A Hybrid Music Recommendation System by M-LSA

Bin Hu
College of Computer Science
Beijing University of Technology
Beijing, China
hubin0514@gmail.com

Meng Guo
College of Computer Science
Beijing University of Technology
Beijing, China
guome@126.com

Hongbin Zhang
College of Computer Science
Beijing University of Technology
Beijing, China
zhb@public.bta.net.cn

Abstract—**In this paper, a hybrid music recommendation system is proposed, which combines collaborative filtering and content-base recommendation. Neither of these two parts can make full use of all the information. Our method integrates both user rating and music content information using an expansion method of LSA (Latent Semantic Analysis) called M-LSA. We use a text representation for music content information, which is obtained by K-means Clustering or HMM method. Experiments on the data of 300 popular songs show that the proposed approach achieves satisfactory results.**

Keywords-music recommendation, collaborative filtering, hybrid system, text representation, M-LSA

I. INTRODUCTION

With the booming of digital multimedia data collections on the internet, it is difficult for people to find the resource they may interested in from a huge database. This is called an information overload. Then we need a service for people to find the resource they want. Recommendation system is an important system for people to overcome the information overload.

Collaborative filtering is a widely used technology in music recommender domain which works by building a database of preferences for songs by users. The user-based and the item-based collaborative filtering are typical methods of this technology found neighbors similar to the user or the item. After several years' develop, the performance of this technology can hardly be improved.

LSA [1] was imported from text categorization problem to recommendation system to alleviate the data sparseness problem by representing objects in a low-dimensional semantic space. However, the effect of LSA is rather limited since it can only consider the co-occurrence relationship between two types of objects. Therefore, several recent studies utilize extra co-occurrence data for enriching the information we have. M-LSA [2] is an expansion method of LSA which can deal with multiple types of objects.

In this paper a hybrid music recommendation system by the M-LSA method is described that music content information can be utilized with the rating data. We use a text representation [3,4] for music content information which has been successfully used in music genres classification and music summary.

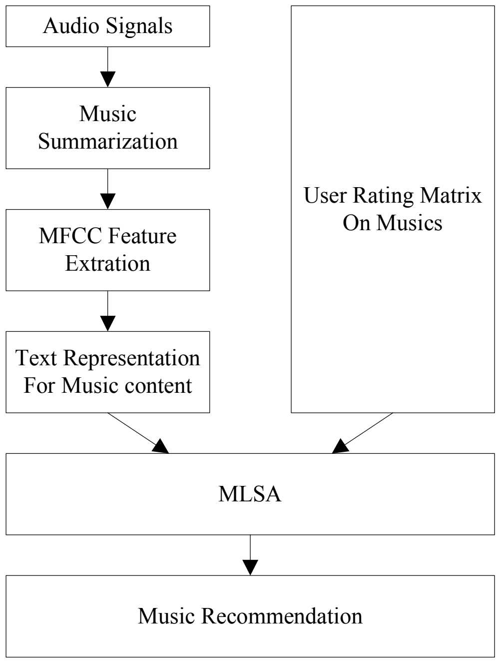

Figure 1. Flowchart of music recommendation.

Our framework is presented in Figure 1. The framework includes two parts: text representation for music content and application of M-LSA. We first use an automatic summarization system [5] to deal with the songs. Then the mel-frequency cepstral coefficients (MFCC) will be used for the feature set. After that, we represent music content in text form. At last, the music content information and the user rating information can be integrated by M-LSA.

The rest of the paper is organized as follows: Section 2 introduce the M-LSA algorithm in detail. Text representation for music content information is described in Section 3. In section 4, the experimental results are given, followed by a conclusion in Section 5.

II. M-LSA ALGORITHM

In this section, we first describe the traditional LSA. We then introduce the expansion method of LSA, M-LSA [2], which can deal with multiple types of objects. In this paper, it means that we can integrate the music content information and the user rating.

A. The LSA Algorithm

Latent Semantic Analysis [1] is a method to alleviate the data sparseness problem by representing objects in a low-dimensional semantic space. It is based on a

978-0-7695-3645-3/09 $25.00 © 2009 IEEE
DOI 10.1109/CINC.2009.74

mathematical operation, Singular Value Decomposition (SVD).

In the user-song rating data, consider there are m users and n songs, the rating matrix between the users and songs $A=[a_{ij}]$, with each entry a_{ij} representing the rating of i-th user for j-th song. For the $m \times n$ matrix A, where without loss of generality $m \leq n$ and $rank(A)=r$, the SVD is defined as [6]:

$$A = U\Sigma V^T \qquad (1)$$

where $U=[u_1,u_2,\cdots,u\ ]$ is an $m \times r$ column-orthonormal matrix whose columns are called left singular vectors; $\Sigma = diag[\sigma_1,\sigma_2,\cdots,\sigma_r]$ is an $r \times r$ diagonal matrix whose diagonal elements are positive singular values sorted in descending order. $V=[v_1,v_2,\cdots,v_r]$ is an $n \times r$ column-orthonormal matrix whose columns are called right singular vectors.

Given an integer $k(k << r)$, LSA uses the first k singular vectors to represent the users and songs in a k-dimensional space [1]. LSA represents each user by a row of $[\sigma_1 u_1,\cdots,\sigma_k u_k]$ and each song by a row of $[\sigma_1 v_1,\cdots,\sigma_k v_k]$.

In LSA, the most important k concepts in the rating data can be represented by the first k singular vectors of A. It can be explained by the mutual reinforcement principle [2] which associates important property for both users and songs.

B. The M-LSA Algorithm

M-LSA is an expansion method of LSA, which conducts latent semantic analysis among multiple types of objects. Similar to LSA, the mutual reinforcement principle can provide a reasonable solution to find the important latent concepts among the multiple co-occurrence data by extend it to multiple-type graph. The principle can be expressed as:

$$w_i \doteq \sum_{\forall j: j \neq i} M_{ij} w_j \ . \qquad (2)$$

We use $w=[w_1,\cdots,w_N]^T$ as the concatenated importance vector and define

$$R = \begin{bmatrix} 0 & M_{12} & \cdots & M_{1N} \\ M_{21} & 0 & \cdots & M_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ M_{N1} & M_{N2} & \cdots & 0 \end{bmatrix} \qquad (3)$$

as the unified co-occurrence matrix. We can rewrite Equation (2) in a matrix format:

$$w \doteq R \cdot w. \qquad (4)$$

Since w will converge to the eigenvector of the co-occurrence matrix R, the first k eigenvectors of R represent the top k important concepts, which span a k-dimensional semantic space to represent all the objects. Specifically, let

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$$

be the top k eigenvalues of R and the corresponding vectors are respectively

$$c_1, c_2, \cdots, c_k.$$

Therefore, the i-th object can be represented by

$$[\lambda_1 c_{1i}, \lambda_2 c_{2i}, \cdots, \lambda_k c_{ki}].$$

III. Text Representation For Music Content

As mentioned above, we need extra information for music recommend. LSA use a text representation for the rating data, so we need to represent the music content information in the same way for the M-LSA algorithm.

Different music signal has special temporal structure composing of different temporal components. Similar temporal components can appear in different music pieces. We call these similar temporal components as music symbols [3]. According to this, we convert music signals to music symbol sequences and apply music semantic description. Each piece of music can be considered as a text-like semantic music document, which is shown in Figure 2.

In our experiments, we first use an automatic summarization system [5] to deal with the songs. After that, each song can be represented by a fixed length of music segment. Then we use the mel-frequency cepstral coefficients (MFCC) for the feature set, which has been used with great success for application in speech processing and audio processing. As shown in Figure 2, the process of text representation for music content is a standard process of Vector Quantization (VQ). Therefore, we use two individual methods to recognize the music symbols: one is the clustering algorithm and another is the HMM approach.

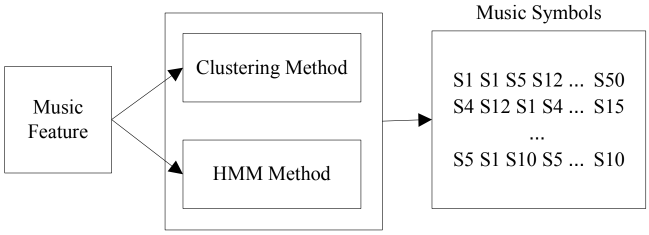

Figure 2. Flowchart of text representation for music.

After recognizing the music symbols, we can represent the music content in text from. Then, we can integrate it with the M-LSA algorithm. A matrix representation of training music document is created first. For example, we have two short music documents:

D1 = “S1 S2 S3”

D2 = “S1 S3 S3 S4”

TABLE I. MUSIC SYMBOL MATRIX

	S1	S2	S3	S4
D1	1	1	0	1
D2	1	0	2	1

Table I shows how many times music symbols appear in a music document.

A. The Clustering Algorithm

The first method to find the music symbols is the clustering algorithm. We use the widely used K-means clustering algorithm to find the music symbols. The given songs after the feature extraction are divided into fixed length frames. After given the number of clustering centers *k*, each frame can be assigned to a cluster or a music symbol in this paper.

B. The HMM Algorithm

The second method is the HMM algorithm. The HMM is a powerful framework for learning and recognition of temporal patterns and has applied in many pattern recognition applications such as speech recognition. An HMM is usually trained based on labeled training examples. But in our case, no labeled training data is available. Therefore, we use an unsupervised learning of ergodic HMMs, which can find the segmentation from the songs itself.

An ergodic HMM is a fully connected finite-state machine. In our case, each state represent a group of similar temporal segments in the songs and then each state of the HMM can be seemed as a music symbol. We use the Gauss Mixture probability density function for continues HMM parameters. The unsupervised Baum-Welch algorithm [7] is used for the training of the HMMs. And then the Viterbi decoding method is used to determine the most likely state sequence for music. Each state in HMM represent a music symbol.

Comparing with the K-means clustering method, HMM clusters data on the temporal level rather than the frame level. It also gives an accurate probabilistic description of music data by the mixture of GMM.

IV. EXPERIMENTS

In this section, the proposed music recommendation system is evaluated on a test database of 300 popular songs rated by 272 users. The rating value is from 1 to 5. After the music summarization, the length of each song is 15 seconds, and each song is stored as 16k Hz, 16 bit, mono. We randomly select 80% ratings for training and the left 20% ratings for testing.

Because the sparseness of our database achieves 20%, far larger then the real sparseness of other database from the music website, 3%, we split the database into different sparseness. So we can compare the performance of different methods in different sparsenesses.

The text representation of music content divides music data into non-overlapping 0.25 second segments. A mel-scaled filter bank consisting of 40 bandpass filters is used to calculate the cepstral coefficients. 13 cepstral coefficients are used in the clustering method and the HMM approach. In the HMM algorithm, each state is modeled by three Gaussian models.

Until now, we have three types of object: users (X_1), songs (X_2) and music content (X_3). The co-occurrence matrices include user-song (M_{12}), user-content (M_{13}) and song-content (M_{23}). M_{12} can get from the music database which are the ratings of users on songs; M_{23} is constructed from the clustering algorithm or HMM algorithm introduced in section 3. M_{13} can be constructed as follows: if a music symbol appears in a song rated by a user, this music symbol and this user has a co-occurrence (corresponding element in M_{13} plus 1). We give weight parameters to M_{12} , M_{13} and M_{23} . Then the relation matrix *R* based on the three matrices can be constructed as follows:

$$R = \begin{bmatrix} 0 & \alpha M_{12} & \beta M_{13} \\ \alpha M_{12}^T & 0 & \gamma M_{23} \\ \beta M_{13}^T & \gamma M_{23}^T & 0 \end{bmatrix} \quad (5)$$

where the weight parameters $\alpha, \beta, \gamma \geq 0$, and $\alpha + \beta + \gamma = 1$. Here we set $\alpha = 0.2$, $\beta = 0.5$, and $\gamma = 0.3$.

We use the Mean Absolute Error (MAE) [8] for the evaluation criterion in our experiments. The MAE has widely been used in evaluating the accuracy of a recommender system. It is calculated by averaging the absolute errors in rating-prediction pairs as follows:

$$MAE = \frac{\sum_{j=1}^{N} \left| P_{i,j} - R_{i,j} \right|}{N} \quad (6)$$

Where $P_{i,j}$ is system's rating (prediction) of item *j* for user *i*, and $R_{i,j}$ is the rating of user *i* for item *j* in the test data. *N* is the number of rating-prediction pairs between the test data and the prediction result. A lower MAE indicates greater accuracy.

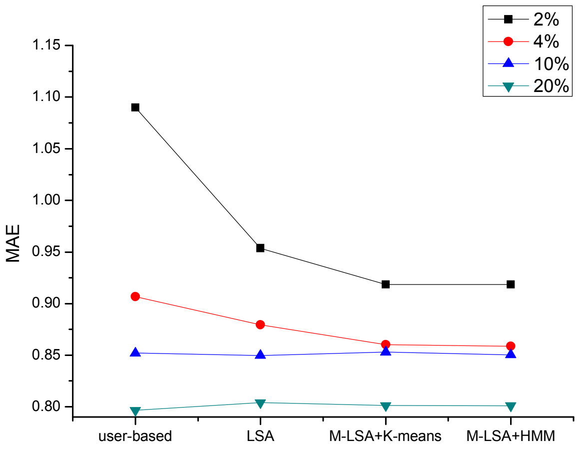

Figure 3. Recommendation result comparison of different methods in defferent sprsenesses.

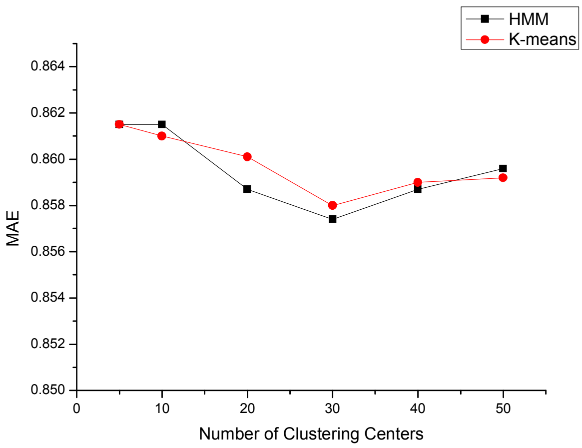

Figure 4. Impact of number of clustering centers.

Figure 3 compares the results of different methods. The figure shows that our hybrid method achieves better result than the traditional methods. The LSA method only achieve a little improvement compared with the user based collaborative filtering method. But after integrating the music content, our method achieves satisfactory results. The later two columns shows that M-LSA with the common K-means method achieves almost the same result as the M-LSA with the HMM method. It also compares the performance of different methods in different sparsenesses, which shows that our method play a greater role in the lower sparseness.

We also study the impact of the number of clustering centers (number of states in HMM). In Figure 4, it shows that both methods with K-means and HMM achieve the best MAE when the number of clustering centers is 30.

V. Conclusion

In this paper, we presented a hybrid music recommendation system which integrates both user rating and music content information using the M-LSA method. Experimental results have indicated that the proposed approach achieves satisfactory results. In future, we will try to use other music feature extraction methods to evaluate our scheme.

References

[1] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis", Journal of the American Society for Information Science and Technology (JASIS), 41(6), pp.391–407, 1990.

[2] X. Wang, J. Sun, Z. Chen, and C. Zhai, "Latent semantic analysis for multiple-type interrelated data objects", in Proc. SIGIR, 2006, pp.236-243.

[3] K. Chen, S. Gao, Y. Zhu, and Q. Sun, "Music Genres Classification using Text Categorization Method", Multimedia Signal Processing, 2006, pp.221-224.

[4] Stephen. Chu, Beth. Logan, "Music summary using key phrases", http://www.hpl.hp.com/techreports/Compaq-DEC/CRL-2000-1.pdf

[5] Meng. Guo, Hongbin. Zhang, "Automatic Summarization for Popular Songs", ICALIP 2008, pp.653-658.

[6] G. H. Golub and C. F. V, Loan. Matrix Computations, third edition, The Johns Hopkins University Press, 1996.

[7] Steve. Young, Phil. Woodland and Gunnar. Evermann, HTK Book, http://htk.eng.cm.ac.uk, 2005.

[8] Breese. J. S, Heckerman. D, and Kardie. C, "Empirical analysis of predictive algorithms for collaborative filtering". In Proceedings of the 14th UAI, pp.43–52.