Recommending More Suitable Music Based on Users' Real Context

Qing Yang[1], Le Zhan[1], Li Han[1], Jingwei Zhang[2,3(✉)], and Zhongqin Bi[4]

[1] School of Electronic Engineering and Automation,
Guilin University of Electronic Technology, Guilin 541004, China
gtyqing@hotmail.com, 1217093432@qq.com, 1007060417@qq.com
[2] Guangxi Cooperative Innovation Center of Cloud Computing and Big Data,
Guilin University of Electronic Technology, Guilin 541004, China
gtzjw@hotmail.com
[3] Guangxi Key Laboratory of Trusted Software,
Guilin University of Electronic Technology, Guilin 541004, China
[4] College of Computer Science and Technology,
Shanghai University of Electronic Power, Shanghai 200090, China
zqbi@shiep.edu.cn

Abstract. Music recommendation is an popular function for personalized services and smart applications since it focuses on discovering users' leisure preference. The traditional music recommendation strategy captured users' music preference by analyzing their historical behaviors to conduct personalized recommendation. However, users' current states, such as in busy working or in a leisure travel, etc., have an important influence on their music enjoyment. Usually, those existing methods only focus on pushing their favorite music to users, which may be not the most suitable for current scenarios. Users' current states should be taken into account to make more perfect music recommendation. Considering the above problem, this paper proposes a music recommendation method by considering both users' current states and their historical behaviors. First, a feature selection process based on ReliefF method is applied to discover the optimal features for the following recommendation. Second, we construct different feature groups according to the feature weights and introduce Naive Bayes model and Adaboost algorithm to train these feature groups, which will output a base classifier for each feature group. Finally, a majority voting strategy decides the optimal music type and each user will be recommended more suitable music based on their current context. The experiments on the real datasets show the effectiveness of the proposed method.

Keywords: Music recommendation · Feature selection · User context

1 Introduction

A large number of music existing in different web sites and users' implicit music preferences have presented a big challenge for lean and tailored user services.

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 124–137, 2019.
https://doi.org/10.1007/978-3-030-12981-1_8

Music recommendation is an important mechanism to provide personalized services for users. Two parts are key to ensure the accuracy of music recommendation, one is how to discover the effective features for recommendations, and the other is the matching strategy between user preferences and different kinds of music. Currently, the popular recommendation strategy for music is to identify users with similar preferences and then to apply collaborative methods for recommendation, which is based on those collected information, for example, which kinds of music users have browsed and listened, namely users' behaviors. It is a good idea to generate the favorite music by applying users' historical behaviors to conclude users' implicit preference, but considering the real user context, such as users' current working state, the most favorite music may be not the most suitable. From the viewpoint of recommendation overfitting and diversification, the recommendation covering users' current context will be more popular and practical, which can help to dig out users' new music preferences.

Collaborative filtering is the most popular recommendation strategy, including user-based collaborative filtering and item-based collaborative filtering, the former focuses on finding a group of similar users and the latter focuses on finding a group of similar items. In fact, collaborative filtering is a transformation from group behaviors to individual characteristics, in which a user or an item is modelled into a vector to show those behaviors that a user has done on items. Obviously, the above model is based on a series of historical behaviors and are more prone to mine the items or users that should be recommended comprehensively, which does not serve for the current context and is also tardy to capture users' interest migration. In order to improve the recommendation effectiveness, this paper proposes the recommendation model covering users' real context and designs a recommendation method based on Naive Bayes model and the Adaboost algorithm. Both users' historical behaviors on music and their current states, including emotional state and working state, etc., are integrated to describe users, and a feature selection process based on the ReliefF algorithm is conducted before recommendation to filter those optimal features for classification training. The Naive Bayes model is introduced to train base classifiers. The Adaboost ensembling algorithm works for integrating each base classifier and recommending those items by a relative majority voting strategy.

This study is organized as following. Section 2 summarizes the work related with music recommendation. Section 3 defines the music recommendation problem serving for users' real context. Section 4 details the proposed recommendation model, including the feature selection and the recommended item generation. Section 5 designs experiments to verify and analyze our method. Section 6 concludes the whole study and discusses the future work.

2 Related Work

With the rapid expansion of digital music and users' diversified amusement requirements, personalized music recommendation has become an important

research topic in the fields of recommendation. Compared with recommendation for e-commence, etc., personalized music recommendation are more complex since it not only pays attention on the closeness between user preferences decided by users' historical behavior and music labels or music content, but also is affected by users' real context since one user could like enjoy different styles of music when he/she is in different scenarios. Currently, mainstream recommendation methods include automatical play list generation [14], content-based recommendations [2], collaborative-filter methods [12], context-based methods [19], and hybrid recommendations [18].

Automatical play list generation focuses on those songs that are similar to the chosen seeds to generate a new play list. Baccigalupo [1] presented a Case Base Reuse(CBR) approach to establish a new play list. CBR system retrieves those most relevant music from the Case Base, and then combines them to generate a new play list, in which music is ranked by its relevance to the pre-specified music.

Collaborative filtering methods recommend pieces of music to a user based on music rating, which are contributed by other users with similar taste. To address the data sparsity problem, Huo [9] applied stack denoising auto-encoder to construct content-based model and then proposed deep learning to cooperate with collaborative filtering. Sarwar [15] experimentally evaluated that item-based collaborative filtering can produce high-quality recommendations. Melville [13] put forward an effective framework for combining content-based method and collaborative filtering, which uses a content-based predictor to enhance existing user data and then provides personalized suggestions through collaborative filtering.

Content-based methods compute similarity between songs, and then recommend songs similar to those known favorite songs. Xing [21] proposed and examined a novel approach to generate latent embeddings for podcast items, which utilizes the aggregated information from the text-based features related with the audio items. Kuo [11] presented a personalized content-based music filtering system to support music recommendation based on user preference on melody style. Further, emotional information have also presented their influence on recommendation [10].

Context-based methods take context into consideration, which include time, place, emotion and so on. Gu [8] put forward a context aware matrix factorization model, named AlphaMF, to tackle with the cold start, which uses the matrix factorization for modelling implicit feedback and introduces the linear contextual features for modelling explicit contextual feedback. Wu [20] proposed a context feature auto-encoding algorithm based on regression tree, which can only deal with numerical features. Trajectory data and location information have also been widely considered for recommendation [4,5,23,25].

Hybrid approaches, which consider both music content and other related information for recommendation, are being paid more attention. Yoshii [24] integrated both rating and music content data by a Bayes network to realize recommendation. Donaldson [6] leveraged spectral graph properties of an item-base collaborative filtering as well as acoustic features of the music signal for

recommendation. For other recommendation applications, [26] designed a distributed storage and query system for optimizing POI recommendations based on location-constraints. [22] designed a stacked denoising autoencoder model for preprocessing recommendation data and then to improve recommendation performance. Group recommendations are also a popular topic. [16] and [17] studied the problem of the recommendation fairness for package to-group recommendations, which also provided an approximate solutions in reasonable time. [7] proposed a new type of group recommendation, namely personalized recommending event-based groups to users, in which both the linear model for capturing explicit features and matrix factorization model for mining past interactions between users and groups are exploited. [27] designed a Nash equilibrium based item group recommendation approach, in which consumers' preferences for an item group are evaluated from two perspectives, namely the attraction from these customers themselves and the social affection from their friends. [3] conducted extensive experiments to evaluate the influence of recommending accuracy from the rating prediction methods, which shows that the rating prediction for individual users are more effective than for groups on both improving recommendation accuracy and reducing the influence from data sparsity.

3 Problem Definition and Recommendation Model

In this section, music recommendation covering user context is given a formal statement in Definition 1. Music categories are initialized for recommendation, several kinds of user features are considered for matching music categories. The current user context includes users' primary attributes, users' historical behaviors, their current emotional state, etc.

Definition 1. ***Music recommendation covering user context.*** *Given a set of music I, a set of music categories C, and a group of users U. Each element $i \in I$ belongs to a subset of P, which is denoted as $i \in c \wedge c \in P \wedge P \subseteq C$. Each element $u \in U$ is described by a series of features, which are namely user context. A user sample can be denoted as a $(l+1)$-tuple, such as $(f_1, f_2, \cdots, f_l, c)$, l is the number of user features and c is to indicate the corresponding music categories that i would like in current context. Music recommendation covering user context tries to find the optimal music categories based on the context of a specific user i and then to recommend the popular music in each found categories to this user.*

Figure 1 presents the music recommendation model covering user context. Figure 1(a) corresponds to U, whose feature set are filtered from $l+1$ into $m+1$ and are presented in Fig. 1(b), c denotes one music category in both Fig. 1(a) and (b). An initial feature selection can help to reserve those features with greater weights. A series of operations will work on the transformed user set, which include enumerating features, learning Bayes classifier, generating recommendation rules, ranking recommended items, etc. Enumerating features tries to find the optimal feature group for the following classification. Here, classifiers exploit the user context to create recommendation rules. A new user can depend on

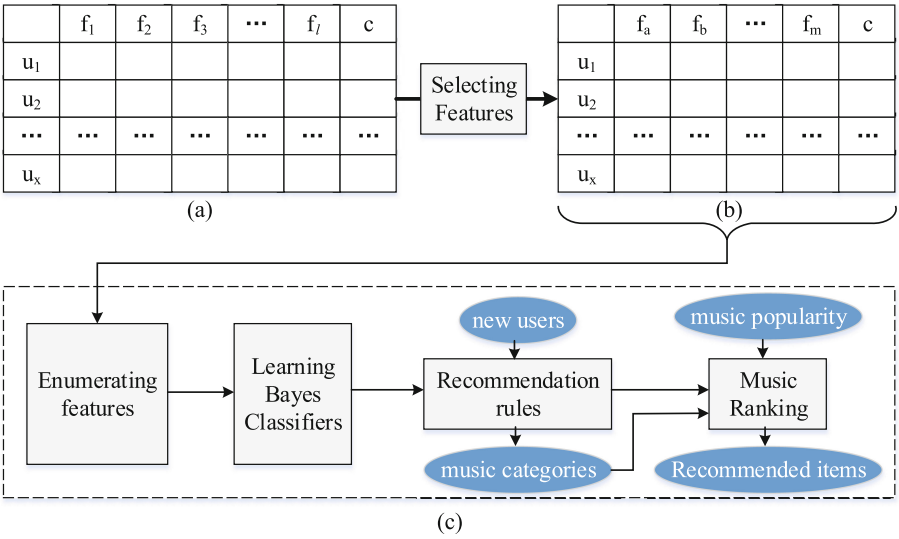

Fig. 1. Music recommendation model

the recommendation rules to obtain his/her related music categories, and then those music in each output category can be recommended to this user on their popularity. The detailed process is presented in Fig. 1(c).

4 Proposed Method

As in Sect. 3, a user u_1 can be denoted as a n-dimensional vector. In order to discover those music categories that u_1 might like on u_1's current states, we design a recommendation rule generation strategy by Bayes classification to find the relationships between users' features and music categories. Those generated recommendation rules can guide to capture both the music categories and the corresponding music that should be recommended to users. Here, we firstly choose the optimal features that are more effective to match users with their favorite music categories. Since music categories need to be represented by multiple labels, we introduce the ReliefF algorithm for the multi-label classification.

4.1 Selecting Features for Ensembling Classification Strategy

Supposing the given data set containing m features and l labels, R^m is the sample feature space and R^l is the sample label space, and the training set can be denoted as $TD = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, $x_i \in R^m, y_i \in R^l$. l is the number of the labels corresponding to music categories. Each feature has a weight, W_A, to indicate its importance for classification, whose updating formula is presented in Formula 7.

$$W_A = \sum_{k=1}^{K} diff(A, T, H_k)\frac{1}{jk} + \sum_{C \notin class(T)} [\frac{p(C)}{1 - p(class(T))}]diff(A, T, M_k(C))\frac{1}{jk} \tag{1}$$

Here, A is a specific feature, T is a random sample from TD and K denotes the number of T's nearest neighbors. The function, $diff(A, T, H_k)$, is defined in Formula 2 and is responsible for computing the difference between T and the k_{th} nearest neighbor, H_k, which have the same label. $class(T)$ returns the label of T. $M_k(C)$ denotes the k_{th} sample with the label C. $p(C)$ denotes the probability of those samples with the label C. $p(class(R))$ is the probability of the label of sample R. j is just the sampling times. Given a threshold δ, if $W_A > \delta$, the feature A will be retained, otherwise A is removed.

$$diff(A, T, H_k) = \frac{T(A) - H_k(A)}{Max(W_A) - Min(W_A)} \tag{2}$$

Based on the Formula 1, $N(N < M)$ features that are optimal for classification will be obtained. Because each feature can contribute different influence to the classification based on those labels, we try to divide these features into several feature groups on their weight ranking. Namely, the first feature group consists of only the feature with highest weight, the second feature group consists of the features ranked the 1st and 2nd on the weights, etc. Each feature group is inserted a new features that have the highest weight and have not existing in any feature group. For one specific feature group, each feature is considered to be independent and the Bayesian classifier is introduced for classification. Depending on the conditional independence assumption, we have Formula 3, in which S corresponding to the cardinality of one feature group. The posterior probability of every label can be computed as Formula 4.

$$p(X = x|Y = y_i) = \prod_{s=1}^{S} p(X^{(s)} = x^{(s)}|Y = y_i) \tag{3}$$

$$p(Y = y_i|X = x) = \frac{p(X = x|Y = y_i)p(y = y_i)}{\sum_{i=1}^{l} p(X = x|Y = y_i)p(y = y_i)} \tag{4}$$

Since the probability of (x, y_i) can be denoted as $p(Y = y_i|X = x)$, we define the maximum label probability of x_i, G_q, as Formula 5.

$$G_q(x_i) = argmax\ p(Y = y_i) \prod_{s=1}^{S} p(X^{(s)} = x^{(s)}|Y = y_i) \tag{5}$$

Then, we can calculate the error rate of classificationerr_{qi} as Formula 6, $G_q(x)$ is the classification result. If $err_{qi} > 0.5$, the feature weight distribution can be updated as Formula 7.

$$err_{qi} = \sum_{i=1}^{N} w_{qi} I(G_q(x_i) \neq y_i) \tag{6}$$

$$w_{(q+1,i)} = \frac{w_{qi}}{Z} \exp\left(-\alpha_q y_i G_q(x_i)\right), i = 1, 2, \ldots, N \tag{7}$$

α_q is the coefficient of the $G_q(x)$ and Z is normalization factor that is defined in Formula 8.

$$Z = \sum_{i=1}^{N} w_{qi} \exp\left(-\alpha_q y_i G_q(x_i)\right) \tag{8}$$

The final classifier result is expressed as Formula 9.

$$G(x) = \sum_{q=1}^{Q} \alpha_q G_q(x) \tag{9}$$

4.2 Constructing Initial Recommendation Rules

The above outputs and divides these optimal features into several feature groups, and each feature group are applied to construct a classifier. Given a unclassified sample x^*, we can calculate the probability of each label for x^*. Supposing the current label is y_t, we can obtain the probability of x^* from all the classifiers contributed by the above feature groups, which is defined in Formula 10.

$$P_t(x^*) = \frac{1}{L} \sum_{l=1}^{L} I(G(x^*) = y_t) \tag{10}$$

L is the number of all feature groups and t is the number of labels. We only need to calculate the probability of different labels for x^* and to generate the most probable label for x^* by the majority voting strategy as defined in Formula 11.

$$Y = argmax(P_t(x^*)) \tag{11}$$

4.3 Music Ranking for Recommendation

Now, we can obtain an ordered sequence of labels on users' current context by those constructed Bayesian classifiers and the majority voting strategy. When a new user appears, we can pick the label with the highest score, namely the most suitable music category considering users' real states, to generate the specific music for this user. A list of music will be presented to this user on the music popularity in the chosen music categories, and the music popularity is contributed by the music websites (https://music.163.com/). The detailed recommendation process is presented in Algorithm 1.

5 Experiments

In this section, we designed and carried out experiments on real data sets to verify our proposed recommendation method. All experiments are run on a computer with dual-core CPU @1.90 GHZ and 4 GB memory, and all code is implemented in Python.

Algorithm 1. Music Recommendation Covering Users' Real Context

Input U_1: training dataset; U_2: test dataset; L: the number of feature groups; F: a set of features; J: the sampling times; δ: the weight threshold of the selected feature;
Ouput the labels corresponding to music categories.

set the weight of each feature to be 0, namely $W_A = 0$
for $j = 1, 2, ..., J$ **do**
  select a sample R randomly, $R \in U_1$
  applying Euclidean distance to select k nearest neighbor samples with the same label of R, denoted as H_k, and also select k nearest neighbor samples with different labels of R, denoted as M_k
  for $A = 1, 2, ..., N$ **do**
    update W_A according to Formula. 1
    if $W_A > \delta$ **then**
      add A to the feature set F
    end if
  end for
end for
divide F into L feature groups
for $i = 1, 2, ..., L$ **do**
  learn a classifier,$G_q(x)$, by the i_{th} feature group on U_1
  repeat
    apply $G_q(x)$ on U_2 and calculate the error rate err_{qi}
    if $err_{qi} \geq 0.5$ **then**
      update the weight of each feature by Formula. 7
    end if
  until $err_{qi} < 0.5$
  calculate the probability of each label by $P_t(x^*)$ and output the label with the highest probability
end for

5.1 Data Sets and Evaluation Metrics

In order to test the effectiveness of the proposed method, we constructed a data set through our designed questionnaire, which is completed by the friends in WeChat. The whole questionnaire is composed of two kinds of questions, one kind of questions are to let users answer their attributes and current states, the other kind of questions is to let users choose one music category under their current states. The whole data set includes 400 user records, 5 music categories, and all music are crawled from https://music.163.com/. Each user is described by 10 features, which are showed in Table 1.

Three metrics, precision, recall and F1-score are introduced for evaluating recommendation performance, which are presented in Formulas 12 to 14. **RecNum** is the number of the recommended music from our proposed model, **RightNum** is the cardinality of the music both recommended by our model and accepted by users. **ExpectedNum** corresponds to the number of all music that should be recommended.

Table 1. User features in data set

f_1	f_2	f_3	f_4	f_5
Emotional state	Times weekly	Character	Gender	Career
f_6	f_7	f_8	f_9	f_{10}
Instrument	Age	Income	History habit	Education

$$Precision = \frac{RightNum}{RecNum} \tag{12}$$

$$Recall = \frac{RightNum}{ExpectedNum} \tag{13}$$

$$F1\text{-}score = \frac{2 * RightNum}{ExpectedNum + RecNum} \tag{14}$$

5.2 Experimental Results and Analysis

Features are key for classification. For the given data set, all features are ranked by **reliefF** algorithm firstly. Then we try different group of features for classification to find the optimal feature group. The combination strategy of features is greedy. Namely, the feature with the greatest weight is considered for classification, and then the feature in the second place is added for classification. All features are considered in order of their weights until the classification accuracy decreases.

We use a 15 cross verification for ranking the features. The data set is divided into 15 parts randomly, for each stage, 14 parts are used for training and one part for testing. The weight of each feature is assigned by the average accuracy contributed by the feature. The detailed experimental results are presented in Table 2, and all weights of the features are presented in Fig. 2, in which **f1** to **f10** correspond to those features in turn in Table 1.

Finding the Optimal Combination of Features. We selected different number of features as input for each Bayes classifier. First, we sorted these features by their weights. Since the music enjoyed by users at different context may be constrained by different features, we set different number of feature groups for experimental verification. Second, we removed the last four features since their weights are small and constructed 5 feature groups, the first group is the combination of f_1 and f_2, the second group is the combination of f_1, f_2 and f_3, etc. Each group of features are used to test their classification performance individually to find the optimal feature group. We carried out the experiment to observe their classification performance of different feature groups, Fig. 3 presents the classification accuracy on different group of features. Obviously, when we consider the five features with the greatest weights, the classification shows the best

Table 2. Experimental results on feature weights

Running times	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
1	0.171	0.031	0.135	0.124	0.072	0.132	0.162	0.084	0.228	0.071
2	0.176	0.026	0.141	0.123	0.073	0.135	0.167	0.080	0.227	0.079
3	0.169	0.030	0.137	0.120	0.065	0.137	0.169	0.073	0.233	0.073
4	0.173	0.026	0.126	0.126	0.069	0.129	0.172	0.075	0.231	0.064
5	0.166	0.024	0.130	0.133	0.076	0.129	0.163	0.086	0.223	0.072
6	0.162	0.027	0.138	0.127	0.068	0.136	0.168	0.087	0.236	0.071
7	0.170	0.031	0.136	0.129	0.078	0.130	0.171	0.081	0.234	0.075
8	0.173	0.034	0.140	0.123	0.081	0.127	0.173	0.082	0.229	0.069
9	0.164	0.029	0.135	0.114	0.077	0.129	0.172	0.077	0.227	0.073
10	0.170	0.033	0.144	0.122	0.075	0.133	0.164	0.073	0.233	0.082
11	0.169	0.035	0.137	0.129	0.071	0.141	0.168	0.077	0.238	0.087
12	0.171	0.027	0.127	0.130	0.079	0.142	0.163	0.079	0.241	0.075
13	0.175	0.018	0.129	0.128	0.072	0.139	0.168	0.074	0.237	0.072
14	0.177	0.033	0.137	0.117	0.064	0.133	0.172	0.087	0.236	0.073
15	0.171	0.029	0.135	0.124	0.068	0.137	0.169	0.089	0.235	0.071

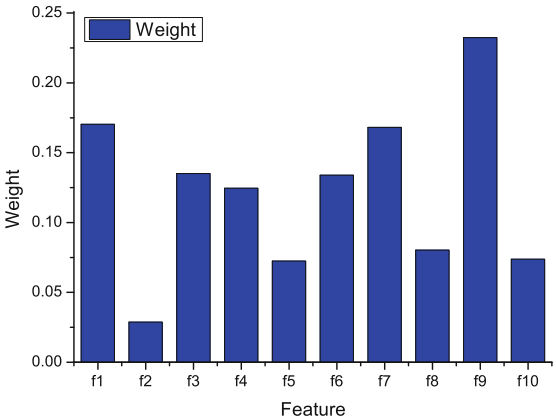

Fig. 2. The weight of each feature

performance. According to Fig. 2, the five features are f_9, f_1, f_7, f_3 and f_6, which will be applied for generating the following recommendation rules.

Generating the Recommended Music. According to the above experimental results, we use the five Bayesian classifiers corresponding to the five specific features to predict the potential music categories for new users. Each classifier

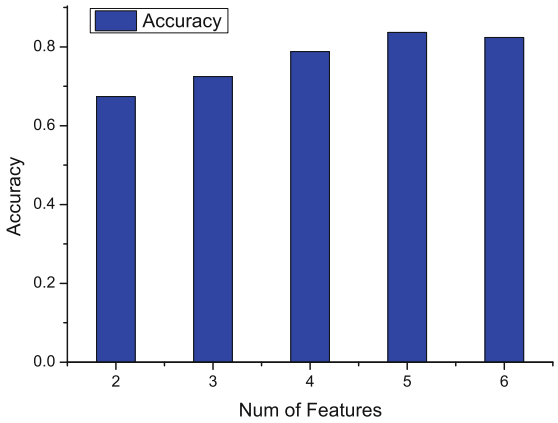

Fig. 3. Accuracy under different group of features.

can output one music category for each user, and we introduce the relative majority voting to conclude the optimal music categories. Those top-k popular music in the output music category are picked for users. Here, the number of the output music categories is set to 2, and k is set to 10. In order to test the effectiveness of the proposed method, we randomly choose the users from the data set and construct different subsets, the user number of each testing data set are 150, 200, 250,300, 350 and 400. The experimental results are presented in Fig. 4.

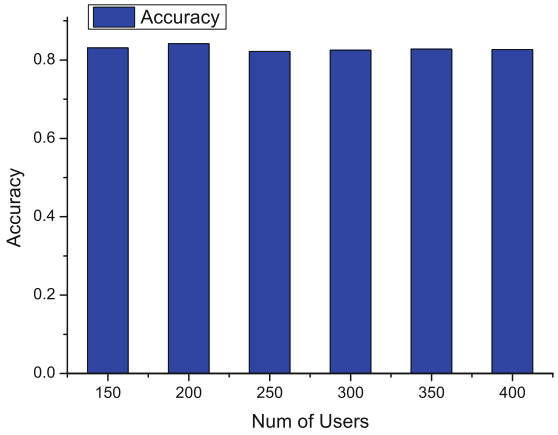

Fig. 4. Accuracy for different number of users.

Conducting Comparison Experiments. We also designed comparative experiments, content-based collaborative filtering, our proposed method

(Adaboost-Bayes) and the Adaboost-Bayes method without considering users' emotions are introduced for the performance comparison on recommendation. For each music categories, five songs are prepared for recommendation, the first two categories are chosen for recommendation, namely 10 songs. The precision, recall and F1-score of three methods are presented in Table 3. Our proposed method, Adaboost-Bayes, outperformed the other two methods, which proved that users' real context have great influence on users' acceptance for the recommended songs.

Table 3. Experimental comparison of different methods

Algorithm contrast	Precision	Recall	F1-score
Collaborate filtering	0.803	0.562	0.645
ABWE	0.742	0.616	0.646
Adaboost-Bayes	0.826	0.578	0.734

6 Conclusions

This study proposes a novel method for music recommendation serving for users' real context, which provides a cooperative mechanism between classification and recommendation. An efficient strategy is designed for guiding classification rules to work for recommendation requirements. Making full use of both users' historical behavior and their current context, the proposed recommendation model uses three functions, namely feature selection, bayes classifying and ensembling strategy for recommendation results, to provide more suitable recommendation. Experiments on real music datasets show the effectiveness of the proposed method. The following work will pay more attention on users' state migration to provide more intelligent recommendation services.

Acknowledgments. This study is funded by the National Natural Science Foundation of China (No. 61462017, 61862013, U1501252, U1711263, 61662015), Guangxi Natural Science Foundation of China (No. 2017GXNSFAA198035), and Guangxi Cooperative Innovation Center of Cloud Computing and Big Data.

References

1. Baccigalupo, C., Plaza, E.: Case-based sequential ordering of songs for playlist recommendation. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS, vol. 4106, pp. 286–300. Springer, Heidelberg (2006). https://doi.org/10.1007/11805816_22
2. Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: using social and content-based information in recommendation. In: Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, pp. 714–720 (1998)

3. Boratto, L., Carta, S., Fenu, G., Mulas, F., Pilloni, P.: Influence of rating prediction on group recommendation's accuracy. IEEE Intell. Syst. **31**(6), 22–27 (2016)
4. Chen, T.-Y., Chen, L.-C., Chen, Y.-M.: Mining location-based service data for feature construction in retail store recommendation. In: Perner, P. (ed.) Advances in Data Mining. Applications and Theoretical Aspects. LNCS, vol. 10357, pp. 68–77. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62701-4_6
5. Dai, J., Yang, B., Guo, C., Ding, Z.: Personalized route recommendation using big trajectory data. In: IEEE International Conference on Data Engineering, pp. 543–554 (2015)
6. Donaldson, J.: A hybrid social-acoustic recommendation system for popular music. In: ACM Conference on Recommender Systems, pp. 187–190 (2007)
7. Gao, L., Wu, J., Qiao, Z., Zhou, C., Yang, H., Hu, Y.: Collaborative social group influence for event recommendation, pp. 1941–1944 (2016)
8. Gu, Y., Song, J., Liu, W., Zou, L., Yao, Y.: Context aware matrix factorization for event recommendation in event-based social networks. In: ACM International Conference on Web Intelligence, pp. 248–255 (2017)
9. Huo, H., Liu, X., Zheng, D., Wu, Z., Yu, S., Liu, L.: Collaborative filtering fusing label features based on SDAE. In: Perner, P. (ed.) Advances in Data Mining. Applications and Theoretical Aspects. LNCS, vol. 10357, pp. 223–236. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62701-4_17
10. Kim, H.H.: A semantically enhanced tag-based music recommendation using emotion ontology. In: Selamat, A., Nguyen, N.T., Haron, H. (eds.) ACIIDS 2013. LNCS, vol. 7803, pp. 119–128. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36543-0_13
11. Kuo, F.F., Shan, M.K.: A personalized music filtering system based on melody style classification. In: IEEE International Conference on Data Mining, p. 649 (2002)
12. Mathew, P., Kuriakose, B., Hegde, V.: Book Recommendation System through content based and collaborative filtering method. In: IEEE International Conference on Data Mining and Advanced Computing, pp. 47–52 (2016)
13. Melville, P., Mooney, R.J., Nagarajan, R.: Content-boosted collaborative filtering for improved recommendations. In: Eighteenth National Conference on Artificial Intelligence, pp. 187–192 (2002)
14. Pampalk, E., Pohle, T., Widmer, G.: Dynamic playlist generation based on skipping behavior. In: Proceedings of the International Conference on Music Information Retrieval, ISMIR 2005, London, UK, 11–15 September 2005, pp. 634–637 (2005)
15. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: International Conference on World Wide Web, pp. 285–295 (2001)
16. Serbos, D., Qi, S., Mamoulis, N., Pitoura, E., Tsaparas, P.: Fairness in package-to-group recommendations. In: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, pp. 371–379. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2017). https://doi.org/10.1145/3038912.3052612
17. Stratigi, M., Kondylakis, H., Stefanidis, K.: Fairness in group recommendations in the health domain. In: IEEE International Conference on Data Engineering (2017)
18. Tiemann, M., Pauws, S.: Towards ensemble learning for hybrid music recommendation. In: ACM Conference on Recommender Systems, pp. 177–178 (2007)
19. Wang, X., Dou, L.: Social recommendation algorithm based on the context of time and tags. In: Third International Conference on Advanced Cloud and Big Data, pp. 15–19 (2016)

20. Wu, W., et al.: Improving performance of tensor-based context-aware recommenders using Bias Tensor Factorization with context feature auto-encoding. Knowl.-Based Syst. **128**(C), 71–77 (2017)
21. Xing, Z., Parandehgheibi, M., Xiao, F., Kulkarni, N., Pouliot, C.: Content-based recommendation for podcast audio-items using natural language processing techniques. In: IEEE International Conference on Big Data, pp. 2378–2383 (2017)
22. Yang, Q., Yao, X., Jingwei, Z., Zhongqin, B.: Exploiting SDAE model for recommendations. In: the 30th International Conference on Software Engineering & Knowledge Engineering, pp. 11–16 (2018)
23. Yin, H., Sun, Y., Cui, B., Hu, Z., Chen, L.: LCARS: a location-content-aware recommender system. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 221–229 (2013)
24. Yoshii, K., Goto, M., Komatani, K., Ogata, T., Okuno, H.G.: Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In: International Conference on Music Information Retrieval, pp. 296–301 (2006)
25. Yu, Z., Tian, M., Wang, Z., Guo, B., Mei, T.: Shop-type recommendation leveraging the data from social media and location-based services. ACM Trans. Knowl. Discov. Data **11**(1), 1 (2016)
26. Zhang, J., Yang, C., Yang, Q., Lin, Y., Zhang, Y.: HGeoHashBase: an optimized storage model of spatial objects for location-based services. Front. Comput. Sci. 1–11 (2018)
27. Zhang, L., Zhou, R., Jiang, H., Wang, H., Zhang, Y.: Item group recommendation: a method based on game theory. In: International Conference on World Wide Web Companion, pp. 1405–1411 (2017)