EVALUATION ON FEATURE IMPORTANCE FOR FAVORITE SONG DETECTION

Yajie Hu, Dingding Li and Ogihara Mitsunori
Department of Computer Science
University of Miami
yajie.hu@umail.miami.edu, d.wang1@miami.edu
ogihara@cs.miami.edu

ABSTRACT

Detecting whether a song is favorite for a user is an important but also challenging task in music recommendation. One of critical steps to do this task is to select important features for the detection. This paper presents two methods to evaluate feature importance, in which we compared nine available features based on a large user log in the real world. The set of features includes song metadata, acoustic feature, and user preference used by Collaborative Filtering techniques. The evaluation methods are designed from two views: i) the correlation between the estimated scores by song similarity in respect of a feature and the scores estimated by real play count, ii) feature selection methods over a binary classification problem, i.e., "like" or "dislike". The experimental results show the user preference is the most important feature and artist similarity is of the second importance among these nine features.

1. INTRODUCTION

In the recent digital world, millions of digital songs are available online and it is difficult for users to manually search favorite songs. A music recommender system is the solution that helps users find their possible favorite songs in the song ocean, and makes a personalized journey for the user to listen these songs one by one. Essentially, the expected recommender system must be able to answer the very ask, i.e., give me my favorite songs. Automatically detecting favorite songs is therefore an important part in a music recommender system. Basically, recommender systems apply for content-based methods, Collaborative Filtering (CF) techniques, or both to detect favorite songs.

Audio content-based methods use favorite songs to predict other songs users may like [6], based on their similarity to the favorite songs. In order to recommend favorite songs to users, a music recommender system using content-based analysis depends on manual or automatic audio content description to compute the similarity among songs.

For instance, Pandora [1] employs a team of musician analysts to listen to music and give each recording its descriptions, which includes melody, harmony, instrumentation, rhythm, vocals, lyrics and etc. A weighted Euclidean distance is used to find similar songs [3].

Z. Cataltepe et al. music recommender system is based on audio similarity and users' listening history [2]. An industrial-strength music recommender system introduces the Euclidean distance of two tracks over a reduced space using Principal Component Analysis [1]. The system uses several audio features, including Mel-Frequency Cepstral Coefficients (MFCC), tempo, key mode and others.

Music was one of the first forms of content to be addressed by collaborative filtering recommender systems such as Ringo, Firey and HOMR [9, 10]. This technique finds users with similar music preferences and recommends items liked by these similar users to the target user [7]. It could be seen as a method to measure the similarity based on the user preference. Hence, the user preference is considered as one type of feature in our work. Some hybrid methods combined with collaborative filtering techniques and content-base methods are proposed to solve the cold-start problem [4, 11]. B. McFee et al. presented a method for deriving item similarity from a sample of collaborative filter data, and use the sample similarity to train a distance metric over acoustic features. The trained distance metric is used to improve the shortcoming of CF techniques, i.e., incomplete data hamper.

This paper focuses on systemic analysis regarding how important both audio content features and collaborating filtering techniques are for favorite songs detection in the real world. The two evaluation methods are described in Section 2. In Section 3, we introduce the data and present the evaluation results. We make the conclusions and discuss the future work in Section 4.

2. METHODS

We evaluate the importance of a type of feature (In the following paragraphs, "a feature" means "a type of feature" instead of a value in a feature.) by the correlation between

[1] http://www.pandora.com/about/mgp

the predicted scores using this feature and the scores by real play count. Moreover, the recommendation problem is converted to a binary classification problem. As a result, the feature selection results are used to evaluate the feature importance.

2.1 Correlation Evaluation

Some content-based recommender systems assume that a song with high similarity to the favorite songs would be liked by the user [1, 2]. Thus, the systems measure the similarity among songs and recommend the song with the highest similarity to the favorite songs. Euclidean distance and Dynamic Time Warping (DTW) algorithm are used to measure the similarity between two songs.

Euclidean distance sees a vector as a point in Euclidean space, and is given by the Pythagorean formula. This metric works for non-sequential features, like tempo.

For sequential feature, like pitch, DTW algorithm is good metric to measure the distance between two sequences, which may vary in time, since DTW algorithm is able to find the optimal alignment between two time series [8]. One time series may be non-linearly "warped" to the other one by stretching or shrinking it along its time series.

If a certain feature is important, the song similarity in terms of the feature should be highly effective to detect whether a song is favorite for a user. The highly effective detection is expected to give high relevant rating score, and the predicted rating score should correlate or anti-correlate the true one. Thus, correlation between the predicted scores and the true scores, which is estimated by play count, is therefore used to evaluate the importance of the feature. The correlation result is normalized to $[-1.0, 1.0]$, which -1.0 is in the case of a perfect negative (decreasing) linear relationship, and +1.0 means a perfect positive (increasing) linear relationship. The greater absolute value is, more important the feature is.

In a user's log, the predicted rating score is given by the following equation.

$$\hat{r}_i^u = \frac{\sum_{s_j \in \mathbf{R}^u, j \neq i} r_j^u \cdot sim_f(s_i, s_j)}{\sum_{s_j \in \mathbf{R}^u, j \neq i} sim_f(s_i, s_j)}, \tag{1}$$

where $\hat{r}_i^u$ denotes the predicted score for song s_i, and $\mathbf{R}^u$ is the set of rated songs. Rating score r_j^u is estimated by the play count of song s_j. $sim_f(s_i, s_j)$ is the similarity between song s_i and song s_j in terms of feature f.

We apply this approach to predict the rating score for each song listened by user u, and get the predicted scores $\{\hat{r}_1^u, \hat{r}_2^u, ..., \hat{r}_n^u\}$. The correlation between $\{r_1^u, r_2^u, ..., r_n^u\}$ and $\{\hat{r}_1^u, \hat{r}_2^u, ..., \hat{r}_n^u\}$ are used to evaluate the importance of feature f.

Furthermore, CF technique predicts the interest of a user in a song by collecting preferences or taste information from other users. This technique is based on the assumption that a user A is more likely to have a user B's opinion on a song x than to have the opinion on x of a user chosen randomly, if A has the same opinion as B on many songs. A user's preference or taste is represented by the opinion on the songs the user listened.

We consider a user as a document and treat the songs which the user listened as terms in the document. Then, the user could be represented by a vector of songs using TF-IDF values. The cosine distance between two vectors are used to measure the similarity between the two corresponding users.

Then, we measure the similarity between two songs considering user similarity and rating scores by the following equation.

$$\begin{aligned}
&\text{if } |\mathbf{U}_i| \leq |\mathbf{U}_j| \\
&\quad sim(s_i, s_j) = \\
&\frac{\sum_{u_m \in \mathbf{U}_i} \left| \lambda \max_{u_n \in \mathbf{U}_j} sim(u_m, u_n)^2 - \eta \left[\left(r_i^m - r_j^n \right) / V \right]^2 \right|}{|\mathbf{U}_i|}, \\
&\text{otherwise} \\
&\quad sim(s_i, s_j) = sim(s_j, s_i),
\end{aligned} \tag{2}$$

where $\mathbf{U}_i$ is the set of users who played song s_i, and u_m denotes the user m. V is the rating scale. λ and η are two regulatory factors.

This measurement allows the similarity of two songs to be high, if two users have high similar tastes, and rate the two songs at the same score. On the contrary, if two users have high similar tastes, and rate two songs at totally different scores, or the two users' tastes are different but the ratings are similar, the two songs must be different in terms of user preference.

This type of similarity is also used to predict song ratings by Equation 1, in order to evaluate the importance of user preference.

2.2 Feature Selection Methods Evaluation

Furthermore, this problem is about classifying a song to a label, i.e., "like" or "dislike", so it is a binary classification problem. We compare several feature selection methods, and select χ^2 Statistic (Chi), Information Gain (IG), Information Gain Ratio (IG Ratio) and Uncertainty to evaluate the importance of features, respectively.

χ^2 Statistic is used to investigate how much a feature depends on the category by the following equation.

$$\chi^2 = \frac{(ad - bc)^2 (a + b + c + d)}{(a + b)(c + d)(b + d)(a + c)}, \tag{3}$$

where a, b, c and d are the number of instances in different cases as shown in Table 1. If the feature is categorical, "Data Type" is the category. If the feature is numerical, "Data Type" is the data range.

IG evaluates the importance of a feature from the view of information theory. Generally speaking, the expected IG is the change in information entropy from a prior state to a posterior state that takes some information as given:

$$\mathrm{IG}(T, f) = \mathrm{H}(T) - \mathrm{H}(T|f), \tag{4}$$

Feature	Data Type 1	Data Type 2	Total
Category 1	a	b	a+b
Category 2	c	d	c+d
Total	a+c	b+d	a+b+c+d=N

Table 1. General notation for a 2 x 2 contingency table

where $\mathrm{IG}(T, f)$ denotes the IG of the feature f on the training data T. $\mathrm{H}(T)$ is the entropy of the training data, and $\mathrm{H}(T|f)$ is the average conditional entropy of T.

IG Ratio introduces Intrinsic Value (IV) to evaluate the feature importance, and it is the ratio between the information gain and the intrinsic value.

$$\mathrm{IGR}(T, f) = \mathrm{IG}(T, f)/\mathrm{IV}(T, f), \tag{5}$$

where $\mathrm{IGR}(T, f)$ is the IG Ratio of feature f and $\mathrm{IV}(T, f)$ is the intrinsic value given by:

$$\mathrm{IV}(T, f) = -\sum_{v} \frac{|\{t \in T|t_f = v\}|}{|T|} \log\left(\frac{|\{t \in T|t_f = v\}|}{|T|}\right) \tag{6}$$

Uncertainty measures the relevance of a feature by calculating the symmetrical uncertainty with respect to the class.

$$\mathrm{Uncentainty}(f) = 2 \cdot \frac{(p(c) - p(c|f))}{p(c) + p(f)} \tag{7}$$

These four methods evaluate the feature importance from different views, and Chi-square statistic and Information Gain have been proved that they surpass among feature selection methods in a text classification task [5]. We therefore apply each of them to evaluate the feature importance.

3. ANALYSIS

An ideal analysis is expected to run on a large song dataset and a big user set. Furthermore, the users should explicitly rank every song in the dataset. However, regrading to the private information protection, the exposed information is limited. This section will compare several dataset, select one dataset and evaluate several types of features.

3.1 Dataset

There are three possible and available dataset online, i.e., Taste Profile Dataset by the Echo Nest, Last.fm dataset 360K and Yahoo! Music User Ratings of Songs by Yahoo! Research.

The Taste Profile Dataset [2] is a huge collection of real world anonymous listener data in the form of Echo Nest Taste Profiles. Considering the protection of individuals private information, the data includes a shuffled hash of persistent session identifiers from a very small random selection of the musical universe and only play counts associated with Echo Nest song IDs that overlap with the Million

[2] http://labrosa.ee.columbia.edu/millionsong/tasteprofile

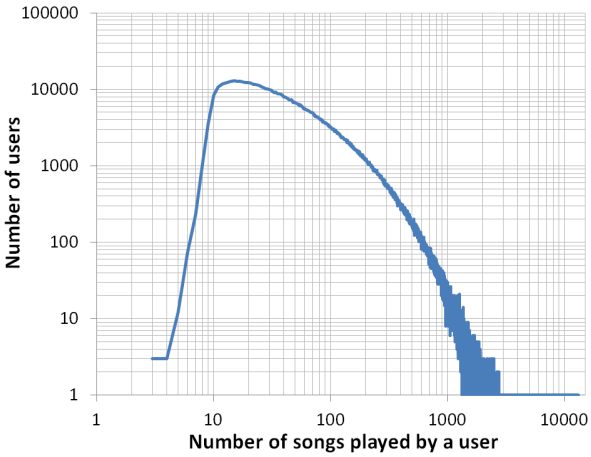


Figure 1. Distribution of users

Song Dataset (MSD) [3]. No usernames, listener details, original IDs, dates, IPs, locations or anything but random user string, Echo Nest song ID, and play count are being released [4]. The Taste Profile dataset has 1,019,318 unique users, 384,546 unique MSD songs and 48,373,586 <*user ID*, *song ID*, *play count*> triplets. The triplets don't have any time stamps and they are not in chronological order. However, the Echo Nest song ID overlaps with the MSD, which is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. The MSD therefore enables researchers look into the songs by many acoustic features and metadata, it doesn't provide the audio file though.

Last.fm dataset 360K [5] collects <*user*, *artist ID*, *plays*> triplets from Last.fm API. *Plays* has two parts: *song title* and *play count*. The triplets don't have any time stamps and not in chronological order. The dataset contains 359,347 unique users and 294,015 artists and 17,559,530 <*user*, *artist ID*, *plays*> triplets. Moreover, more information about users is available, including *gender*, *age*, *country* and *date* (The "date" description is not found). The song metadata and social tags are searchable on Last.fm by *song title* and *artist ID*, but the acoustic features are not available on Last.fm.

Yahoo! Music User Ratings of Songs [6] represents a snapshot of the Yahoo! Music community's preferences for various songs. It contains over 717 million ratings of 136 thousand songs given by 1.8 million users of Yahoo! Music services. The rating triplet is <*user ID*, *song ID*, *rate*>, which *rate* is an integer from 1 to 5. All information of each song in the dataset is artist, album and genre. The users, songs, artists and albums are represented by randomly assigned numeric id's so there is no identifying information revealed. Consequently, it is impossible to accompany more information to the song.

The dataset comparison is summarized in Table 2. Al-

[3] http://labrosa.ee.columbia.edu/millionsong/

[4] http://blog.echonest.com/post/11992136676/taste-profiles-get-added-to-the-million-song-dataset

[5] http://mtg.upf.edu/node/1671

[6] http://webscope.sandbox.yahoo.com/catalog.php?datatype=r

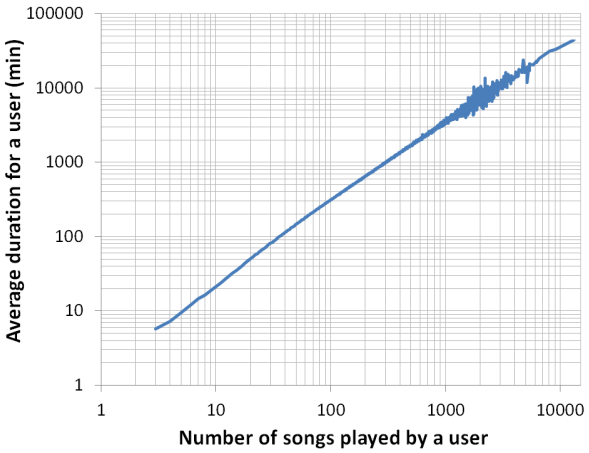


Figure 2. Distribution of estimated duration

though the Taste Profile Dataset doesn't provide explicit rating values, features and metadata provided by MSD supplies us for the probability of evaluating feature importance, and play counts could implicitly represent rating scores. Hence, we select the Taste Profile Dataset to do the experiment.

The distribution of users by the number of songs is shown in Figure 1. The user logs, of which the number of played songs are less than three, are removed by the Taste Profile dataset. The distribution mainly locates in [10, 1,000], and it has a long tail.

Users' preferences would be changed by different contexts in a long period so the overall listening duration of a user should be taken into account. The Taste Profile dataset doesn't show the actual listening duration of a song. We roughly estimate how long a user listens to the songs in the user log by Equation 8.

$$T = \sum_{i \in S}^{i} n_i \cdot t_i \cdot \left(1 - \frac{1}{n_i + 1}\right), \tag{8}$$

where n_i is the play count of song i by the user. t_i is the duration of song i, which can be obtained from MSD. The estimated duration of playing songs by users is shown in Figure 2.

3.2 Evaluation

MSD provides many types of features for a song as shown in Table 3. We select eight features from the MSD, i.e., *loudness*, *pitches*, *tempo*, *duration*, *song hotness*, *artist similarity*, *artist hotness* and *artist familiarity*, and these features cover the three categories in Table 3. Furthermore, user preference is counted in. The similarity is able to indirectly represent the performance of CF techniques in favorite song detection.

Euclidean distance is applied to measure the distance of songs for non-sequential features, while DTW method measures the distance of songs for sequential features, like pitches. DTW has a quadratic time and space complexity that limits its use to only small time series data. Thus, we apply FastDTW [8] to accelerate the similarity measurement to O(n) time.

Category	Features
Acoustic features	bars, beats, sections, segment, **loudness**, **pitches**, timbre, tatums, key, mode and **tempo**
Song metadata	sample rate, **duration**, release, **song hotness**, title and year
Artist metadata	terms, **similar artists**, **artist hotness**, **artist familiarity**, artist location, artist name and musicbrainz tags

Table 3. Features provided by MSD (Bold features are selected)

Play count is considered as an implicit rating based on the assumption that the rating is positive if the song is played many times, and vice versa. Considering the bias of the ratings, the normalized rating $r'_{i,j}$ from -1.0 to 1.0 is given by:

$$r_i^{j\prime} = r_i^j - \frac{1}{2}\left(\bar{r}_{i\cdot} + \bar{r}_{\cdot}^{j}\right), \tag{9}$$

where $\bar{r}_{i\cdot}$ is the average rating of all ratings to song s_i and $\bar{r}_{\cdot}^{j}$ denotes the average rating by user u_j.

We evaluate the feature importance by the correlation between predicted scores and normalized scores as described in Section 2.1. As all non-sequential features but user preference could represent a song to a vector, they are evaluated by feature selection methods mentioned in Section 2.2. The numeric value of a feature is assigned into categories by predefined window to apply feature selection methods. The normalized rating less than -0.1 is seen as "dislike" while the song is beloved if the rating greater than 0.1. Because the ratings between -0.1 and 0.1 is blur to classify the song to like or dislike, these ratings are ignored. The evaluation result is normalized to [0.0, 1.0] in order to compare the results among different selection methods.

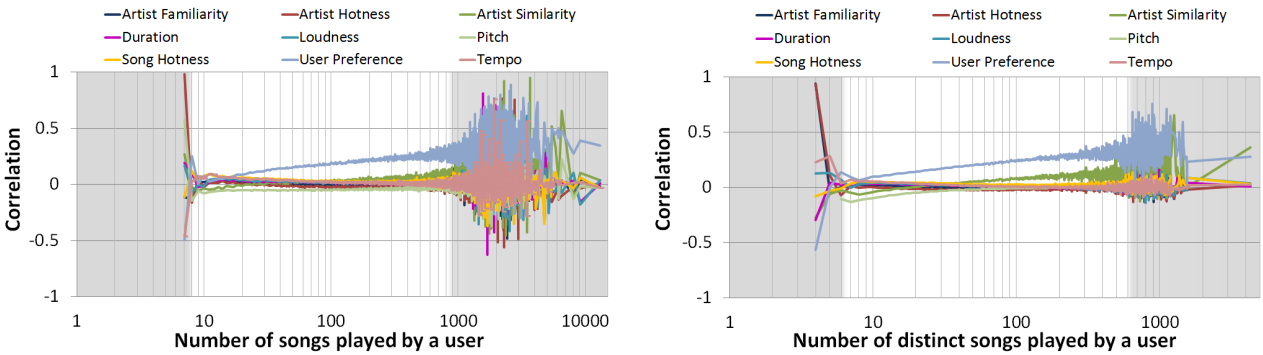
3.3 Result And Discussion

The value of each plot in Figures 3 and 4 is the mean of the results by different users who play the same number of songs. Figure 3 shows the correlation between the predicted ratings and the normalized ratings by play count in two views, including nine features. Figure 3(a) shows the distribution of the correlation by the overall played songs. In Figure 3(b), the correlation varies by the distinct songs, which means that the available information about songs increases as the distinct songs increases.

In Figure 3, the gray shadow covers the number of played songs by which there are fewer than 30 users so that the correlation results in the shadow are not statistically reliable. In the remaining part, user preference similarity is the most important feature except at the cold start, namely, CF technique is remarkable for favorite song detection. Basically, artist similarity is the secondary important feature. The other curves fluctuate around 0.0, which means

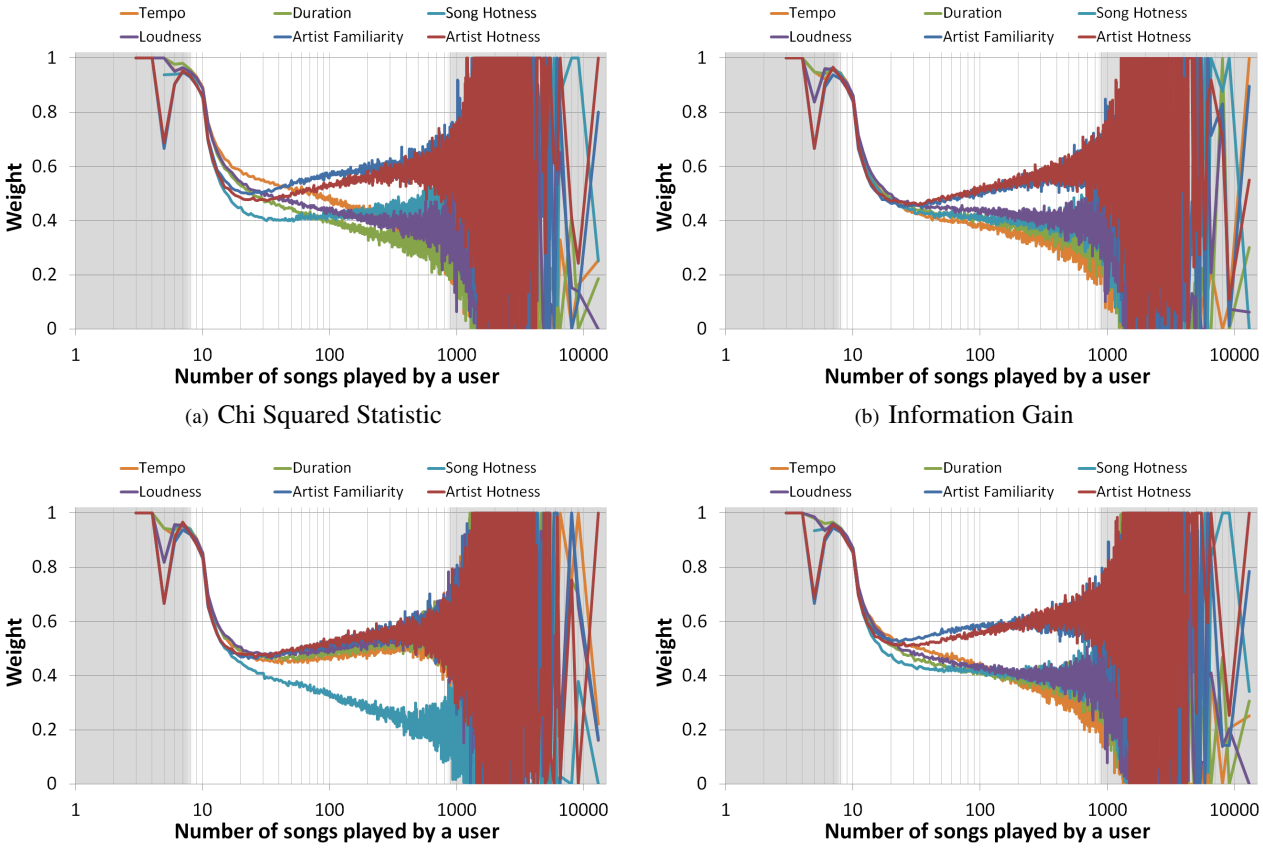
Dataset	Taste Profile Dataset	Last.fm dataset 360K	Yahoo! Music
Number of songs	**384,546**	294,015 artists	∼136,000
Number of users	1,019,318	359,347	**∼1,800,000**
Number of triplets	48,373,586	17,559,530	**∼717,000,000**
Rating type	play count	play count	**rate**
Other information	**features and metadata provided by MSD**	metadata provided by Last.fm	genre

Table 2. Available dataset comparison



(a) Based on overall songs (b) Based on distinct songs

Figure 3. Correlation between the predicted ratings and the normalized ratings



(a) Chi Squared Statistic (b) Information Gain

(c) Information Gain Ratio (d) Uncertainty

Figure 4. Feature selection results by different feature selection methods

the corresponding features are not critical for favorite song detection.

At the cold start period, it is frustrating that the selected features don't make a great contribution to favorite song detection. As the number of played songs increases, the number of users reduces, and the fluctuation of curves becomes vast but the main trend doesn't change significantly. Thus, in a long playing period, the importance of user preference is basically stable.

We leverage feature selection methods to measure the feature importance, and the feature selection results are normalized to [0.0, 1.0]. Zero means the feature doesn't play an important role and one means the feature is crucial. Figure 4 presents the evaluation results. The gray shadow also covers the region that the number of users are less than 30.

When the number of played songs is less than 10, most features are highly effective to distinguish "like" and "dislike". Then, the curves remarkably separate from each other. As the play counts increase, artist hotness and artist familiarity rise and other features descend more or less except Figure 4(c). Referring to Figure 2, the importance of features varies by the overall duration. In a long period, the user preferences would be changed by different contexts. As a result, the importance of tempo, duration, song hotness and loudness becomes weak while that of artist familiarity and artist hotness grows. Therefore, artist familiarity and artist hotness are more stable in a long listening period than other features.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we compare nine features by correlation and feature selection methods. Among these features, user preference and artist similarity play important roles in favorite song detection. Artist familiarity and artist hotness are stable in a long listening period. The evaluation result shows that collaborative filtering technique has high performance for favorite song detection. Song hotness is not as important as people thought.

If audio files of songs are available, we will employ a feature extraction tool to gain more features, and compare them in the future. If more information on the user log is exposed, such as time stamps and sequence information, the preference variation model is expected to be analyzed.

5. REFERENCES

[1] Pedro Cano, Markus Koppenberger, and Nicolas Wack. An industrial-strength content-based music recommendation system. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 673–673, 2005.

[2] Z. Cataltepe and B. Altinel. Music recommendation based on adaptive feature and user grouping. In *22nd international symposium on Computer and information sciences.*, pages 1–6, 2007.

[3] Oscar Celma. *Music recommendation and discovery, the long tail, long fail, and long play in the digital music space*. Springer, Berlin, 2010.

[4] Justin Donaldson. A hybrid social-acoustic recommendation system for popular music. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 187–190, 2007.

[5] George Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, March 2003.

[6] Keiichiro Hoashi, Kazunori Matsumoto, and Naomi Inoue. Personalization of user profiles for content-based music retrieval based on relevance feedback. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 110–119, 2003.

[7] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 195–202, 2009.

[8] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[9] U. Shardanand. Social information filtering for music recommendation. Master's thesis, Massachusetts Institute of Technology, 1994.

[10] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 210–217, 1995.

[11] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okumo. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. *12th International Society for Music Information Retrieval Conference*, pages 103–108, 2011.