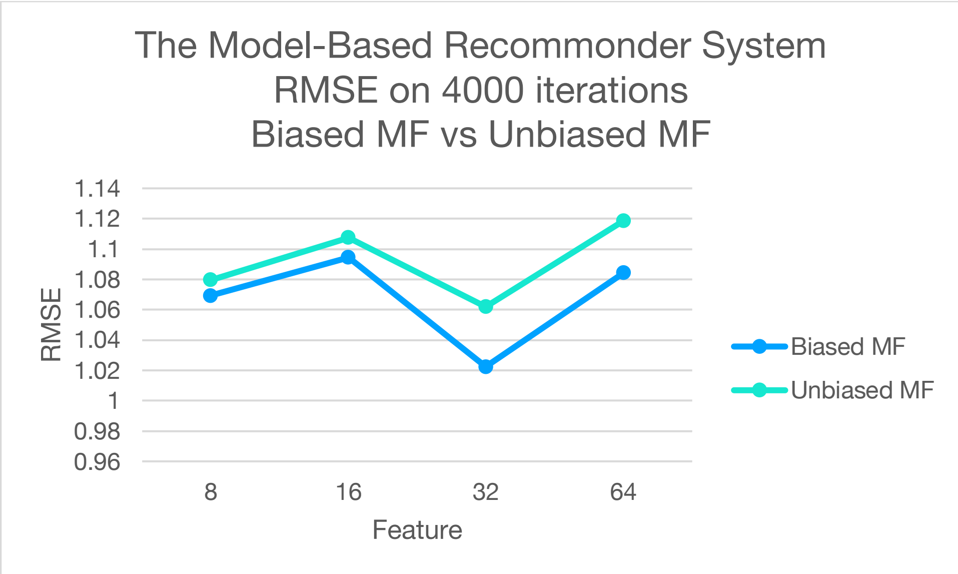

Figure 2. The RMSE result comparation between Biased MF and Unbiased MF.

Based on the table 5 and figure 2, it presents the biased MF is a little bit suitable in the recommender system.

Table 6. The model-based CF result about 1m dataset with latent feature 8.

Model-Based CF	Biased RMSE	Unbiased RMSE
1m	0.8779	0.8844

Table 6 introduces the RMSE result experimented with 1m dataset. The simulation takes around 3 days to finish, there are some reasons will be discussed in next section. Compared with the result of 100k dataset, the 1m dataset CF improves the RMSE by 20.71% and 24.38 for biased and unbiased model respectively. Furthermore, the model-based CF also increases the accuracy of recommendation around 11.22% by choosing the memory-based user-user model in same dataset (1m).

Table 7. The RMSE result within different iterations on feature 32.

Iterations	RMSE	RMSE
1000	1.1601	1.1856
2000	1.1739	1.2042
3000	1.1763	1.2129
4000	1.0224	1.0619
5000	1.1739	1.2167
6000	1.1798	1.2118

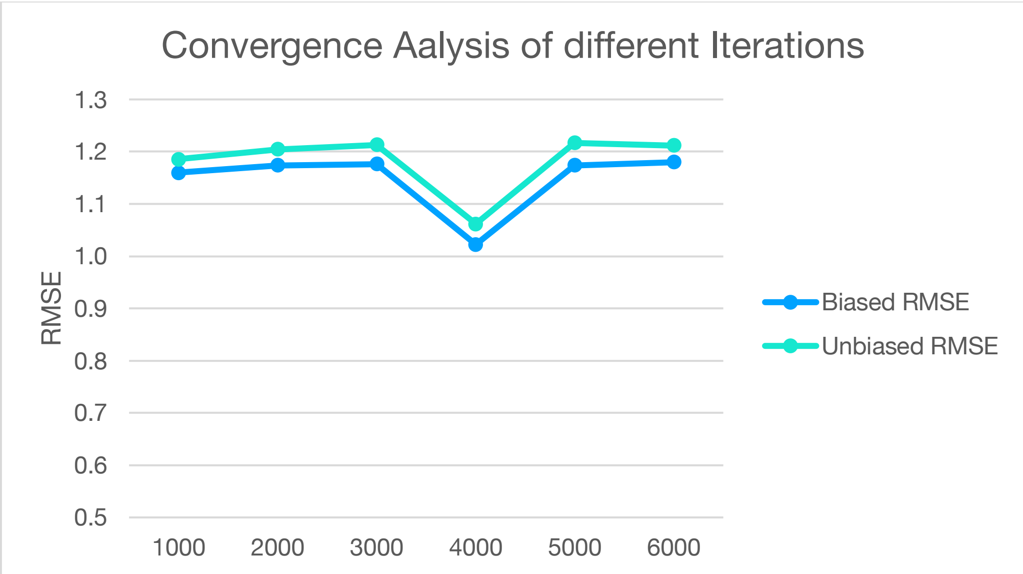

Figure 3. The convergence analysis based on different iterations.

As the previous table and figure, we choose iteration 4000 within experiment.

Table 8. RMSE result of memory-based and model-based CF in different datasets.

100k Dataset RMSE		**1m Dataset RMSE**	
Biased-Model 1.0676	Unbiased-Model 1.0919	Biased-Model 0.8844	Unbiased-Model **0.8779**
User-based 1.1387	Item-based 1.0531	User-based 0.9837	Item-based 0.9315

6. Discussion and Evaluation

6.1 Analysis of two approaches

This study is conducted based on two MovieLens datasets which are MovieLens 100K with 100.836 ratings created by 610 users across 9742 movies and MovieLens 1M with 1,000,209 ratings created by 6040 users across 3900 movies. In both datasets, users rated movies in the range of 1 to 5. We expected to use different approaches which include memory-based CF and model-based CF to simulate the recommender system in order to provide the one with the least RMSE (root mean square error).

6.1.1 Memory-based CF

Fistly, different approaches are applied to calculate the smiliarity between users or items. As it can be seen from the result, the approach does not influence the result much. As cosine similarity has a relatively low RMSE than pearson similiarity, it is used in our implementation.

Furthermore, it is obvious that the item-based content-based CF approach is slightly more accurate than that of user-based approach. One of the possible reason may be that the relationship between movies is more robust than that of users due to the large variety of different rating prefrences among users.

Memory-based CF is easy to implement and compute and it is very efficient. The estimatings can be calculated within a short period. However, the performance may not be that satisfiable in terms of accuracy and RMSE.

6.1.2 Model-based CF

In the model-based CF, the number of features which can also be treated as number of hidden nodes may have a considerable impact on the performance. Therefore, the appropriate value of feature numbers should be chosen. After multiple testing on the same dataset, it is shown that 32 hiddle features may be suitable for both biased and unbiased MF.

In addition, another serious problem of the model-based CF that need to be solved is overfitting. During the testing, it can be illustrated that 4000 iterations have a relatively low RMSE and this is because the model has been just converged near that number of iteration and further training may lead to overfitting.

During the design process, the effect of the biases is also evaluated and it can be seen from Figure 2 that adding biases to users and items improves the performance of the model.

Table 8 indicates that the matrix factorisation has much lower RMSE compared with the memory-based CF. Although MF takes time to use SGD to learn the model properly, it has a significantly lower RMSE than that of memory-based CF and it can predict the ratings efficiently.

6.2 Error

Although the result has achieved the goal of recommend systems through the two approached applied in data sets, this project still exists some error that will affect the final result.

6.2.1 High RMSE

Although various Figures has given good results for different approaches, the RMSE values in all cases are still higher than expectation. One of the reasons of high RMSE is the sparsity of the data. Since the large variety of movies, many movies are rated by only few users, which results in weak correlation between ratings, and this increases the difficulty of providing valid information. Therefore, some improvement could be added on the project to solve the problem of data sparsity.

7. Conclusion and future work

In conclusion, both memory-based collaborative filtering and model-based collaborative filtering are implemented in this project. And both user-based and item-based content-based CF are included. From the statistic results, model-based CF has a better performance than memory-based CF in terms of RMSE which is usually used to evaluate the accucucy of the recommend systems. Therefore, matrix factorization will be chosen to implement the recommend system due to the lowest RMSE. This advantage of this method is mainly high accuracy and thus more reasonable prediction of users' preferences. On the other hand, drawback is it takes long time to learn. However, in the real world, we only need to train the model once and we can easily recommend the suitable movies for particular users.

In other words, memory-based CF can be treated as lazy learning as it takes tiny time to learn the model but predicting is time consuming. In contrast, model-based CF is one application of eager learning. After it has completely learned the model, predicting can be processed very efficiently and this is more suitable for the application like recommend system with considerably large dataset.

Furthermore, some improvement may be added on this implement in the future.

Firstly, pre-training may be helpful in some cases. It means the model is pre-trained with one dataset and the weights achieved after training are used as the initial weights to train the actual dataset. This may be helpful when two datasets has similar features and in this case pre-training may improve the efficiency of the actual training process.

Moreover, SVD++ is one approach to improve the performance of matrix factorization. In SVD++, it is assumed that the movies that one user has seen and rated also contains the information about the user's preference. Therefore, this additional information is added to model and it is approved that this may further improve the performace of basic SVD. The ratings are calculated as following:

$$r_{ui} = q_i^T (p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j) + b_u + b_i + \mu$$

Where I_u is the set of movies that the user has seen and y_j represents the preference bias of user to the movie.

8. References

Basilico, J. and Hofmann, T., 2004, July. Unifying collaborative and content-based filtering. In Proceedings of the twenty-first international conference on Machine learning (p. 9). ACM.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177-186). Physica-Verlag HD.

Breese, J.S., Heckerman, D. and Kadie, C., 1998, July. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp. 43-52). Morgan Kaufmann Publishers Inc..

He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.S., 2017, April. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web (pp. 173-182). International World Wide Web Conferences Steering Committee.

He, X., Zhang, H., Kan, M.Y. and Chua, T.S., 2016, July. Fast matrix factorization for online recommendation with implicit feedback. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 549-558). ACM.

Jeong, B., Lee, J. and Cho, H., 2010. Improving memory-based collaborative filtering via similarity updating and prediction modulation. *Information Sciences*, *180*(5), pp.602-612.

Koren, Y., 2008, August. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 426-434). ACM.

Mnih, A. and Salakhutdinov, R.R., 2008. Probabilistic matrix factorization. In Advances in neural information processing systems (pp. 1257-1264).

Pazzani, M.J., 1999. A framework for collaborative, content-based and demographic filtering. Artificial intelligence review, 13(5-6), pp.393-408.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J., 1994, October. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*(pp. 175-186). ACM.

Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms. Www, 1, pp.285-295.

Shardanand, U. and Maes, P., 1995, May. Social information filtering: algorithms for automating" word of mouth". In *Chi*(Vol. 95, pp. 210-217).

Yu, K., Schwaighofer, A., Tresp, V., Xu, X. and Kriegel, H.P., 2004. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, *16*(1), pp.56-69.

9. Appendix

Rating Dataset Format:

1	userId	movieId	rating	timestamp
2	1	1	4	964982703
3	1	3	4	964981247
4	1	6	4	964982224
5	1	47	5	964983815
6	1	50	5	964982931
7	1	70	3	964982400
8	1	101	5	964980868
9	1	110	4	964982176
10	1	151	5	964984041
11	1	157	5	964984100
12	1	163	5	964983650
13	1	216	5	964981208
14	1	223	3	964980985
15	1	231	5	964981179
16	1	235	4	964980908
17	1	260	5	964981680
18	1	296	3	964982967
19	1	316	3	964982310
20	1	333	5	964981179
21	1	349	4	964982563
22	1	356	4	964980962
23	1	362	5	964982588
24	1	367	4	964981710
25	1	423	3	964982363
26	1	441	4	964980868
27	1	457	5	964981909
28	1	480	4	964982346
29	1	500	3	964981208

Movie Dataset Format:

1	movieId	title	genres
2	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
3	2	Jumanji (1995)	Adventure|Children|Fantasy
4	3	Grumpier Old Men (1995)	Comedy|Romance
5	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
6	5	Father of the Bride Part II (1995)	Comedy
7	6	Heat (1995)	Action|Crime|Thriller
8	7	Sabrina (1995)	Comedy|Romance
9	8	Tom and Huck (1995)	Adventure|Children
10	9	Sudden Death (1995)	Action
11	10	GoldenEye (1995)	Action|Adventure|Thriller
12	11	American President, The (1995)	Comedy|Drama|Romance
13	12	Dracula: Dead and Loving It (1995)	Comedy|Horror
14	13	Balto (1995)	Adventure|Animation|Children
15	14	Nixon (1995)	Drama
16	15	Cutthroat Island (1995)	Action|Adventure|Romance
17	16	Casino (1995)	Crime|Drama
18	17	Sense and Sensibility (1995)	Drama|Romance
19	18	Four Rooms (1995)	Comedy
20	19	Ace Ventura: When Nature Calls (1995)	Comedy
21	20	Money Train (1995)	Action|Comedy|Crime|Drama|Thriller
22	21	Get Shorty (1995)	Comedy|Crime|Thriller
23	22	Copycat (1995)	Crime|Drama|Horror|Mystery|Thriller
24	23	Assassins (1995)	Action|Crime|Thriller
25	24	Powder (1995)	Drama|Sci-Fi
26	25	Leaving Las Vegas (1995)	Drama|Romance
27	26	Othello (1995)	Drama
28	27	Now and Then (1995)	Children|Drama
29	28	Persuasion (1995)	Drama|Romance
30	29	City of Lost Children, The (Citv© des enfants perdus, La) (1995)	Adventure|Drama|Fantasy|Mystery|Sci-Fi

This is the simple demonstration of how the work shows the recommend result for a particular user. Top unseen movies with highest ratings will be recommended in descending order of ratings.

```
- - - - - - - - - - - - - - - - -  Recommand Results  - - - - - - - - - - - - - - - - -
|                1       Saving Santa (2013)                                            |
|                2       Lamerica (1994)                                                |
|                3       Wings, Legs and Tails (1986)                                   |
|                4       Heidi Fleiss: Hollywood Madam (1995)                           |
|                5       What Men Talk About (2010)                                     |
|                6       Palindromes (2004)                                             |
|                7       Live Nude Girls (1995)                                         |
|                8       In the Realm of the Senses (Ai no corrida) (1976)              |
|                9       What Happened Was... (1994)                                    |
|                10      Thin Line Between Love and Hate, A (1996)                      |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```