Implementation of User-based Recommendation Algorithms for Music Application

Authors: Chen Qian[1]

[1] Department of Engineering Systems and Environment, University of Virginia, 22904, VA, United States.

Acknowledgements:

This study was for University of Virginia SYS 6014 Decision Analysis, Spring 2020.

Abstract: In recent years, digital music has become the mainstream consumer content sought by many young people. However, how to help users quickly and accurately obtain music tracks that users are interested in presents difficulties. Firstly, with playlist data acquired from a famous Chinese music application, this study employs a lot of user-based collaborative filtering algorithms, such as Singular value decomposition (SVD), k-nearest neighbors (k-NN), and so on. Then some accuracy metrics (including Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE)) are selected to evaluate the performance of different models. Finally, the performances of different models are summarized and the payoffs of music recommendation system proposed in this study are discussed according to R-squared (R^2). Based on this study, users could save time on searching songs that they like and enterprise could attracts and retains customers so as to make a profit.

***Keywords*:** Digital Music, User-based Collaborative Filtering Algorithms, Recommendation System

1. Introduction

With the development of information technology and the Internet, people have gradually entered the era of information overload from an era of lack of information. Information consumers want to easily find the content they are interested in, and information producers want to push their content to the most suitable place. How to distinguish target audience now turned into a problem.

The recommendation system learns the user's preference information and the relationship between the user and the items and provides the user with items he or she may be interested in. As an effective method to solve information overload, it is increasingly used in various fields, such as book recommendation, music recommendation, and online shopping. Recommendation system plays an important role in the business of Amazon, Google, Netflix, and other companies.

Research on recommendation systems originated in the fields of cognitive science, statistics, information retrieval, and communication. For example, in 1979, Rich proposed a system Grundy, which uses a small amount of useful information to build user models, and then recommend novels that users may be of interest based on these models [1].

In the mid of 1990s, researchers began to use user rating information on used items to predict user ratings on unused items, and then recommend higher-rated items to users. Since then, the recommendation system has developed rapidly as an independent field of research. For example, in 1994, Resnick proposed GroupLens, a system for filtering news articles based on a collaborative filtering method, which uses users' ratings on articles to predict user interest [2].

There are many methods that can be used to predict the user's rating of an item. Adomavicius classified the recommendation system into content-based recommendation algorithms and collaborative recommendation algorithms [3]:

Content-based algorithm recommends items that are similar to items users liked in the past. It was mainly used for text recommendation in the early period. For example, Balabanovic described a Fab system for web page recommendation [4]. He uses the 100 most important keywords to represent a web page, and calculates the similarity between web pages based on these keywords, and then recommends. In addition, there are methods such as clustering, decision trees, and neural networks. These methods learn and train a model from the relevant data of items, and make recommendations based on this. For example, Pazzani used the Bayesian classifier to recommend web pages [5].

Collaborative algorithm recommendas to the user the items of other users that are similar to his interests. In Breese's study, collaborative algorithms are classified into memory-based and model-based [6]. The memory-based collaborative algorithm uses all users' ratings to predict items; the model-based collaborative algorithm trains a model based on historical rating data. Sometimes it is not possible to get a clear rating from the user. Hu et al. proposed a method to use implicit feedback, that is, the user's behavioral trace to predict the user's preference for the item [7].

This study would explore the application of user-based collaborative filtering algorithm on music recommendation system. This study introduces a lot of models to predict users' rating on different songs. A comparison is conducted between the proposed algorithms by selecting different accuracy metrics. In this study, the payoffs of using music recommendation system would be discussed.

2. Data

2.1 Data Collection

The original data are crawled from NetEase Cloud Music (https://music.163.com/) and stored as json file. The json file is about 16GB which is hard to deal with, this means it should be imported and saved as csv file, which is easy to deal with. This study extracts 5 useful features: playlist_id, playlist_name, song_id, song_name, and popularity. Because the limitation on computing abilily of the computer, this study ultilized the first 500,000 rows data as data set, which is enough for future training.

2.2 Data Process

In the modeling process of the recommendation system, this study employs the Python library Surprise (a simple Python recommendation system engine, http://surpriselib.com/), which is one of the scikit series. The supporting format of data for Surprise is user, item, and rating. This study drops missing value directly and the data set could be shown as Table 1.

Table 1. Music data set

	playlist_id	**song_id**	**popularity**
0	423245641	414691355	80
1	423245641	410802620	100
2	423245641	419549837	60
3	423245641	419485281	45
4	423245641	412016420	65
5	423245641	421160284	85
⋮	⋮	⋮	⋮

Additionally, the information of playlist_id to playlist_name and song_id to song_name is stored as Table 2 and Table 3.

Table 2. Playlist_id to playlist_name

	playlist_id	playlist_name
0	445959714	Over The Horizon-SAMSUNG GALAXY THEME
1	370389263	DeadWeight
2	363476047	Popping
3	707536911	farrux
4	553382471	Our Favorite Pop
5	473141053	Michael Jackson ' selection
⋮	⋮	⋮

Table 3. Songlist_id to songlist_name

	song_id	song_name
0	414691355	Lost (As I Am)
1	410802620	Next Escape
2	419549837	Silhouette
3	419485281	Feel My Love
4	412016420	Hit It
5	421160284	Catch U
⋮	⋮	⋮

Then, playlist_id, song_id, and popularity would be treated as user, item, and rating. There is a normolization on Popularity and rescale it from 1 to 5, which is convinient for future training. In this way, popularity is transformed into rating and the distribution of user's rankings is shown in Figure 1.

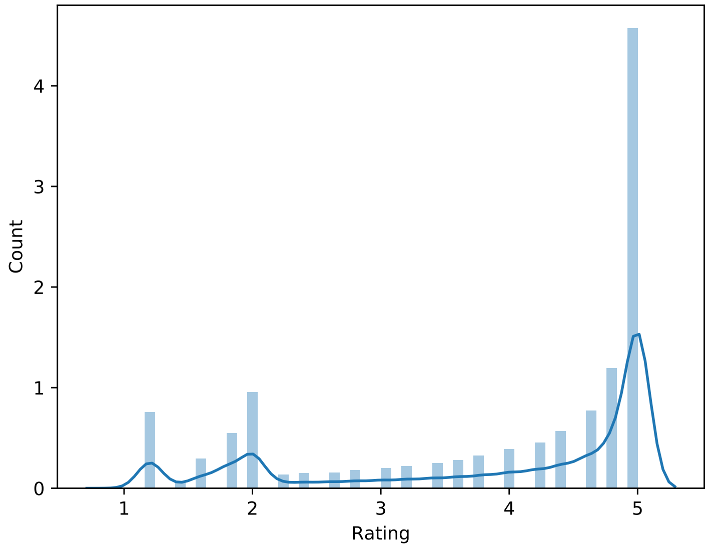

Figure 1. Distribution of users' rankings

3. Methodology

The workflow of this study, as shown in Figure 2, can be summarized as follows: (1) The user data set is processed as the format of user, item, and raring; (2) A lot of user-based collaborative filtering algorithms are used to build the model; (3) These algorithms are compared based on Mean Squared Error (RMSE) and Mean Absolute Error(MAE). And the payoffs of music recommendation system are evaluated based on R-squared.

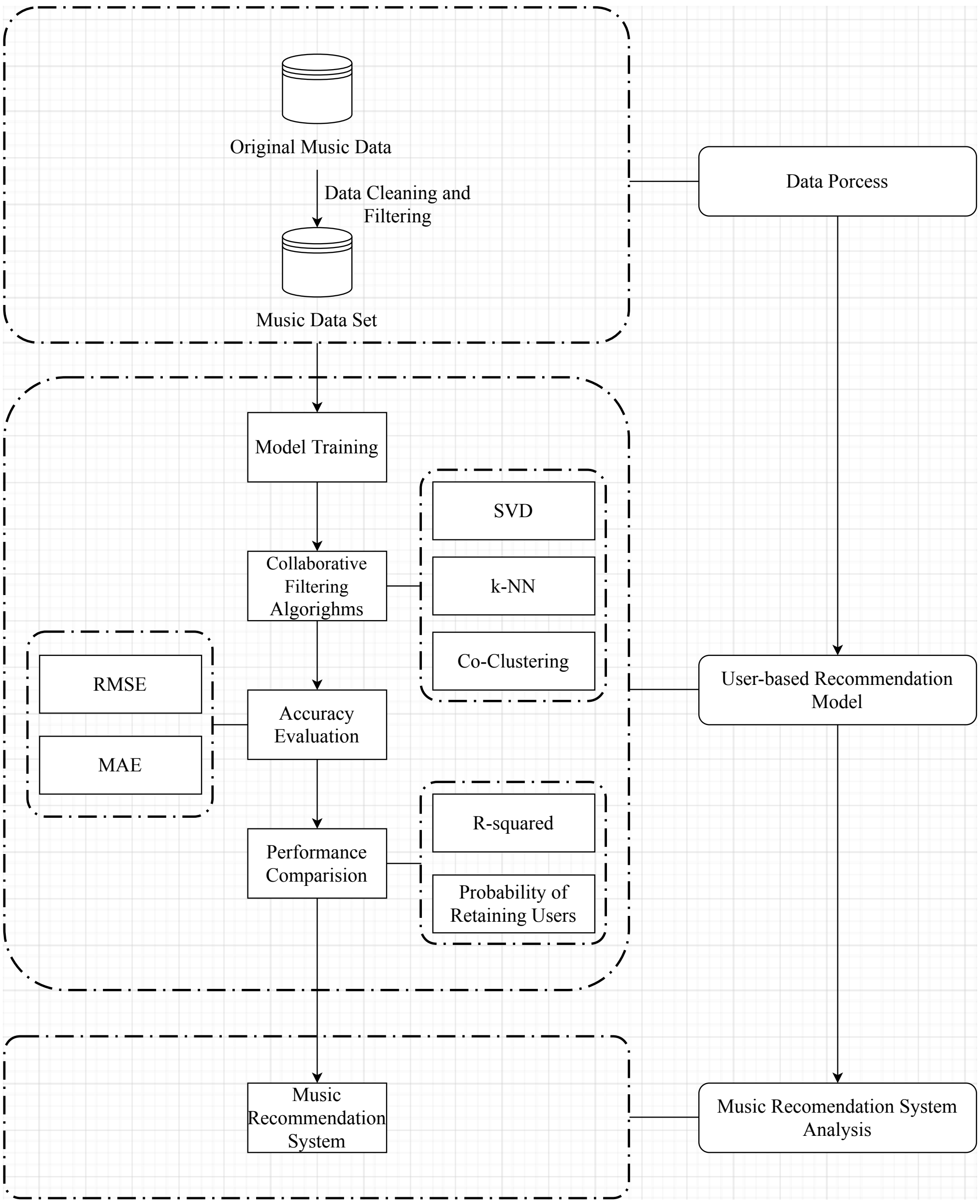

Figure 2. The workflow of implementation of user-based recommendation algorithms for music application

3.1 User-based Recommendation Algorithms

3.1.1 Sigular value decomposition

Singular value decomposition (SVD) is a famous algorithm [8], which is popularized by Simon Funk during the Netflix Prize.

The prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \tag{1}$$

If user $\hat{r}_{ui}$ is unknown, then the bias b_u and the factors p_u are assumed to be zero. The same applies for item i with b_i and q_i.

To estimate all the unknown, the following regularized squared error is needed to be minimized:

$$\sum_{r_{ui} \in R_{\text{train}}} (r_{ui} - \hat{r}_{ui})^2 + \lambda\left(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2\right) \tag{2}$$

The minimization is performed by a very straightforward stochastic gradient descent:

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u) \tag{3}$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i) \tag{4}$$

$$p_u \leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u) \tag{5}$$

$$q_i \leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda q_i) \tag{6}$$

where $e_{ui} = r_{ui} - \hat{r}_{ui}$.

3.1.2 K-nearest neighbors

K-nearest neighbors (k-NN) is a basic R [9].

For basic k-NN, the prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} \text{sim}\,(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}\,(u, v)} \tag{7}$$

For centered k-NN, the prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - \mu_v)}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)} \tag{8}$$

K-NN Baseline takes into account a baseline rating [10], the prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)} \tag{9}$$

3.1.3 Co-clustering

Co-clustering is a set of techniques in Cluster Analysis [11]. Basically, users and items are assigned some clusters C_u, C_i, and some co-clusters C_{ui}. The prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \overline{C_{ui}} + \left(\mu_u - \overline{C_u}\right) + \left(\mu_i - \overline{C_i}\right) \tag{10}$$

where $\overline{C_{ui}}$ is the average rating of co-cluster C_{ui}, $\overline{C_u}$ is the average rating of u's cluster, and $\overline{C_i}$ is the average rating of i's cluster.

3.1.4 Random Prediction

Random algorithm predicts a random rating based on the distribution of the training set, which is assumed to be normal. The prediction $\hat{r}_{ui}$ is generated from a normal distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ where $\hat{\mu}$ and $\hat{\sigma}$ are estimated from the training data using Maximum Likelihood Estimation:

$$\hat{\mu} = \frac{1}{|R_{\text{train}}|} \sum_{r_{\text{ui}} \in R_{\text{train}}} r_{ui} \tag{11}$$

$$\hat{\sigma} = \sqrt{\sum_{r_{\text{ui}} \in R_{\text{train}}} \frac{(r_{ui} - \hat{\mu})^2}{|R_{\text{train}}|}} \tag{12}$$

This algorithm is used to be a baseline algorithm.

3.2 Accuracy Metrics

3.2.1 Root Mean Squared Error

The root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed [12].

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\hat{R}|}\sum_{\hat{r}_{ui}\in\hat{R}}(r_{ui}-\hat{r}_{ui})^2} \quad (13)$$

3.2.2 Root Mean Squared Error

The mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon [13].

$$\mathrm{MAE} = \frac{1}{|\hat{R}|}\sum_{\hat{r}_{u}\in\hat{R}}|r_{ui}-\hat{r}_{ui}| \quad (14)$$

3.3 Evaluation on Payoffs

3.3.1 R-squared

R-squared (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model [14]. Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable. In this study, R-squared is used to evluate the payoffs of applying this music recomendation system.

$$\mathrm{R}^2 = 1 - \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{y_i}-\mathrm{y_{pred}})^2}{\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{y_i}-\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{y_i})^2} \quad (15)$$

3.3.2 Payoffs Calculation

The original probability of retaining a user successfully is assumed as 0.5. Based on the result of different algorithms, this probability would be calculated and renewed as follows:

$$\mathrm{P} = 0.5 \times (1 + \mathrm{R}^2) \quad (16)$$

4. Results

4.1 Comparation on different algorithms

The result of different algorithms is shown in Table 4.

Table 4. Result of comparation

	Model	RMSE	MAE	R_Squared
0	SVD	0.8152	0.5790	0.6256
1	k-NN	1.0574	0.5870	0.3658
2	Centered k-NN	1.1191	0.7378	0.2946
3	k-NN Baseline	0.7430	0.4813	0.6888
4	Co-Clustering	1.1453	0.7625	0.2746
5	Random	1.7211	1.3606	-0.6757
⋮		⋮	⋮	⋮

According to this result, we could know that k-NN Baseline performed best tin this study. Then we would use k-NN Baseline algorithm to make music recommendation.

4.2 Result of the music recomendation system

In this part, k-NN Baseline algorithm is empolyed to build the music recommendation model. The top 5 similar playlists to (363476047, Popping) are shown in Table 5.

Table 5. Top 5 similar playlist to (363476047, Popping)

	playlist_id	playlist_name
0	626822582	DVLM - Bringing The Madness
1	121241848	Dancing in the room, do da poppin like this!
2	437097435	Crazy Frog
3	119356097	Over 10,000 song rankings
4	119474296	2016 European and American annual best new orders
⋮	⋮	⋮

It shows that the top 3 similar playlists have a extremely closing theme to Poping, this means our music recommendation is relatively accurate. In this study, the playlist could be approximately thought as user, then the personalize recommendation could be offered to users.

4.3 Payoffs of music recommendation system

Based on a good music recommendation system, users are likely to retain and continue using the music application. The result of music recommendation system is shown is Table 6.

Table 6. Payoffs of music recommendation system

	Recommendation	**Probability of retaining users**
0	None	0.5000
1	Random	0.1622
2	k-NN Baseline	0.8444
⋮	⋮	⋮

It shows that the music recommendation system could get 68.88% improvement than a music system without recommendation. And comparing to a random recommendation system, an appropriate music recommendation could get an improvement of 420.59%.

5. Discussion & Conclusion

This study implements a music recommendation system based on k-NN algorighm. The data is craweled from a famous Chinese music application. During the process of building music recommendation model. Some famous collaboritive filting algorithms are compared and k-NN Baseline could get the best performance. From the recommendation result, the veracity and reliablity of the music recommendation system is proved. Moreover, the music recommendation system has a generous payoffs comparing to a random recommendation system and music system without recommendation.

Despite the result of this music recommendation system is good enough to apply to offer recommendations to users. The performance of the user-based collaborative algorithms could be improved further based on a hybrid approach combing content-based recommendation algorithms and user-based collaborative algorithms. Future work would explore the possibility of designing more advanced and effective algorithms.

References:

[1] E. Rich, "User modeling via stereotypes," *Cognitive science*, vol. 3, no. 4, pp. 329–354, 1979.

[2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, 1994, pp. 175–186.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[4] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[5] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine learning*, vol. 27, no. 3, pp. 313–331, 1997.

[6] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *arXiv preprint arXiv:1301.7363*, 2013.

[7] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 eighth IEEE international conference on data mining*, 2008, pp. 263–272.
[8] B. Robin, "Hybrid recommender systems: Survey and experiments, user modelingand user-adapted interaction," 2012.
[9] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
[10] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, pp. 1–24, 2010.
[11] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *Fifth IEEE international conference on data mining (ICDM'05)*, 2005, pp. 4-pp.
[12] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
[13] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.
[14] J. Miles, "R squared, adjusted R squared," *Wiley StatsRef: Statistics Reference Online*, 2014.