An Improved Content-Based Music Recommending Method with Weighted Tags

Lu Ding, Ning Zheng, Jiang Xu, and Ming Xu

College of Computer, Hangzhou Dianzi University 310018 Hangzhou, China
ddbb-love@163.com,{nzheng,jian.xu,mxu}@hdu.edu.cn

Abstract. Content-based filtering is widely used in music recommendation field. However, the performance of existing content-based methods is dissatisfactory, because those methods simply divided the listened songs into like or unlike set, and ignored user's preference degree. In this paper, an enhanced content-based music recommending method was proposed by quantifying the user preference degree to songs with weighted tags. Firstly, each listened song was classified into like or unlike set according to user's playing behaviors, such as skipping and repeating. Secondly, the songs' social tags were collected from LastFm website and weighted according to their frequency in the collected tags.Finally, the user's preference degree for each song was quantified with the weighted tags, and the candidate songs with high preference degrees would be recommended to him. On the LastFm dataset, the experimental results demonstrate that the proposed method outperforms those traditional content-based methods in both rating and ranking prediction.

Keywords: Recommend Systems, Social Tagging, Content Filtering.

1 Introduction

Along with the development of World Wide Web, millions of free online music makes it hard for people to find out what they like manually. The recommender systems provide a widely adopted solution to the information overload problem, and can automatically help people to decide what to listen.

The current music recommendation technologies mainly fall into two categories: content-based filtering [1] and collaborative filtering [2]. Content-based filtering technology analyzes the similarity between users or items by the metadata, such as user profiles and music acoustic features. In contrast, collaborative filtering technology analyzes the similarity between users or items by users' past behaviors, no requiring domain knowledge. The Latent factor models like matrix factorization (MF) and the neighborhood model are typical approaches in collaborative filtering. They found the relationships between users and items by analyzing users' listening histories. Besides, there are other new techniques, such as LDA methods [3] and graph-based models [4].

Though the traditional recommenders can effectively predict which song a user likes, they can hardly make an understandable explanation why they do

X. He et al. (Eds.): MMM 2015, Part I, LNCS 8935, pp. 360–371, 2015.

these recommends, and difficultly answer the question of what kind music a user likes or unlike

In this paper, a personalized tag model was proposed for music recommendation. The music's metadata was replaced with the social tags to represent songs. Here, the social tags are keywords generated by internet users on a platform and they are used to describe and categorize an object, concept or idea. As they are originally created by users' own way, they contain meaningful concepts to users. Furthermore, a song's top tags are the most popular ones which are attached by the users on this platform, so that they can stand for the social opinion to songs. Comparing with the audio features, such as pitch and tempo, social tags can better classify and label resource, besides they can make the recommender more understandable. Moreover, the tags were weighted on the perspective of statistic analysis of each user's implicit feedbacks to build each user's personalized tag model. On the LastFm dataset, experiments demonstrate that the proposed method can outperform the traditional content-based method in both rating and ranking prediction.

The remainder of this paper is organized as follows. Section 2 introduces prior work related to this paper. Then the details of the proposed method are presented in Section 3. In Section 4, experiments are discussed, followed by conclusion and future work in Section 5.

2 Related Work

In this section, some of research works related to tag-based music recommender systems are presented.

The methods which recent research papers apply in tag-based music recommendation can be classified into two categories. One way is to use tag data as content information to compute users or items similarity. For examples, Bosteels et al. [5] used social top tags to calculate similarity values between listened songs and candidate songs, and then used those values as the fuzzy relationship degrees to compare performance of some different heuristics. It can reduce the predicting failure rate comparing with the method in [6], which used the audio-based similarity. The main difference between the proposed method and [5] is that the tags' weights were equal in [5] but in the proposed method tags were assigned with different weights according to their frequency in the collected tags. Kim [7] assigned weight to tags according to the intensity of tag's emotion which was judged by the SentiWordNet [8], after then, user profiles using the weighted tags were generated and a user-based collaborative filtering algorithm was executed. The main barrier in [7] is the sparsity problem as the tags each user assigned to a song are very poor.

The other way is to make use of the tag data to build recommend models. For example, Zhang et al. [9] proposed a random walk model, which was based on Page Rank liked random walks among the user-item, user-tag and item-tag bipartite graphs. Besides, it also made use of the tag information to build item graphs and user graphs based on a probabilistic method. As it applied random walks on ternary interaction graphs to capture transitive associations between users and items, the sparsity problem had been alleviated but it brought the huge time consumption on building graphs and searching. Hariri et al. [10] proposed a LDA model to predict what topics next song contains. In that paper, the tag data was used to establish the topic modeling module. Each song was represented as a set of topics. By matching the current topic sequence with history frequent topic sequential pattern, the next topics were found. Then songs which contain these topics were recommended. The main problem is that the recommender's performance relies on the number of LDA topics, so that different settings result in different frequent sequential patterns and affect the recommendation performance. Taramigkou et al. [3] also used the LDA model, the difference is the tags are artists' tags which are assigned by users. Then a users graph was build and the weights of edges were assigned using the cosine similarity between topic vectors generated for each user by LDA model. Finally, the Dijkstra algorithm [11] was used as a graph search approach to recommend a list of artists. But all these mentioned models didn't do a good job of explaining why the song is recommended, and answering the question what users like or unlike. In this paper, those weighted tags are straightly used to form users' profiles. The greater weight the tag has, the more a user like the tag, vice versa.

3 The Proposed Method

Give the history logs which contain songs that have been listened previously by users, each user's music taste can be established by analyzing these logs. This paper uses social tagging web sites to retrieve each song's top tags and use these tags as features to build up users' personalized tag model. Then for a given user, a list of songs which are close match with his/her tag model will be recommended.

This section first presents the format of users' listening logs from LastFm website, and the method of capturing extra information for each entry. Then these complementary logs are used to analysis users' behaviors. At last, the personalized tag model for music recommendation is presented.

3.1 Acquiring the Extra Information

The logs is formatted one entry per line as follows: userid \ timestamp \ musicbrainz-artist-id\ artist-name\ musicbrainz-track-id \ track-name. The timestamp was the moment when a user started this track. The musicbrainz-artist-id and musicbrainz-track-id are MusciBrainz Identifiers (MBIDs),[1] an MBID is a 36 character Universally Unique Identifier that is permanently assigned to each entity in the MusicBrainz database. For example, the artist Adele has an artist MBID of cc2c9c3c-b7bc-4b8b-84d8-4fbd8779e493, and her song Best for Last has a recording MBID of 84c00aff-b2cf-4bf7-a2e3-9460820efb03. The aim of MusicBrainz is to be the universal lingua franca for music by providing a reliable

[1] http://musicbrainz.org/doc/MusicBrainz_Identifier

and unambiguous form of music identification. So if the logs'multi-entries have the same musicbrainz-track-id, then it means they are the same track objects, and different musicbrainz-track-ids mean different track objects.

Having these musicbrainz-track-ids, the duration and top5 tags of each track in the users' history logs can be got by using the method named track.getinfo, which is provided by the LastFm API. This extra information is very important, and it will be combined with the original logs to analysis users' behaviors and establish each user's personalized tag model in the following sections.

3.2 Analysis of Users' Behaviors

To capture users' music bias, it is necessary to find the implicit feedbacks through looking inside the logs. The proposed method is based on a simplistic assumption that a user like a song if he continuously repeated it, and a user dislike a song if he skipped it several times. Despite once skip didn't infer dislike, you skipped a special song many times may infer you dislike it. The main goal of this subsection is to present the key method of finding out such behaviors.

Supposing you listen to the songs like this: there are some buttons which can be used to control the music player to "play next", "play back", and when you started a song the timestamp and its' mbid was recorded in the log. At the beginning, you listened to two songs and during this time you didn't press any button. Then, you listened to the third one but in the half way you pressed the "play next" button because you felt it's not good. After that, you completely finish listening to the fourth one and found this song is very fit for you, so you pressed the "play back" button in order to re-listen it, after listening it twice you closed the player. During this procedure five entries were wrote into the log, which can be simply formatted as follows:

Session 1=$\{\langle$ t1,m1$\rangle$,$\langle$t2,m2$\rangle$,$\langle$t3,m3$\rangle$,$\langle$t4,m4$\rangle$,$\langle$t5,m4$\rangle\}$. One tuple stands for one entry, and the notations are defined in Table 1. All tuples are in chronological order. What we need to do is to find out the "play next", "play back" actions in this session. An algorithm was proposed:

1. Step 1, calculate the length of each song played. As the timestamp is the start moment, so p1=t2-t1;
2. Step 2, compare the length with song's duration. If you pressed "play next", then length<duration, in this session p3<d3;
3. Step 3, compare adjacent songs' mbids, if you re-listened a song, then adjacent songs' mbids were the same. Algorithm 1 shows the details for analyzing the whole logs.

More formalized description of users' behaviors is displayed below:

1. Shuffling, it means in one session the user u just played the track i once. It corresponds to "nothing", and is marked as "Normal"
2. Skipping, it means in one session the user u skip the track i. It corresponds to "play next", and is marked as "Skip"

3. Repeating, it means in one session the user u continuously plays the track i. It corresponds to "play back", and is marked as "Repeat"

The commonly used notations for this work are displayed in Table 1.Two main functions are the length's calculation of each entry in the logs heard, and the method to find repeating behavior. They are defined as follows:

Table 1. Commonly Used Notations

Representations	Descriptions
L	the set of entries in the completed users' logs
U	user set
l,u	entry l,user u
p_l	the length of track in the l be heard
S_l,R_l,N_l	the Skip/Repeat/Normal Flag of l
$L(u)$	the sequence of entries produced by user u
d_l	the duration of track in l
t_l	the timestamp of l
m_l	the mbid of track in l

$$\Delta(i,j) = t_i - t_j(i,j \in L)\ . \tag{1}$$

$$com(i,j) = \begin{cases} 1 & \text{if } m_i \text{ equals to } m_j \\ 0 & \text{otherwise} \end{cases} (i,j \in L)\ . \tag{2}$$

In Algorithm.1 the purpose of line 6-9 is to calculate the length of each song had been listened. The purpose of line 13-14 is to find the Repeating behavior, and mark it "Repeat". The purpose of line 15-16 is to find the Skipping behavior, and mark it "Skip". The purpose of line 17-18 is to find the Shuffling behavior, and mark it "Normal". The purpose of line 23 is to add these marks into user's original datasets.

3.3 The Proposed Model

As it was mentioned in section 3.1, the extra information of top tags will be used in the proposed model. The reason for choosing these tags is that they not only describe various features of the songs including genre, artist name and era, but also describe users' attitudes toward the songs, such as sadness, mellowness, and so on. Although people may have different opinions about songs, top tags with frequency above a minimum threshold capture the social opinion about each song. These features can often be very helpful in explaining the commonalities in a set of songs selected by a user. Table 2 shows a set of songs' top tags as example.

Algorithm 1. Analyzing users' behaviors

```
Input: $L, U$, Session Threshold value $\lambda$, Skip Threshold value $\theta$
Output: $L^+$
Initialize the Skip, Repeat, Normal Flags with 0 for each $l$ in $L$;
for each user $u$ in $U$ do
    for $i = 0 \to length.L(u)$ do
        calculate $\Delta(l_i, l_{i+1})$ use function(1);
        // $l_{i+1}$ denotes the $(i+1)$th entry in the $L(u)$
        if $\Delta(l_i, l_{i+1}) < \lambda$ then
            $p_{l_{i+1}} \leftarrow \Delta(l_i, l_{i+1})$
        else
            $p_{l_{i+1}} \leftarrow -1$
        end if
        if $p_{l_{i+1}} \neq -1$ then
            calculate $com(l_i, l_{i+1})$ use function(2);
            if $com(l_i, l_{i+1})$ then
                $R_{l_i} \leftarrow 1$ and $R_{l_{i+1}} \leftarrow 1$
            else if $p_{l_{i+1}} < \theta$ then
                $S_{l_{i+1}} \leftarrow 1$
            else
                $N_{l_{i+1}} \leftarrow 1$
            end if
        else
            $R_{l_{i+1}} \leftarrow 0, S_{l_{i+1}} \leftarrow 0, N_{l_{i+1}} \leftarrow 0$
        end if
        update the Skip, Repeat, Normal Flags with $S_{l_i}, R_{l_i}, N_{l_i}$ for each $l$ in $L(u)$;
    end for
    $L^+ \leftarrow L^+ \cup updated\_L(u)$
end for
return $L^+$
```

Table 2. Top tags of songs

Artist name	Track name	Top tags
Valerie Etienne	The Birds Sing	jazz, chillout, lounge, groovy, female vocalist
U2	If God Will Send His Angels	rock, u2, soundtrack,90s, irish
Beck	Lost Cause	mellow, alternative, indie, sad, singer-songwriter
Bjork	Show Me Forgiveness	alternative, icelandic, bjork, female vocalists, electronic

Genres like “jazz”, eras like “90s” ,artists likes “u2”, mood likes “mellow” can be found in the Table 2. Using these top tags, more diverse and deeper factors can be found, and the recommender can be more precise and understandable.

To build the proposed model, all the songs’ top tags were collected in skip set, repeat set, and normal set respectively. After getting rid of the same tags, weights were assigned to them according to frequency.

Marking a user u’s skip tags set, repeat tags set and normal tags set as ULT, LT and NT respectively, the function to compute weight is:

$$score(t_i|u, ULT) = fr(t_i|ULT) . \tag{3}$$

$$score(t_i|u, LT) = fr(t_i|LT) . \tag{4}$$

$$score(t_i|u, NT) = fr(t_i|NT) . \tag{5}$$

Where $fr(t_i|ULT)$ is the overall frequency of tag t_i in user u’s skip tags set. The personalized tag model for music recommendation is formalized as follows:

$$TModel(u) = \alpha \cdot score(u, ULT) + \beta \cdot score(u, LT) + \gamma \cdot score(u, NT) . \tag{6}$$

Where α, β, γ are independent params to control the power of different tags in ULT, LT and NT , $-1 \leq \alpha < 0$; $0 < \gamma \leq \beta \leq 1$; denoting a candidate song i’s top tag set is iT , then its predict score is :

$$\begin{aligned} \widehat{r}_{ui} = \alpha \cdot & \sum_{t_i \in \{iT \cap ULT\}} score(t_i|u, ULT) \\ + \beta \cdot & \sum_{t_i \in \{iT \cap LT\}} score(t_i|u, LT) \\ + \gamma \cdot & \sum_{t_i \in \{iT \cap NT\}} score(t_i|u, NT) \end{aligned} \tag{7}$$

Where $\sum_{t_i \in \{iT \cap ULT\}} score(t_i|u, ULT)$ is the accumulated weight of tags in both of iT and ULT , we treat this value as the possible degree of user u hate this candidate song i. Where $\sum_{t_i \in \{iT \cap LT\}} score(t_i|u, LT)$ is the possible degree of user u like this song i, and $\sum_{t_i \in \{iT \cap NT\}} score(t_i|u, NT)$ is the possible degree of user u neither hate nor like this song i.

In Algorithm.2 the purpose of line 3-5 is to remove the same tags in intersections. The purpose of line 7 is to assign weight to tags. The purpose of line 8 is to establish user’s personalized tag model. Based on the user’s personalized tag model, the predicted score of a candidate song can be calculated using the function (7). The higher the song’s score is, the more the user like it, vice versa.

Algorithm 2. Building the proposed model

```
Input: $L^+$,$U$ ,Params $\alpha, \beta, \gamma$
Output: A list of personalized tag models $TModel$
for each user $u$ in $U$ do
    split $L^+(u)$ into $LT^+(u), ULT^+(u), NT^+(u)$
    $\overline{LT}(u) \leftarrow LT^+(u) - (LT^+(u) \cap ULT^+(u)) - (LT^+(u) \cap NT^+(u))$
    $\overline{ULT}(u) \leftarrow ULT^+(u) - (LT^+(u) \cap ULT^+(u)) - (ULT^+(u) \cap NT^+(u))$
    $\overline{NT}(u) \leftarrow NT^+(u) - (LT^+(u) \cap NT^+(u)) - (ULT^+(u) \cap NT^+(u))$
    for each tag $t$ in $\overline{LT}(u)/\overline{ULT}(u)/\overline{NT}(u)$ do
        calculate $f_{\overline{LT(u)}}(t)$, $f_{\overline{NLT(u)}}(t)$, $f_{\overline{NT(u)}}(t)$ use function (3),(4),(5);
        $TModel(u) \leftarrow \alpha \cdot f_{\overline{LT}} + \beta \cdot f_{\overline{NLT}} + \gamma \cdot f_{\overline{NT}}$
    end for
    append $TModel(u)$ to $TModel$;
end for
return $TModel$
```

4 Experimentations

4.1 Dataset

The public dataset (Last.fm Dataset-1K users) [2] from the website of Music Recommendation Datasets for Research was chosen to evaluate the proposed method in this paper. This dataset contains the whole listening habits during May, 5th 2009 to May 2010 for 992 users. The total number of records is 19,150,868. Each entry includes "userid", "timestamp", "artid", "artname", "trackid" and "trackname", and there have 961,416 unique trackids.

4.2 Evaluation Measures

To evaluate the proposed model, two different kinds of evaluation metric are chosen. One is the rating prediction, and the other is the traditional top-K recommendation evaluation metric. The recent 1/11 entries of each user's records are chosen as test set, and the remaining as training set.

The first task of this recommender is to predict whether the user u will repeat/skip/normal play the track i in test set according to its' predict score which is produced by u's personalized tag model. Then counted how many times it does the right decision. More formally, each user corresponds to a (R, S, N, rr, ss, nn) tuple, where R, S and N are the sets of repeated, skipped and normal played songs by the user, respectively. And "rr" means this repeated song is judged to be repeated, and it's a correct judgment, and the same to"ss"and "nn". Then the hit rate of repeated songs is $Hit(R) = \frac{\sum rr}{\sum |R|}$, the hit rate of skipped songs is $Hit(S) = \frac{\sum ss}{\sum |S|}$, and the hit rate of normal played songs is $Hit(N) = \frac{\sum nn}{\sum |N|}$. The overall hit rate is $Hit = \frac{\sum (rr+ss+nn)}{\sum (|R|+|S|+|N|)}$. The second task of our recommender

[2] http://www.dtic.upf.edu/ ocelma/MusicRecommendationDataset/lastfm-1K.html

is to rank the songs in test set according to the predicted scores, then recommend top-K songs to the user. We compute the precision and recall of top-K recommendations as follows:

$$Precision@K = \frac{\sum\limits_{u \in U} |R_k(u) \cap T(u)|}{\sum\limits_{u \in U} |R_k(u)|} . \tag{8}$$

$$Recall@K = \frac{\sum\limits_{u \in U} |R_k(u) \cap T(u)|}{\sum\limits_{u \in U} |T(u)|} . \tag{9}$$

Where $R_k(u)$ is the recommended songs set for user u, $T(u)$ is the accepted songs set of user u.

And compute the F1-meature value as follows:

$$F1@K = \frac{2 \times Precision@K \times Recall@K}{Precision@K + Recall@K} . \tag{10}$$

4.3 Performance Comparison

In this section, the proposed model and the Fuzzy theory method [5] are compared to evaluate whether the weighted tags can improve the recommender's performance when compared with the unweighted tags. Tracks were considered skipped when the user listened to less than 50% of them.For the listening session segmentation, we let the songs to be in the same session, if the user's inactive intervals between adjacent songs are less than 1 hour, which refer to Xiang et al. [12]. And the parameters such as α, β, γ are determined experimentally. In this experiment we set $\alpha = 1, \beta = -1, \gamma = 0.5$.

First Task Result. After getting song i's predicted score , do the judgments as follow: 1) song i will be skipped, if $\widehat{r}_{ui} < -|threshold|$. 2) song i will be repeated, if $\widehat{r}_{ui} > |threshold|$. 3) song i will be normal played, if $-|threshold| \leq \widehat{r}_{ui} \leq |threshold|$. The threshold value is changed from 0 to 1 by step 0.05. In fuzzy theory method the skip hit rate is 8.5% and the overall hit rate is 94.78%.

Fig. 1 shows the hit ratios of the proposed model for different threshold values. It can be found that along with the bigger value of threshold, both of the $Hit(R)$ and $Hit(S)$ decrease, and both of the $Hit(N)$ and overall Hit increase. The reason of these phenomena is that the threshold divides user bias space into three areas. The bigger of threshold the larger of normal area, the smaller of repeat area and skip area. The best performance of overall hit rate is 91.73% when the threshold is 0.25, meanwhile the normal hit rate is 99.9%, the skipped hit rate is 9.02%, and the repeated hit rate is 1.91%.

Comparing to the fuzzy theory method, the proposed approach do have significant improvement in the skip hit rate. It has a gain of 6.12% over the fuzzy model (8.5%), although the overall hit rate slightly reduces 3.22%. It indicates the proposed model can better identify what users unlike.

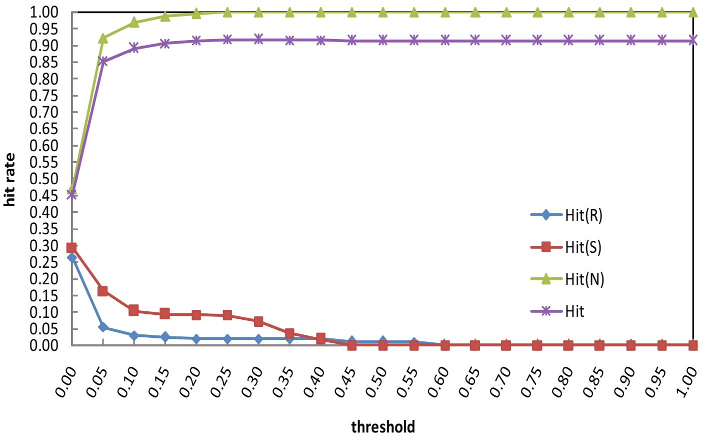
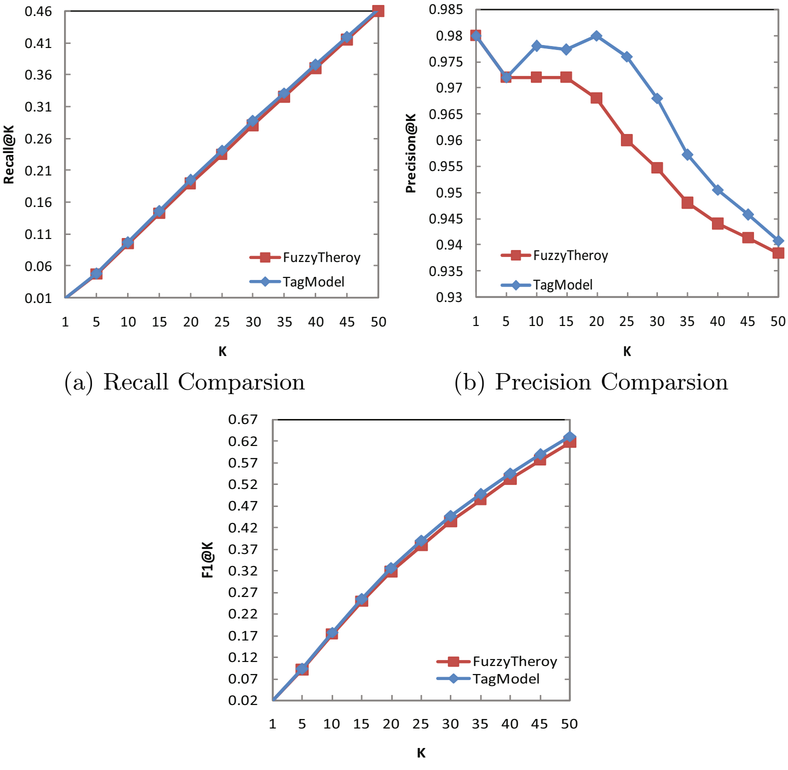


Fig. 1. Hit ratio of personalized Tag Model for different threshold



(a) Recall Comparsion

(b) Precision Comparsion

(c) F1-measure Comparison

Fig. 2. Comparing Fuzzy theory method and personalized Tag Model with $k \in [1, 50]$

Second Task Result. The proposed model and the fuzzy theory method are compared in this section. The results are showed in Fig. 2. Here, the horizontal axis stands for different number of recommendations, and the three vertical axes stand for the value of recall, precision and F1-measure respectively. From Fig. 2, it could be observed that the proposed model is better than the fuzzy theory method in precision, recall and F1-measure. Specifically, the proposed Model achieves an average improvement in precision of 1.68% over the fuzzy method. And it achieves an average improvement in recall of 0.748% and an average improvement in F1-measure of 0.751% over the fuzzy method.

5 Conclusion

This paper proposes a simple and effective personalized tag model to mine users' music bias. Due to the weighted tags, it can explain what user likes or unlike understandably as the greater weight the tag has, the more the user like it, vice versa. Experiments conducted on the LastFm dataset show that the personalized tag model outperforms the traditional content-based model in recall, precision and F1-measure, and also has a big improvement in detecting skipping behavior. In the future, we plan to take the time into account to build a dynamic personalized tag model.

Acknowledgments. This work is supported by the Natural Science Foundation Natural Science Foundation of China under Grant No.61070212 and 61003195, the Zhejiang Province Natural Science Foundation Natural Science Foundation of China under Grant No.Y1090114 and LY12F02006, the Zhejiang Province key industrial projects in the priority themes of China under Grant No 2010C11050, the soft science research project of Hangzhou (No. 20130834M15).

References

1. Ferman, A., Errico, J., Beek, P., Sezan, M.: Content-based filtering and personalization using structured metadata. In: 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 393–393. ACM, Portland (2002)
2. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative Filtering model. In: 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 426–434. ACM, Las Vegas (2008)
3. Taramigkou, M., Bothos, E., Christidis, K., Apostolou, D., Mentzas, G.: Escape the Bubble: Guided Exploration of Music Preferences for Serendipity and Novelty. In: 7th ACM Conference on Recommender Systems, RecSys 2013, pp. 335–338. ACM, Hong Kong (2013)
4. Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., Sun, J.: Temporal recommendation on graphs via long-and short-term preference fusion. In: 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010, pp. 723–732. ACM, Washington, DC (2010)

5. Bosteels, K., Pampalk, E., Kerre, E.E.: Evaluating and Analysing Dynamic Playlist Generation Heuristics Using Radio Logs and Fuzzy Set Theory. In: Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe, pp. 351–356 (2009)
6. Pampalk, E., Pohle, T., Widmer, G.: Dynamic Playlist Generation based on Skipping Behavior. In: Proceedings of the 6th International Society for Music Information Retrieval Conference, ISMIR 2005, London, pp. 634–637 (2005)
7. Kim, H.H.: A Semantically Enhanced Tag-Based Music Recommendation Using Emotion Ontology. In: Selamat, A., Nguyen, N.T., Haron, H. (eds.) ACIIDS 2013, Part II. LNCS, vol. 7803, pp. 119–128. Springer, Heidelberg (2013)
8. Esuli, A., Sebastiani, F.: SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In: Proceedings of the 5th Conference on Language Resources and Evaluation, LREC 2006, Genova, Italy, pp. 417–422 (2006)
9. Zhang, Z., Daniel, D.Z., Ahmed, A., Jing, P., Zheng, X.L.: A Random Walk Model for Item Recommendation in Social Tagging Systems. ACM Transactions on Management Information Systems 4(2) 8, 1–24 (2013)
10. Hariri, N., Mobasher, B., Burke, R.: Context-aware music recommendation based on latent topic sequential patterns. In: 6th ACM Conference on Recommender Systems. RecSys 2012, pp. 131–138. ACM, Dublin (2012)
11. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1(1), 269–271 (1959)
12. Xiang, W., Qi, L., Enhong, C., Liang, H., Jingsong, L., Can, C., Guoping, H.: Personalized Next-song Recommendation in Online Karaokes. In: 7th ACM Conference on Recommender Systems, RecSys 2013, pp. 137–140. ACM, Hong Kong (2013)