Recommender system using collaborative filtering

COMP9417 Assignment

Peiguo Guan	z5143964
Taoran Sun	z5150998
Wanze Liu	z5137189
Yunhe Hu	Z3351805

1. Introduction

In the era of internet, individuals are overwhelmed with huge amount of information online. This common situation has brought an obvious problem that internet users have difficult to choose the most appropriate services or contents. Nowadays, individuals are seeking more personalized services and experiences from service and content providers. Amazon, widely known as an online book reseller, deployed a recommender system in 1998 which can recommend books to its customers based on purchasing history, past preference and demographic information. As at today, recommendation systems are used not only for books, but also almost all types of internet entertainment services including movie, music and news.

The purpose of this assignment is to implement a collaborate filter recommender system based on a movie rating dataset and recommend a list of movies to a user based on the user's previous rating history. Three collaborate filtering algorithms are tested in this assignment including User-Based method, Content-Based method and Singular Value Decomposition (SVD) method.

2. Related Work

The MovieLens dataset has been used by a large number of researchers to develop collaborate filtering system since 1999. Herlocker, Konstan, Borchers and Riedl (1999) conducted a study to solve the collaborate filtering problem by using similarity analysis and neighborhoods selection. Herlocker et al. also introduced two different methods to compute similarity, namely Pearson correlation coefficient and Spearman rank correlation. In addition, the effective of parameter normalization methods has been examined. The result indicates that average z-score normalization method is the best method for personalized recommender system. However, the average deviation from mean normalization method performs similar to the average z-score normalization method and performs much better for non-personalized recommender system.

Howe and Forbes (2008) conducted a further examination on the parameter normalization based on Herlocker et al.'s study. The result indicates that parameter normalization has a significant role when applying collaborate filtering algorithm and cosine vector similarity could be better than Pearson correlation coefficient similarity in some cases. The study also mentioned that different datasets favor different parameter normalization methods.

3. Implementation

The dataset used in this assignment is the ML-100K one downloaded from MovieLens website at https://grouplens.org/datasets/movielens/. There are several different data files in this dataset including user rating, user demographic information, user occupation and movie information.

In the first step, only the ratings given by each individual user are considered. The extra date about user demographic information and movie specific information are ignored. The user id, movie id and rating are extracted from u.data file and the movie id and movie title are extracted from u.item data file. A M×N matrix is created by combing user and movie data where each row represents a user, each column represents a movie and the value represent user's ratings against a movie.

To implement the prediction model, similarity among users and movies have to be calculated. In this implementation, Cosine Similarity method is chosen to calculate the similarity. The formula is given by:

$$\text{Cosine Similarity: } Sim(u_i, u_k) = \frac{r_i \cdot r_k}{|r_i||r_k|} = \frac{\sum_{j=1}^{m} r_{ij} r_{kj}}{\sqrt{\sum_{j=1}^{m} r_{ij}^2 \sum_{j=1}^{m} r_{kj}^2}}$$

To recommend a list of movies to a target user, the main process is to calculate the similarity between the target user and all other users choose highest rated movies from those N users. The formula is given by:

$$r_{ij} = \frac{\sum_k Similarties(u_i, u_k) r_{kj}}{Number\ of\ ratings}$$

Because different users have different rating baseline when given rating, an absolute rating is not appropriate in this implementation. A method is used to normalize the individual rating. The overall formula to calculate rating is given by:

$$r_{ij} = \overline{r_i} + \frac{\sum_k Similarties(u_i, u_k)(r_{kj} - \overline{r_k})}{Number\ of\ ratings}$$

In addition to the above method, K Nearest Neighbors method is used to show the difference between using all user's information and only the top K similar users' information. Once the similarity is calculated, select the top K (an arbitrary number) similar users and calculated movie ratings based on the average ratings given by these K users.

To coupe with the problem with cold start (new user with a small amount of previous ratings), a content-based similarity among all the movies is also calculated using the Cosine Similarity method. The result shows similarities between each pair of movies. In this case, the matrix that was used in calculating user similarity is transposed. each row represents a movie, each column represents a user and the value represent user's ratings against a movie. Once the similarity matrix for all the movies is calculated, predict an

unrated movie rating based on the similarity between this movie and all other movies that have been rated by the user and the ratings for those rated movies.

The predicted rating for a specific movie is compared with the actual rating given by a user to show the error. The Root-Mean-Square Error method is used to illustrate the overall error for the test dataset.

In the second step, SVD is also tested to show the movie rating prediction. SVD is a matrix factorization technique that used to reduce the features of a dataset. A matrix is represented by two matrices whose dot product is the original matrix. In this implementation, Python's Scipy SVDS function is used. In stead of computing one singular value and vector, SVDS provides addition parameter k that determine the number of singular value and vector to be computed. k values from 1 to 50 are used to demonstrate the impact of different k values.

In the third step, additional information is considered when calculating user-based similarity matrix and content-based similarity matrix. For user-based similarity matrix, another user similarity matrix is calculated based on user demographic information including gender and occupation using Cosine Similarity method. Combining the similarity matrixes from step one and the new user similarity matrix, a new movie rating prediction matrix is calculated. And the error is calculated with Root-Mean-Square Error method. In this implementation, weight v_d (weight for user demographic information similarity matrix) in range (0, 1, 0.1) has been tested to demonstrate the impact of adding user demographic information into similarity matrix. For content-based similarity matrix, another movie similarity matrix is calculated based on only the movie genres information using Cosine Similarity method. Combing this matrix with the content-based movie similarity matrix from step one. A weight v_g representing the weight of this new matrix in range (0, 1, 0.1) is assigned when combining the matrix to demonstrate the impact of addition movie genres information.

4. Result

The goal of the implemented prediction model is to recommend a list of movies to a user. For this model, three methods are used to predict the movie ratings given by a specific user and the difference between the predicted ratings and the actual ratings for the same movie is noted as the error. The result is the overall Root-Mean-Square Error for all users and related movie ratings in the test dataset. The results for user-based, content-based and SVD methods are shown below:

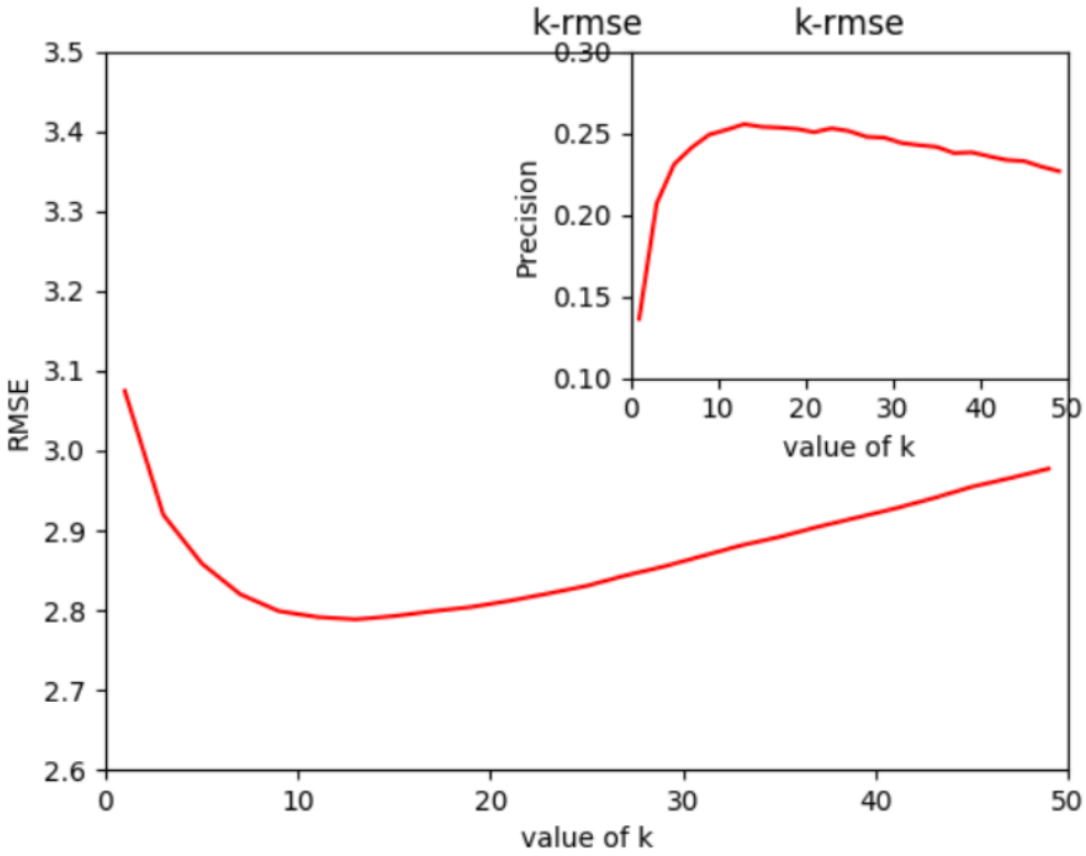
	Root-Mean-Square Error
User-based similarity	3.1406
User-based similarity with KNN	3.0244

Content-based similarity	3.4865
SVD (best result when k = 12)	2.7885

For User-based similarity method, adding the constrain of K Nearest Neighbors reduces the error measure by approximately 4%. This indicates less similar users' rating information is less relative and can be treated as useless information. This also brings out the impact of demographic information which will be analyzed later in this report.

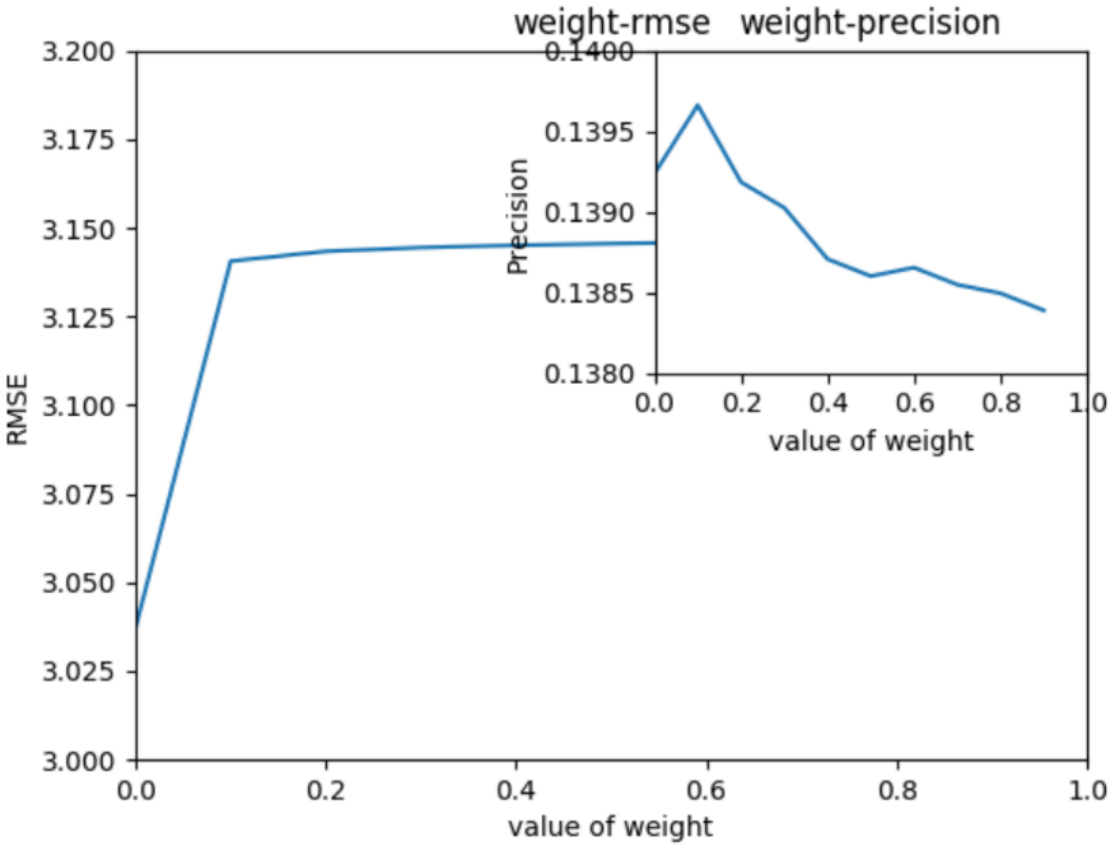
For Content-based similarity method, the error measure is greater than the error measure of User-based similarity method. This is a strong indication that different users have different tastes in term of movie rating. A user could give a movie rating that is very different from the average rating for that movie.

SVD method provides the best result comparing to user-based similarity method and content-based similarity method. However, the result of SVD method is largely depending on the k value chosen. For different k value used in the SVD method, the result is shown as:



Increasing the k value has positive effect at the beginning of this simulation. However, when k reaches 12, futher increasing of the value brings negative effect. This model gives the worst result when k is 1 where most of the information in the similarity matrix are ignored. When comparing the result of this method again user-based similarity method and content-based similarity method, SVD method always give better result in terms of Root-Mean-Square Error.

When user demographic information is added into the user-based similarity matrix, the result is shown in the following graph:

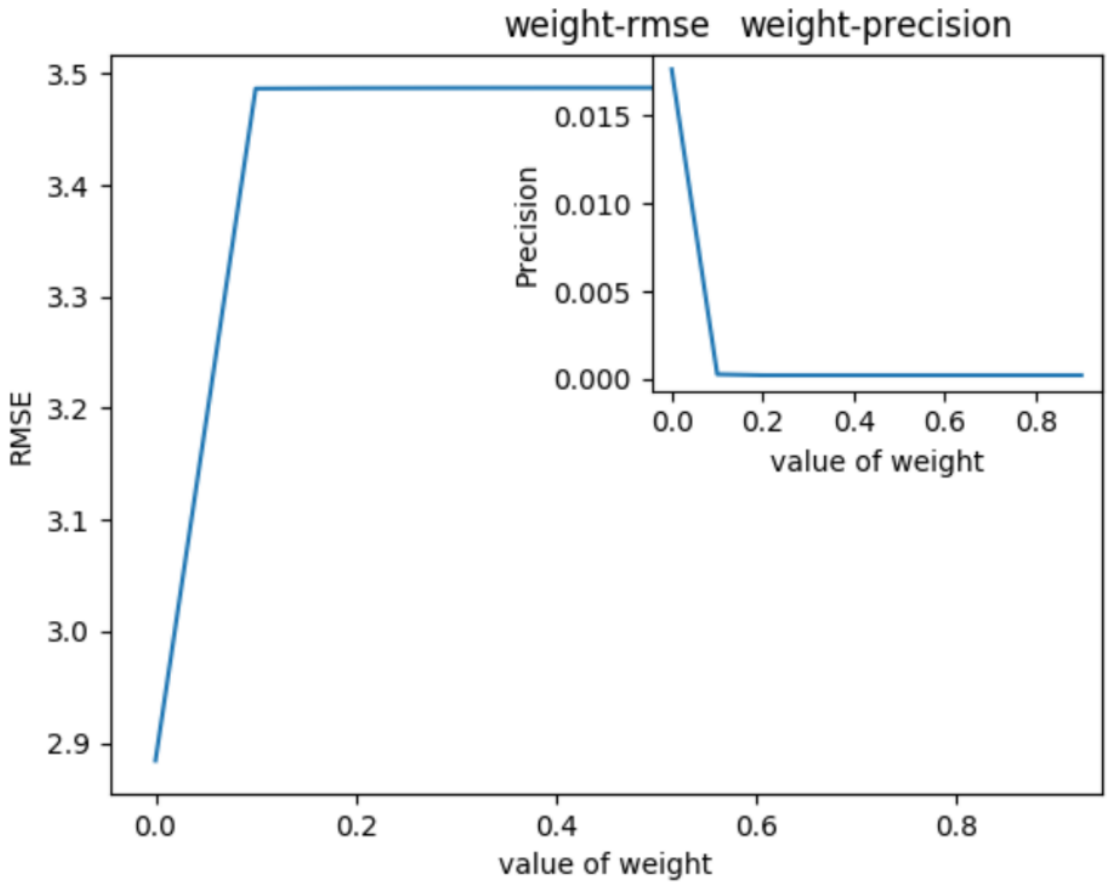


The model gives the best result when the weight of user-based similarity for movie rating is 0 and the weight of user-based similarity for demographic information is 1. However, the results difference between different weights settings are not significant. Once the weight of user-based similarity for movie rating reaches 0.1, this matrix will become the dominant component of the model.

On another hand, the model precision gives the bets result when the weight of user-based similarity for movie rating is 0.1 and the weight of user-based similarity for demographic information is 0.9. Increasing the weight of weight of user-based similarity for movie rating will significantly decrease the model precision.

The inclusion of user demographic information provides much better result in term of precision without significantly affecting the result in term of Root-Mean-Square Error.

Since the data only contains the user's rating on the move, it is necessary to select features to determine whether the predication accuracy of the model has influence under the influence of different features and use the filleter search to select it. By searching the data source, considering the category of the movie, and the age of the user may have an impact on the prediction results and ratings, these features are extended and given a certain weight. When movie genres information is added into the content-based similarity matrix, the result is shown in the following graph:



The model gives the best result when the weight of content-based similarity for movie rating is 0 and the weight of content-based similarity for movie genres information is 1. This result implies that the content-based similarity model performs the best when considering only the movie genres. Once the weight of content-based similarity for movie rating reaches 0.1, the movie genres information becomes redundant and the movie rating information will dominant the overall performance of this prediction model. In addition, the overall result will be much worse comparing the result using only movie genres information. The model precision shows the same pattern as Root-Mean-Square Error. The model is most accurate when using only the movie genres information.

5. Conclusion

This prediction model is designed to recommend a list of movies to a user based on user rating information, user demographic information and movie information. Three different methods are tested including user-based similarity method, content-based similarity method and SVD method. In conclusion, SVD method is more suitable to construct a prediction model for movie recommendation. This model provides better results in terms of both model Root-Mean-Square Error and model precision. The user-based similarity method provides the second-best result in terms of both model Root-Mean-Square Error and model precision. The content-based similarity method provides the worst result for this prediction model and this specific dataset.

6. Future Work

This assignment has tested three different methods to build collaborate filtering recommendation system. The dataset used in this assignment was originally collected in 1999. During that time, only limited information such as basic personal information was available to collect. However, in the year of 2019, there are much more information flying on the internet. As shown by the result, adding additional personal information such as occupation could possibly increase the performance of prediction model. A new model could be designed with more personal information such as social network activities. Gulati and Eirinaki (2019) tested a recommendation system with the additional social neighborhoods influence information. The result clearly indicates adding the social network information could improve the prediction model's performance. Future studies can be conducted on more social network information, demographic information and possibly geographic information. Naz, Maqsood and Durani (2019) provided a hybrid kernel mapping solution that combining User-Based method and Content-Based method together. This solution reduces some common problems associated with recommendation system such as cold start and data sparsity. Based on this idea, a more complicated hybrid kernel method that involves more fundamental recommendation system algorithms.

Reference

Gullati, A. & Eirinaki, M. 2019, 'With a Little Help from My Friends (and Their Friends): Influence Neighborhoods for Social Recommendations', in *Proceedings of The World Wide Web Conference*, San Francisco, CA, USA, May 13-17, 2019, pp. 2778-2784.

Herlocker, J. L., Konstan, J. A., Borchers, A. & Riedl, J. 1999, 'An algorithmic framework for performing collaborative filtering', in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 1999, pp. 230-237.

Howe, A. E. & Forbes, R. D. 2008, 'Re-considering neighborhood-based collaborative filtering parameters in the context of new data', in *Proceedings of the 17th ACM conference on Information and Knowledge management*, Napa Valley, California, USA, October 26-30, 2008, pp. 1481-1482.

Naz, S., Maqsood, M., & Durani, M. Y. 2019, 'An efficient Algorithm for Recommender System Using Kernel Mapping Techniques', in *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, Penang, Malaysia, Feb 19-21, pp. 115-119.