Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content

Jiajun Bu[1], Shulong Tan[1], Chun Chen[1], Can Wang[1], Hao Wu[1], Lijun Zhang[1], Xiaofei He[2]
{bjj, laos1984, chenc, wcan, haowu, zljzju}@zju.edu.cn, xiaofeihe@cad.zju.edu.cn

[1]Zhejiang Key Laboratory of Service Robot
College of Computer Science, Zhejiang University
Hangzhou, China, 310027

[2]State Key Laboratory of CAD&CG
College of Computer Science, Zhejiang University
Hangzhou, China, 310027

ABSTRACT

Acoustic-based music recommender systems have received increasing interest in recent years. Due to the semantic gap between low level acoustic features and high level music concepts, many researchers have explored collaborative filtering techniques in music recommender systems. Traditional collaborative filtering music recommendation methods only focus on user rating information. However, there are various kinds of social media information, including different types of objects and relations among these objects, in music social communities, such as Last.fm and Pandora. This information is valuable for music recommendation. However, there are two challenges to exploit this rich social media information: (a) There are many different types of objects and relations in music social communities, which makes it difficult to develop a unified framework taking into account all objects and relations. (b) In these communities, some relations are much more sophisticated than pairwise relation, and thus cannot be simply modeled by a graph. In this paper, we propose a novel music recommendation algorithm by using both multiple kinds of social media information and music acoustic-based content. Instead of graph, we use hypergraph to model the various objects and relations, and consider music recommendation as a ranking problem on this hypergraph. While an edge of an ordinary graph connects only two objects, a hyperedge represents a set of objects. In this way, hypergraph can be naturally used to model high-order relations. Experiments on a data set collected from the music social community Last.fm have demonstrated the effectiveness of our proposed algorithm.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*; H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*methodologies and techniques*

General Terms

Algorithms, Experimentation.

Keywords

Recommender systems, music recommendation, hypergraph, social media information

1. INTRODUCTION

As the World Wide Web becomes the source and distribution channels of diverse digital music, a large amount of music tracks are accessible to people. Since it is usually difficult and time consuming for a user to find and choose his/her desired music, the music recommender system becomes an indispensable tool [26]. Music recommendation is valuable in many real world applications, such as social music communities, online music stores, and some music devices (e.g. PCs and MP3 players) where music recommendation can be used to generate music playlists.

For the tasks of music recommendation, the most common approach is to analyze the audio signal directly. These methods are called acoustic-based music recommendation [22, 9, 8, 30]. Due to the semantic gap between low level acoustic features and high level music concepts [10], the results of acoustic-based music recommendation are not satisfactory. It is necessary to consider more information in the recommender systems [11]. Some researchers try to utilize the user rating information by applying collaborative filtering methods [41, 20, 38, 40]. There is also work which exploits the information in the meta data (e.g., genre) associated with music tracks [4, 28, 27]. However, all these approaches only utilize limited kinds of information, without considering rich social media information.

In typical music social communities, such as Last.fm[1] and Pandora[2], there is rich social media information including various types of objects and relations among these objects. Fig. 1 shows an example of Last.fm. In Last.fm, each user can make friends with other users, join groups, listen to music tracks, and use tags to bookmark resources like music tracks, albums and artists. There are also some relations among resources, such as inclusion relations between tracks and albums. Additionally, similarity relations between music tracks can be computed based on audio content.

The various social media information mentioned above is

[1]http://www.last.fm
[2]http://www.pandora.com

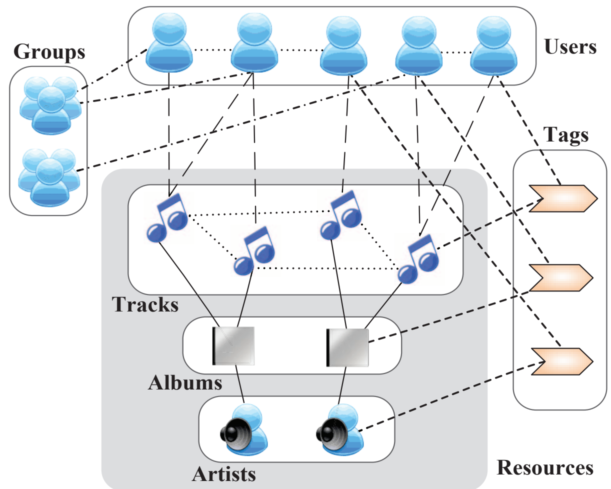


Figure 1: Various types of objects and relations in the social music community Last.fm. The relations include friendship relations, membership relations, listening relations, tagging relations, inclusion relations among resources (e.g., tracks and albums) and similarity relations between music tracks.

very useful for music recommendation. A key step in recommender systems is to build the users' preference profiles [42] which may be inferred from their multi-type actions on resources, such as rating and tagging. Music similarity relations can be used in music recommendation by recommending similar music to each user. Moreover, a user's interest may be affected by his/her friends [19, 24].

However, there are two major challenges to exploit all this information. First, there are many different types of objects and relations in social media communities, which makes it difficult to develop a unified framework taking into account all objects and relations simultaneously. Second, in social media communities, some relations are beyond pairwise. For example, more than two users join in the same group, or a user bookmarks a music track by a tag. We call this kind of relation *high-order* relation. Traditional methods [45, 2] which deal with pairwise relations cannot properly model these high-order relations.

Traditional recommendation algorithms, such as *Collaborative Filtering* (*CF*) [29, 16, 21], only consider the user-item rating matrix and fail to take advantage of other kinds of social media information. Recently, there has been considerable interest in making use of social media information to enhance the recommendation performance [39, 36, 19, 24, 33, 43]. For example, some previous works employed ordinary graphs to model tagging data for recommendation problems [19, 43]. Fig. 2(a) shows a simple example of using ordinary graph to model the tagging relations. There are three tagging relations: u_1 bookmarks resources r_1 and r_2 with tags t_1 and t_2, respectively, and u_2 bookmarks resource r_1 with tag t_2. Fig. 2(b) shows our unified hypergraph approach for modeling the tagging relations. In our unified hypergraph model, the high-order relations among the three types of objects can be naturally represented as triples: (u_1, t_1, r_1), (u_1, t_2, r_2), and (u_2, t_2, r_1). Clearly, the ordinary graph model fails to capture the tagging relations

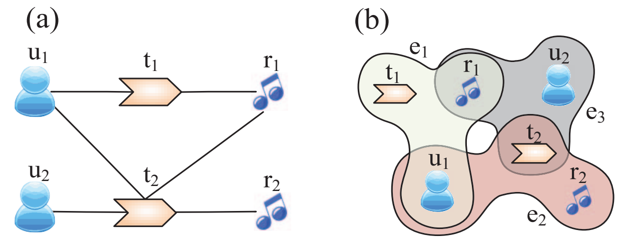


Figure 2: Tagging relations represented in two models: (a) ordinary graph model, and (b) our unified hypergraph model. This hypergraph contains six vertices and three hyperedges, i.e., (u_1, t_1, r_1), (u_1, t_2, r_2), and (u_2, t_2, r_1).

precisely. For example, from Fig. 2(a), it is unclear whether u_2 bookmarks r_1, r_2, or both.

In this paper, we use unified hypergraphs to model multi-type objects and relations in music social communities. Similarities between music tracks based on acoustic signals are treated as one kind of relations. In this way, we combine acoustic-based and collaborative filtering recommendation in a unified framework. A hypergraph is a generalization of the ordinary graph in which the edges, called *hyperedges*, are arbitrary non-empty subsets of the vertex set [3]. Each vertex of the hypergraph corresponds to an object of any type. The hyperedges are used to model high-order relations, as shown in Fig. 2(b). By using the unified hypergraph model, we can accurately capture the high-order relations among various types of objects without loss of any information. We further consider music recommendation as a ranking problem on this hypergraph to find the music tracks that the user desires.

The points below highlight the contributions of this paper:

1. Multi-source media fusion. We integrate multi-source media information, including multiple kinds of social media information and music acoustic signals, in music recommendation to improve the performance.

2. We propose to model high-order relations in social media information by hypergraph instead of traditional graph. In this way, there is no information loss in representing various types of relations.

3. We empirically explore the contributions of different types of social media information to recommendation performance. Our results are helpful for practical music recommender systems.

The rest of this paper is organized as follows. Section 2 reviews the related work. In Section 3, we introduce the formal definition of the problem and describe how to perform ranking on the hypergraph. In Section 4, we discuss how to apply hypergraph ranking in music recommendation. Extensive experimental results are presented in Section 5. We conclude our paper and provide suggestions for future work in Section 6.

2. RELATED WORK

In this paper, we combine acoustic-based and collaborative filtering music recommendation methods to exploit rich social media information and acoustic-based content information using hypergraph based learning techniques. Our work is related to hybrid music recommendation, recommendation using social media information and graph/hypergraph-based learning. In this section we provide a brief review of these works.

2.1 Hybrid music recommendation

There are several hybrid approaches combining acoustic-based and collaborative filtering music recommendation to improve the overall accuracy of predictions [41, 20, 38, 14, 40]. Yoshii et al. [41, 40] integrate both rating and music content information by using probabilistic models. Unobservable user preferences are directly represented by introducing latent variables. Li et al. [20] propose an item-based probabilistic model utilizing audio features to capture accurate similarities among items (i.e., music). Tiemann et al. [38] investigate ensemble learning methods for hybrid music recommendation. They apply ensemble learning methods to combine outputs of item-based collaborative filtering and acoustic-based recommendation. Donaldson [14] exploits music co-occurring information in playlists and acoustic signals for a hybrid music recommender system by unifying spectral graph and acoustic feature vectors. All the above works use conventional collaborative filtering methods and only utilize limited kinds of information, without considering more sophisticated social media information.

2.2 Recommendation Using Social Media Information

It has been shown that social media information, such as tagging relations and friendship relations, is valuable for recommendation. Tso-Sutter et al. [39] reduce three types of objects in tagging relations (users, resources and tags) to two types by treating tags as either users or resources, and then apply traditional item-based or user-based collaborative filtering algorithms [1], respectively. Diederich et al. [13] introduce TF-IDF tag profiles for the users and use these profile vectors to measure user-user similarities in the use-based CF algorithm. Zhang et al. [43] propose a recommendation algorithm by integrating diffusion on user-tag-item tripartite graphs . Ma et al. [24] propose a probabilistic factor analysis framework which naturally fuses the users' tastes and their trusted friends' favors together. To utilize both friendship and tagging relations, Konstas et al. [19] create a collaborative recommender system which constructs a social graph over users, tags and resources. Sen et al. [33] address resource recommendation by inferring users' tag preferences firstly and then compute resource item preferences based on tag preferences. They propose some heuristic methods to make use of various social media information, such as click-through and search information, in the step of tag preferences generation. Knees et al. [18] utilize web-based musical artist similarity information to reduce the number of necessary acoustic-based music similarity calculations and then use music similarity in the task of music playlist generation.

Although the above approaches have achieved great success in resource recommendation applications, they fail to make full use of the high-order relations in the social media communities. In this work, we propose to use hypergraph, rather than the ordinary graph, to precisely capture the high-order relations and hence enhance the recommendation performance.

2.3 Graph-based Ranking and Hypergraph

Our work is also related to graph-based ranking and hypergraph learning [2, 3, 35, 44, 12, 45].

Zhou et al. propose a manifold ranking algorithm which ranks data objects with respect to the intrinsic geometrical structure in the data [45]. They first construct a weighted graph and set the query point, then let all data points spread their ranking scores to their nearby neighbors via the weighted graph. The spread process is repeated until a global stable state is achieved. Agarwal [2] proposes to model the data objects as a weighted graph, and incorporate this graph structure into the ranking function as a regularizer. In this way, the obtained ranking function varies smoothly over the graph. To generate personalized tag recommendation, Guan et al. propose a graph-based ranking algorithm for interrelated multi-type objects [15].

Recently, there has been a lot of interest in learning with hypergraph [3, 7]. Bulò et al. introduce a hypergraph clustering algorithm to extract maximally coherent groups from a set of objects using high-order (rather than pairwise) similarities [7]. Zhou et al. develop a general framework which is applicable to classification, clustering and embedding on hypergraph data [44]. These studies only focus on classification, clustering and embedding on hypergraphs. However, by modeling the multiple types of social media objects and their relations as a hypergraph, we consider music recommendation as a ranking problem on hypergraph.

3. RANKING ON UNIFIED HYPERGRAPH

In this section we discuss how to model various types of objects and their relations in a unified hypergraph model and how to perform ranking on unified hypergraph. We begin with the description of the problem and the notations.

3.1 Notation and Problem Definition

Let $G(V, E, w)$ denote a hypergraph where V is the set of vertices, E is the set of hyperedges, and w is a weight function defined as $w : E \to \mathbb{R}$. Each hyperedge $e \in E$ is a subset of V. The degree of a hyperedge e is defined by $\delta(e) = |e|$, that is, the cardinality of e. If every hyperedge has a degree of 2, the hypergraph reduces to an ordinary graph. The degree $d(v)$ of a vertex v is $d(v) = \sum_{e \in E | v \in e} w(e)$. We say that there is a *hyperpath* between vertices v_1 and v_k if there is an alternative sequence of distinct vertices and hyperedges $v_1, e_1, v_2, e_2, ..., e_{k-1}, v_k$, such that $\{v_i, v_{i+1}\} \subseteq e_i$ for $1 \le i \le k-1$. A hypergraph is *connected* if there is a hyperpath for every pair of vertices [44]. We define a vertex-hyperedge incidence matrix $\mathbf{H} \in \mathbb{R}^{|V| \times |E|}$ whose entry $h(v, e)$ is 1 if $v \in e$ and 0 otherwise. Then we have:

$$d(v) = \sum_{e \in E} w(e) h(v, e), \quad (1)$$

$$\delta(e) = \sum_{v \in V} h(v, e). \quad (2)$$

Let $\mathbf{D}_e$ and $\mathbf{D}_v$ be two diagonal matrices consisting of hyperedge and vertex degrees, respectively. Let $\mathbf{W}$ be a $|E| \times |E|$ diagonal matrix containing hyperedge weights.

In the following, we define *unified hypergraph* which will be used to model the high-order relations among different types of objects. A unified hypergraph is a hypergraph that has multi-type vertices and hyperedges. Suppose a unified hypergraph has m types of vertices and n types of hyperedges. The vertex set of the i-th type is denoted by $V^{(i)}$ and the hyperedge set of the j-th type is denoted by $E^{(j)}$. We define $V = \bigcup_{i=1}^{m} V^{(i)}$ and $E = \bigcup_{j=1}^{n} E^{(j)}$. In social music communities, different kinds of objects, such as users, tags, resources and groups, can be viewed as different types of vertices in a unified hypergraph, and different types of relations among objects can be viewed as different types of hyperedges. A hyperedge in unified hypergraph can be a set of vertices with either the same type or different types. The former kind of hyperedge captures the relations among the same type of objects, while the latter one captures the relations across different types of objects.

The problem of ranking on unified hypergraph is addressed in a "query and ranking" manner as follows. Given some query vertices from V, rank the other vertices on the unified hypergraph according to their relevance to the queries. Let $\mathbf{y} = [y_1, y_2, \cdots, y_{|V|}]^T$ denote the query vector and y_i, $i = 1, \cdots, |V|$, denote the initial score of the i-th vertex. We will discuss how to set the query vector in detail in Section 4.4. Similarly, let $\mathbf{f} = [f_1, f_2, \cdots, f_{|V|}]^T$ be the vector of ranking scores.

3.2 Regularization Framework for Ranking on Unified Hypergraph

There are many existing algorithms for learning on hypergraph [3, 7, 44, 12]. However, most of them focus on classification, clustering, and Euclidean embedding. In this subsection, we discuss how to perform ranking on unified hypergraph by using similar idea of [44].

The cost function of $\mathbf{f}$ is defined as follows:

$$Q(\mathbf{f}) = \frac{1}{2}\sum_{i,j=1}^{|V|}\sum_{e\in E}\frac{1}{\delta(e)}\sum_{\{v_i,v_j\}\subseteq e} w(e)\left\|\frac{f_i}{\sqrt{d(v_i)}} - \frac{f_j}{\sqrt{d(v_j)}}\right\|^2 + \mu\sum_{i=1}^{|V|}\|f_i - y_i\|^2, \tag{3}$$

where $\mu > 0$ is the regularization parameter. The optimal ranking result is achieved when $Q(\mathbf{f})$ is minimized:

$$\mathbf{f}^* = \arg\min_{\mathbf{f}} Q(\mathbf{f}). \tag{4}$$

The first term of the right-hand side in Eq.(3) is the smoothness constraint. Minimizing it means that vertices should have similar ranking scores if they are contained in many common hyperedges. For instance, if two music tracks are listened by many common users, they will probably have similar ranking scores. Another example is the ranking of the users. If two users join in many common interest groups (or if they listen to many common music tracks, etc.), they will probably have similar ranking scores. The second term measures the difference between the obtained ranking scores and the pre-given scores. The parameter μ controls the relative importance of these two terms. Note that each hyperedge is normalized by its degree $\delta(e)$, that is, the number of vertices contained in this hyperedge. In this way, the hyperedges with different sizes will be equally treated.

The first term of the right-hand side in the cost function (3) can be rewritten as follows:

$$\begin{aligned}
& \frac{1}{2}\sum_{i,j=1}^{|V|}\sum_{e\in E}\frac{1}{\delta(e)}\sum_{\{v_i,v_j\}\subseteq e} w(e)\left\|\frac{f_i}{\sqrt{d(v_i)}} - \frac{f_j}{\sqrt{d(v_j)}}\right\|^2 \\
=\ & \frac{1}{2}\sum_{i,j=1}^{|V|}\sum_{e\in E}\frac{w(e)h(v_i,e)h(v_j,e)}{\delta(e)}\left\|\frac{f_i}{\sqrt{d(v_i)}} - \frac{f_j}{\sqrt{d(v_j)}}\right\|^2 \\
=\ & \sum_{i,j=1}^{|V|}\sum_{e\in E}\frac{w(e)h(v_i,e)h(v_j,e)}{\delta(e)}\left(\frac{f_i^2}{d(v_i)} - \frac{f_i f_j}{\sqrt{d(v_i)d(v_j)}}\right) \\
=\ & \sum_{i=1}^{|V|} f_i^2 \sum_{e\in E}\frac{w(e)h(v_i,e)}{d(v_i)}\sum_{j=1}^{|V|}\frac{h(v_j,e)}{\delta(e)} \\
& - \sum_{i,j=1}^{|V|}\sum_{e\in E}\frac{f_i w(e)h(v_i,e)h(v_j,e)f_j}{\sqrt{d(v_i)d(v_j)}\delta(e)} \\
=\ & \sum_{i=1}^{|V|} f_i^2 - \sum_{i,j=1}^{|V|}\sum_{e\in E}\frac{f_i w(e)h(v_i,e)h(v_j,e)f_j}{\sqrt{d(v_i)d(v_j)}\delta(e)} \\
=\ & \mathbf{f}^T\mathbf{f} - \mathbf{f}^T\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{f}.
\end{aligned} \tag{5}$$

We define a matrix

$$\mathbf{A} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}. \tag{6}$$

Then we can rewrite the cost function (3) in the matrix-vector form:

$$Q(\mathbf{f}) = \mathbf{f}^T(\mathbf{I} - \mathbf{A})\mathbf{f} + \mu(\mathbf{f} - \mathbf{y})^T(\mathbf{f} - \mathbf{y}).$$

Requiring that the gradient of $Q(\mathbf{f})$ vanish gives the following equation:

$$\frac{\partial Q}{\partial \mathbf{f}}|_{\mathbf{f}=\mathbf{f}^*} = (\mathbf{I} - \mathbf{A})\mathbf{f}^* + \mu(\mathbf{f}^* - \mathbf{y}) = 0\ .$$

Following some simple algebraic steps, we have

$$\mathbf{f}^* = \frac{\mu}{1+\mu}\Big(\mathbf{I} - \frac{1}{1+\mu}\mathbf{A}\Big)^{-1}\mathbf{y}. \tag{7}$$

We define $\alpha = 1/(1+\mu)$. Noticing that $\mu/(1+\mu)$ is a constant and does not change the ranking results, we can rewrite $\mathbf{f}^*$ as follows:

$$\mathbf{f}^* = \big(\mathbf{I} - \alpha\mathbf{A}\big)^{-1}\mathbf{y}. \tag{8}$$

It can be shown that the matrix $\mathbf{I} - \alpha\mathbf{A}$ is invertible. The proof is omitted due to space limitation. Note that, the matrix $\mathbf{I} - \alpha\mathbf{A}$ is highly sparse. Therefore, the computation can be very efficient.

4. MUSIC RECOMMENDATION ON HYPERGRAPH

In this section, we introduce our approach for Music Recommendation on Hypergraph (MRH).

4.1 Data Collection

To evaluate our algorithm, we have collected data from Last.fm in December 2009. Firstly, we collected the top 340 most popular artists, as well as the users who are interested in those artists. Adding all these users' friends, we obtained the candidate set of the users. Then we reduced the candidate set of users by restricting that each user has at least one friend within the set. The final user set is denoted by

Table 1: Objects in our data set.

Objects	Notations	Count
Users	U	2596
Groups	G	1124
Tags	Ta	3255
Tracks	Tr	16055
Albums	Al	4694
Artists	Ar	371

Table 2: Relations in our data set.

Relations	Notations	Count
Friendship relations	R_1	4503
Membership relations	R_2	1124
Listening relations	R_3	304860
Tagging relations on tracks	R_4	10936
Tagging relations on albums	R_5	730
Tagging relations on artists	R_6	36812
Track-album inclusion relations	R_7	4694
Album-artist inclusion relations	R_8	371
Similarities between tracks	R_9	-

U. We collected other objects and relations based on this user set. We downloaded all the groups in which these users join, and reduced the set of groups by ensuring that each group has at least five members in the final user set. The final group set is denoted by G. For resource objects and relations, we crawled each user's top 500 frequently played music tracks to form the candidate set of tracks. In order to get the inclusion relations among resources, we downloaded all corresponding artists and albums of all tracks in the candidate track set, and removed those albums that contain less than five tracks in the candidate track set. After that, we obtained the final sets of resources, i.e., track set, album set and artist set, denoted by Tr, Al, and Ar, respectively. We collected the tagging relations which are essentially triples, i.e., (user, tag, music track), (user, tag, music album) or (user, tag, artist). For each user, we downloaded all his/her tagging relations. We only kept those relations in which the resource is in Tr, Al or Ar obtained previously. The final set of tags is denoted by Ta. Finally, we downloaded the music files (in mp3 or wma formats) from the Web. The objects and relations used in our experiments are summarized in Table 1 and Table 2 respectively. Similarities between music tracks are computed based on music content.

4.2 Acoustic-Based Music Similarity

Acoustic measures of music similarity have been extensively studied in recent years [23, 37, 5, 25]. These algorithms mainly focus on several central problems: 1) what representative features to extract; 2) how to model the feature distributions of music; 3) how to measure the similarity between distribution models.

To compactly represent the music content, in this paper we derive features from Mel-frequency cepstral coefficients (MFCCs) [5]. MFCCs are prevalent in audio classification. A given music track is segmented into short frames and the MFCC is computed for each frame. Similar to [23], we use K-means to group all the frames of each track into several clusters. For all the clusters, the means, covariances, and weights are computed as the signature of the music track. To compare the signatures for two different tracks, we employ the Earth-Mover's Distance (EMD) [31].

4.3 Unified Hypergraph Construction

We take into account six types of objects and nine types of relations in the data set mentioned above. The objects include users, groups, tags and three types of resources (i.e., tracks, albums and artists). The relations are divided into four categories, social relations, actions on resources, inclusion relations among resources, and acoustic-based music similarity relations. Social relations include friendship relations and membership relations (e.g., an interest group), denoted by R_1 and R_2, respectively. Actions on resources involve four types of relations, i.e., listening relations (R_3), and tagging relations on tracks, albums and artists (R_4, R_5 and R_6). Inclusion relations among resources are the inclusion relations between tracks and albums, albums and artists (R_7 and R_8). Acoustic-based music similarity relations are denoted by R_9.

The six types of objects form the vertex set of the unified hypergraph. So $V = U \bigcup G \bigcup Ta \bigcup Tr \bigcup Al \bigcup Ar$. And there are nine types of hyperedges in the unified hypergraph, each corresponding to a certain type of relations, as listed in Table 2. We denote the hyperedge sets as $E^{(i)}$ corresponding to R_i, $i = 1, \cdots, 9$. The construction of the nine types of hyperedges is listed as follows:

- $E^{(1)}$: We build a hyperedge corresponding to each pairwise friendship and set the hyperedge weight to be 1.

- $E^{(2)}$: For each group, we build a hyperedge which contains vertices corresponding to all the users in this group, as well as the group itself. Note that, group itself is also an object. We set the hyperedge weight to be 1.

- $E^{(3)}$: For each user-track listening relation, we build a hyperedge containing the user and the music track. The weight $w(e_{ij}^{(3)})$ ($e_{ij}^{(3)} \in E^{(3)}$) is set to be the frequency that the user u_i listens to the track tr_j

  $$w(e_{ij}^{(3)}) = |\{(u_i, tr_j)|u_i \in U \text{ and } tr_j \in Tr\}|,$$

  where $|Q|$ denotes the number of elements contained in set Q. To eliminate the bias, we normalize the weight as

  $$w(e_{ij}^{(3)})' = \frac{w(e_{ij}^{(3)})}{\sqrt{\sum_{k=1}^{|Tr|} w(e_{ik}^{(3)})}\sqrt{\sum_{l=1}^{|U|} w(e_{lj}^{(3)})}}. \quad (9)$$

  Moreover, in order to treat different types of relations (except similarity relations between tracks) equally, the weight is further normalized as follows:

  $$w(e_{ij}^{(3)})^* = \frac{w(e_{ij}^{(3)})'}{ave(w(e_{i\cdot}^{(3)})')}, \quad (10)$$

  where $ave(w(e_{i\cdot}^{(3)})')$ is the average of normalized weights for user u_i.

- $E^{(4)}/E^{(5)}/E^{(6)}$: We build hyperedges for tagging relations on three types of resources as illustrated in Figure 2(b). Each hyperedge contains three vertices (corresponding to a user, a tag and a resource) and the weight is set to be 1.

Table 3: The incidence matrix H of the unified hypergraph and the sub-matrices.

	$E^{(1)}$	$E^{(2)}$	$E^{(3)}$	$E^{(4)}$	$E^{(5)}$	$E^{(6)}$	$E^{(7)}$	$E^{(8)}$	$E^{(9)}$
U	$UE^{(1)}$	$UE^{(2)}$	$UE^{(3)}$	$UE^{(4)}$	$UE^{(5)}$	$UE^{(6)}$	0	0	0
G	0	$GE^{(2)}$	0	0	0	0	0	0	0
Ta	0	0	0	$TaE^{(4)}$	$TaE^{(5)}$	$TaE^{(6)}$	0	0	0
Tr	0	0	$TrE^{(3)}$	$TrE^{(4)}$	0	0	$TrE^{(7)}$	0	$TrE^{(9)}$
Al	0	0	0	0	$AlE^{(5)}$	0	$AlE^{(7)}$	$AlE^{(8)}$	0
Ar	0	0	0	0	0	$ArE^{(6)}$	0	$ArE^{(8)}$	0

- $E^{(7)}/E^{(8)}$: We build a hyperedge for each album which contains all the tracks in this album and the album itself. Similarly, the hyperedge for an artist contains all the albums belonging to the artist and the artist oneself. The weights of the hyperedges corresponding to albums and artists are set to be 1.

- $E^{(9)}$: We build a k nearest neighbor (knn) graph based on acoustic-based music similarities and build hyperedges for our unified hypergraph corresponding to the edges of the knn graph. The weight $w(e_{ij}^{(9)})$ is the similarity of tracks tr_i and tr_j computed in Section 4.2. To eliminate the bias, we normalize the weight as

$$w(e_{ij}^{(9)})' = \frac{w(e_{ij}^{(9)})}{max(w(e^{(9)}))}. \quad (11)$$

  where $max(w(e^{(9)}))$ is the maximum of all music similarities. We introduce a parameter c to control the relative importance between acoustic content of music tracks and other social media information. Finally, the weight is

$$w(e_{ij}^{(9)})^* = c * w(e_{ij}^{(9)})'. \quad (12)$$

Finally, we get the vertex-hyperedge incidence matrix $\mathbf{H}$, as shown in Table 3, and the weight matrix $\mathbf{W}$.

4.4 Methodology

Our music recommendation algorithm MRH has two phases, offline training and online recommendation. In the offline training phase, we first construct the unified hypergraph as described above and get the vertex-hyperedge incidence matrix $\mathbf{H}$ and the weight matrix $\mathbf{W}$. Then the vertex degree matrix $\mathbf{D}_v$ and the hyperedge degree matrix $\mathbf{D}_e$ are computed based on $\mathbf{H}$ and $\mathbf{W}$. Finally, we calculate $(\mathbf{I} - \alpha\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2})^{-1}$, denoted as $(\mathbf{I} - \alpha\mathbf{A})^{-1}$, with α properly set. In the online recommendation phase, we need to build the query vector $\mathbf{y}$ first. Then the ranking results $\mathbf{f}^*$ can be computed.

Our approach can also be applied to other applications by choosing different vertices as queries and considering the ranking results of different vertex types. For example, if we choose a user as the query, the ranking results of music tracks can be used for music track recommendation (i.e., the primary focus of this paper), the ranking results of the users can be used for friend recommendation, and the ranking results of groups can be used for interest group recommendation. For the tag recommendation problem [34, 15], we should set the target user and the target resource as queries and consider the ranking results of tags.

There are three methods to set the query vector $\mathbf{y}$ for music track recommendation: (1) Set the entry of $\mathbf{y}$ corresponding to the target user u to be 1 and all others to be 0. (2) Set the entries of $\mathbf{y}$ corresponding to the target user u, as well as all the other objects connected to u by some hyperedge, to be 1. (3) Set the entry of $\mathbf{y}$ corresponding to the target user u to be 1. Also, if u is connected to an object v, then set the entry of $\mathbf{y}$ corresponding to v to be $A_{u,v}$. Note that, $A_{u,v}$ is a measure of the relatedness between u and v. The first method fails to consider the closely related objects which may also reflect the user's interest. The second method may not be a good choice, since intuitively different objects reflect the user's interest with different degrees. Therefore, in our experiments we adopt the third method. After setting the query vector, the ranking results $\mathbf{f}^*$ can be computed. For the music track recommendation problem, we only consider the ranking results of music tracks as mentioned above. Finally, we can recommend to the user the top ranked tracks which he/she has not listened to before.

5. EXPERIMENTS

In this section, we investigate the use of our proposed approach for music track recommendation.

5.1 Compared Algorithms

We compare our MRH algorithm with five recommendation algorithms. The first one is an user-based Collaborative Filtering (CF) method [29, 19] which only uses listening relations. We choose user-based CF algorithm because, unlike traditional data sets for CF, our data set has much more music tracks than users. Given a target user u_i, let r_{u_i,tr_p} be a predicted ranking score of user u_i for music track tr_p, which is given by [19]

$$r_{u_i,tr_p} = \overline{w}(e_{i.}^{(3)})^* + \frac{\sum_{j=1}^{k}(w(e_{jp}^{(3)})^* - \overline{w}(e_{j.}^{(3)})^*)s_{u_i,u_j}}{\sum_{j=1}^{k} s_{u_i,u_j}}. \quad (13)$$

$$\overline{w}(e_{i.}^{(3)})^* = \frac{\sum_{p=1}^{|Tr|} w(e_{ip}^{(3)})^*}{|\{tr_p | tr_p \in Tr \text{ and } w(e_{ip}^{(3)})^* \neq 0\}|} \quad (14)$$

s_{u_i,u_j} is the similarity weight between users u_i and u_j. k is the number of nearest neighbors of user u_i. We employ the cosine-based approach [6, 32] to compute the similarities between users:

$$s_{u_i,u_j} = \frac{\sum_{p=1}^{|Tr|} w(e_{ip}^{(3)})^* w(e_{jp}^{(3)})^*}{\sqrt{\sum_{p=1}^{|Tr|}\left(w(e_{ip}^{(3)})^*\right)^2}\sqrt{\sum_{p=1}^{|Tr|}\left(w(e_{jp}^{(3)})^*\right)^2}}. \quad (15)$$

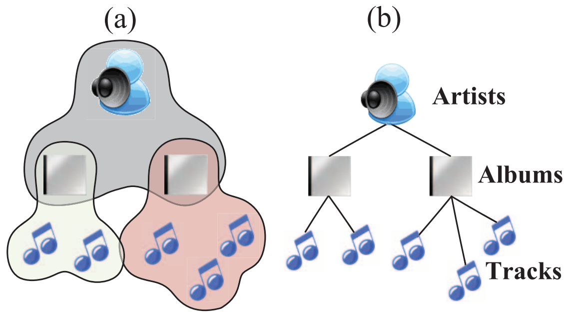


Figure 3: Inclusion relations represented in two models: (a) our unified hypergraph model, and (b) ordinary graph model.

Based on the obtained similarities, we use the significance weighting method proposed in [17] to improve the recommendation performance. Specifically, if the number of co-listened music tracks between two users, denoted by n, is less than a threshold number N, then we multiply their similarity by n/N. In our experiment, we empirically set the value of N to be 20, and the number of nearest neighbors k to be 5, to achieve the best performance.

The second compared algorithm is a acoustic-based music recommendation method [38] which uses listening relations and music similarity relations. This method is denoted by *AB*.

The third compared algorithm uses all the information in our downloaded data set. Unlike MRH, we use the ordinary graph to model social media information. Specifically, we model the tagging relations by graph structure as shown in Fig. 2(a), and model the membership and inclusion relations by tree structure as shown in Fig. 3. The graph ranking algorithm described in [45] is applied to compute the optimal ranking scores. We call this algorithm *Recommendation on Unified Graph* (*RUG*).

The fourth compared algorithm is our MRH method but only using listening relations and music similarity relations (i.e., R_3 and R_9). This method is denoted by *MRH-hybrid*.

The fifth compared algorithm is our MRH method but not using music similarity relations. It uses all the other eight types of relations. This method is denoted by *MRH-social*.

5.2 Evaluation

To evaluate the performance of our MRH algorithm and the other compared algorithms, for each user, we randomly select 20% listening relations as test data for evaluation purpose. If the user has access to a certain music track tr in the test set, we require that he/she has no access to tr in the training set. To achieve this, we remove all the corresponding tagging relations, leaving us with the final training set.

For evaluation metrics, we use Precision, Recall, F1, Mean Average Precision (MAP) and Normalized Discount Cumulative Gain (NDCG) to measure the performance of different recommendation algorithms. Precision is defined as the number of correctly recommended items divided by the total number of recommended items. Recall is defined as the number of correctly recommended items divided by the total number of items which should be recommended (i.e., those actually listened by the target user). F1 is the harmonic mean of Precision and Recall. Average Precision (AP) is the average of precisions computed at the point of each correctly recommended item in the recommendation list:

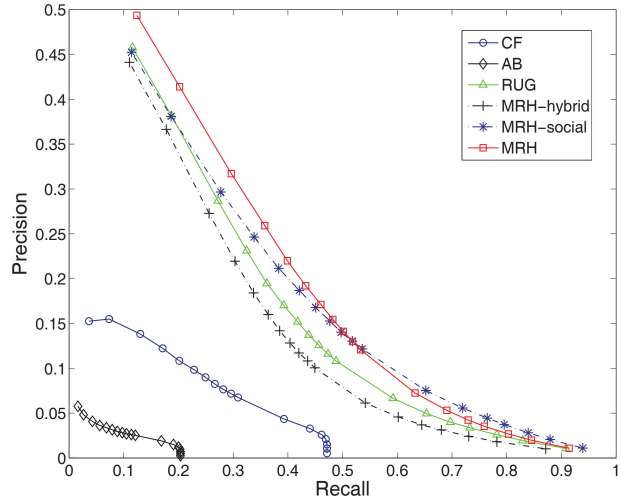


Figure 4: Recall-Precision curves for all the six algorithms.

$$\mathrm{AP} = \frac{\sum_i^N \mathrm{Precision@}i * \mathrm{corr}_i}{\text{Number of correctly recommended items}}, \tag{16}$$

where Precision@i is the precision at ranking position i, N is the number of recommended items, and $\mathrm{corr}_i = 1$ if the item at position i is correctly recommended, otherwise $\mathrm{corr}_i = 0$. MAP is the mean of average precision scores over all users. NDCG at position n is defined as:

$$\mathrm{NDCG@}n = \frac{1}{\mathrm{IDCG}} \times \sum_{i=1}^{n} \frac{2^{r_i} - 1}{log_2(i+1)}, \tag{17}$$

where r_i is the relevance rating of item at rank i. In our case, r_i is 1 if the user has listened to this recommended music and 0 otherwise. IDCG is chosen so that the perfect ranking has a NDCG value of 1.

5.3 Performance Comparison

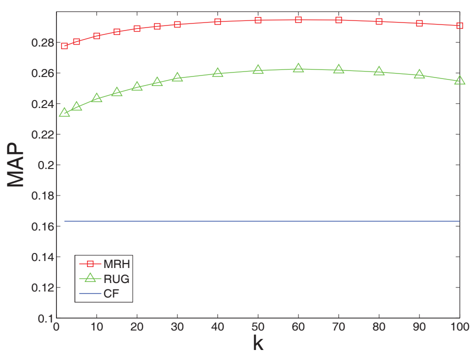
We use all evaluation metrics mentioned in Section 5.2 to measure the performance of each recommendation algorithm. Fig. 4 shows the recall-precision curves for all six algorithms. We report the performance of all six algorithms in terms of MAP, F1 and NDCG in Table 4 (MAP and F1) and Table 5 (NDCG). It is evident that our proposed algorithm significantly outperforms other recommendation algorithms in most cases, especially at lower ranks. Note that, our proposed MRH algorithm models the high-order relations by hyperedges, whereas RUG uses the ordinary graph to approximate these high-order relations. The superiority of MRH over RUG indicates that the hypergraph is indeed a better choice for modeling complex relations in social media communities. Acoustic-based (AB) method works the worst. This is because acoustic-based method incurs the semantic gap and similarities based on acoustic content are not always consistent with human knowledge [10]. CF algorithm does not work well too. This is probably because the user-track matrix in our data set is highly sparse, with only

Table 4: **Comparison of recommendation algorithms in terms of MAP and F1. (Bold typeset indicates the best performance. * indicates statistical significance at $p < 0.001$ compared to the second best.)**

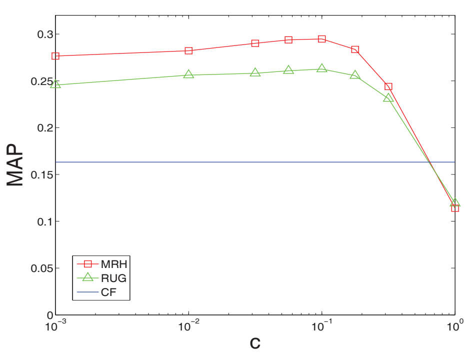
	MAP	F1@5	F1@10	F1@20	F1@30	F1@50	F1@70	F1@100	F1@200
CF	0.1632	0.0557	0.0929	0.1243	0.1329	0.1294	0.1197	0.1064	0.0765
AB	0.0762	0.0226	0.0303	0.0377	0.0403	0.0421	0.0415	0.0401	0.0334
RUG	0.2626	0.1729	0.2323	0.2587	0.2516	0.2237	0.1988	0.1701	0.1169
MRH-hybrid	0.2470	0.1653	0.2224	0.2451	0.2377	0.2099	0.1855	0.1581	0.1076
MRH-social	0.2755	0.1705	0.2311	0.2654	0.2660	0.2440	0.2202	**0.1906**	**0.1318***
MRH	**0.2948***	**0.1855***	**0.2510***	**0.2839***	**0.2799***	**0.2509***	**0.2227**	0.1892	0.1270

Table 5: **Comparison of recommendation algorithms in terms of NDCG. (Bold typeset indicates the best performance. * indicates statistical significance at $p < 0.001$ compared to the second best.)**

	NDCG@5	NDCG@10	NDCG@30	NDCG@50	NDCG@70	NDCG@100	NDCG@200
CF	0.1522	0.1713	0.2519	0.2987	0.3278	0.3579	0.4120
AB	0.0733	0.0820	0.1241	0.1532	0.1749	0.2027	0.2556
RUG	0.4849	0.4318	0.3826	0.4109	0.4345	0.4587	0.5037
MRH-hybrid	0.4587	0.4091	0.3640	0.3911	0.4124	0.4346	0.4753
MRH-social	0.4759	0.4268	0.3866	0.4197	0.4480	0.4763	0.5264
MRH	**0.5192***	**0.4650***	**0.4174***	**0.4484***	**0.4740***	**0.4987***	**0.5419***



(a) k vs. MAP



(b) c vs. MAP

Figure 5: The parameter settings of k and c for music similarity relations. Firstly, we fix c at 0.1 empirically and let k vary. (a) shows the performance measured by MAP. Then we fix k at 60 and let c vary. (b) shows the performance measured by MAP.

about 0.6% non-zero entries. MRH-hybrid only uses similarity relations among music tracks and listening relations, but it works much better than AB and CF.

Comparing to MRH-social, MRH uses similarity relations among music tracks additionally. We find that using this acoustic information can improve the recommendation result, especially when recall is small. This is because acoustic information can alleviate some well-known problems associated with data sparseness in collaborative recommender systems, e.g., user bias, non-association and cold-start problems [20].

5.4 Exploring Parameter Settings

There are three parameters in our algorithm, i.e., the number of nearest neighbors k mentioned in Section 4.3, c in Eq. (12) and α in Eq. (8).

To explore the influence of the parameters k and c, we use MAP as the evaluation metric. Fig. 5 shows the results. Firstly, we fix c at 0.1 empirically and let k vary. Fig. 5(a) shows the performance measured as a function of k. The best result is obtained when k is around 60. Then we fix k at 60 and let c vary. Fig. 5(b) shows the performance measured as a function of c. The best result is obtained when $c = 0.1$. As can be seen, our algorithm consistently outperforms the other two compared algorithms in a wide range of parameter variation. In our experiments, we set k to be 60 and c to be 0.1 for MRH, MRH-hybrid and RUG.

α is a common parameter shared by our proposed MRH algorithm and RUG [45]. In our experiments, we set α to be 0.98 for MRH, MRH-hybrid, MRH-social, and RUG empirically.

5.5 Social Information Contribution

To explore the contributions of different types of social media information to the recommendation performance, we investigate the performances of MRH on four different subsets of social media information. The first subset only contains listening relations (i.e., R_3), which is considered as the base relations. The second subset contains listening relations and social relations (i.e., R_1, R_2). The third subset contains lis-

Table 6: Comparison of MRH on different subsets of social information in terms of MAP and F1. (Bold typeset indicates that the performance is better than that of using the listening relations (R_3) alone. * indicates statistical significance at $p < 0.001$ compared to the algorithm by using listening relations alone.)

	MAP	F1@5	F1@10	F1@30	F1@40	F1@60	F1@70	F1@100	F1@200
MRH on R_3	0.2303	0.1430	0.1996	0.2332	0.2143	0.1945	0.1772	0.1695	0.1184
MRH on R_1, R_2, R_3	**0.2308**	**0.1444**	**0.1998**	**0.2337**	**0.2146**	0.1943	0.1772	0.1695	0.1181
MRH on R_3, R_4	0.2303	**0.1432**	**0.1997**	0.2332	0.2143	0.1945	**0.1773**	0.1695	0.1184
MRH on R_3, R_7, R_8	**0.2757***	**0.1748***	**0.2339***	**0.2642***	**0.2413***	**0.2176***	**0.1970***	**0.1878***	**0.1299***

tening relations and tagging relations on tracks (i.e., R_4). The fourth subset contains listening relations and inclusion relations (i.e., R_7, R_8). From Table 6, we can see that inclusion relations significantly improve the recommendation performance. By using inclusion relations among resources, we can recommend music tracks in the same or similar albums, as well as the tracks performed by the same or similar artists. As can be seen, there is slight improvement at low recall region by using social relations. Intuitively, the users' tastes may be inferred from friendship and membership relations. Tagging relations do not improve the performance. That is because people usually bookmark music tracks they have already listened to. Therefore, there is strong correlation between listening relations and tagging relations, and thus the usage of tagging relations is limited.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we address the music recommendation problem in music social communities, and focus on combining various types of social media information and music acoustic signals. We model the recommendation problem as a ranking problem on a unified hypergraph and propose a novel algorithm for music recommendation on hypergraph (MRH). MRH constructs a hypergraph to model the multi-type objects in a music social community as vertices, and the relations among these objects as hyperedges. Similarities among music tracks based on acoustic signals are treated as one kind of relations. In this way, the high-order relations in social information can be naturally captured. In addition, collaborative filtering and acoustic-based music recommendation is combined in a unified framework. Based on the constructed hypergraph, we then use a regularization framework to derive the ranking results for query vertices. We treat a user as the query and recommend the top ranked music tracks to the user. The experiments on a data set collected from the music social community Last.fm have demonstrated that our proposed algorithm significantly outperforms traditional recommendation algorithms and the rich social media information is very useful for music recommendation.

MRH can also be used for recommender systems in other kinds of social media communities, such as movies and pictures. In this work, we treat all types of social relations (except music similarity relations) equally. However, in practical applications, different types of relations may have different importance. For example, in some pure social networks such as Facebook[3] and LinkedIn[4], the tastes of the users can be affected by their friends significantly. In this case, we should assign relatively higher weights to social relations such as friendship and membership relations. On the other hand, for special interest social media communities (e.g., Last.fm and YouTube[5]), the unified hypergraph model should put more emphasis on the users' actions on resources (e.g., rating and tagging) and the relations among resources (e.g., inclusion relations).

Moreover, as mentioned in Section 4.4, our approach is not limited to music track recommendation. We can exploit it in different applications, such as friend recommendation and personalized tag recommendation. These problems are left for our future work.

Acknowledgments

This work was supported by China National Key Technology R&D Program (2008BAH26B00, 2007BAH11B06) and National Natural Science Foundation of China (60875044, 90920303).

7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] S. Agarwal. Ranking on graph data. In *Proc. the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[3] S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *Proc. the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

[4] J.-J. Aucouturier and F. Pachet. Scaling up music playlist generation. In *Proc. IEEE International Conference on Multimedia and Expo*, Lusanne, Switzerland, 2002.

[5] A. Berenzweig, B. Logan, D. P. W.Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

[6] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. the 14th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, 1998.

[7] S. R. Bulò and M. Pelillo. A game-theoretic approach to hypergraph clustering. In *Advances in Neural Information Processing Systems 22*, Vancouver, Canada, 2009.

[8] R. Cai, C. Zhang, L. Zhang, and W.-Y. Ma. Scalable music recommendation by search. In *Proc. the 15th International Conference on Multimedia*, Augsburg, Germany, 2007.

[9] P. Cano, M. Koppenberger, and N. Wack. Content-based music audio recommendation. In *Proc. the 13th ACM International Conference on Multimedia*, Singapore, 2005.

[10] Ò. Celma. Foafing the music: Bridging the semantic gap in music recommendation. In *Proc. the 5th International Semantic Web Conference*, Athens, Georgia, 2006.

[11] Ò. Celma and P. Lamere. If you like the beatles you might like...: a tutorial on music recommendation. In *Proc. the*

[3]http://www.facebook.com

[4]http://www.linkedin.com

[5]http://www.youtube.com

16th ACM International Conference on Multimedia, Vancouver, British Columbia, Canada, 2008.

[12] S. Chen, F. Wang, and C. Zhang. Simultaneous heterogeneous data clustering based on higher order relationships. In *Proc. the 7th IEEE International Conference on Data Mining Workshops*, Omaha, Nebraska, 2007.

[13] J. Diederich and T. Iofciu. Finding communities of practice from user profiles based on folksonomies. In *Proc. the 1st International Workshop on Building Technology Enhanced Learning Solutions for Communities of Practice*, Crete, Greece, 2006.

[14] J. Donaldson. A hybrid social-acoustic recommendation system for popular music. In *Proc. ACM Conference on Recommender Systems*, New York, NY, 2007.

[15] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *Proc. the 32nd ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, MA, 2009.

[16] A. S. Harpale and Y. Yang. Personalized active learning for collaborative filtering. In *Proc. the 31st ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, 2008.

[17] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, 1999.

[18] P. Knees, T. Pohle, M. Schedl, and G. Widmer. Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proc. the 8th ACM International Workshop on Multimedia Information Retrieval*, Santa Barbara, California, 2006.

[19] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *Proc. the 32nd ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, MA, 2009.

[20] Q. Li, S. H. Myaeng, and B. M. Kim. A probabilistic music recommender considering user opinions and audio features. *Information Processing and Management*, 43(2):473–487, 2007.

[21] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proc. the 31st ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, 2008.

[22] B. Logan. Music recommendation from song sets. In *Proc. the 5th International Conference on Music Information Retrieval*, Barcelona, Spain, 2004.

[23] B. Logan and A. Salomon. Music similarity function based on signal analysis. In *Proc. IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, 2001.

[24] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proc. the 32nd ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, MA, 2009.

[25] C. McKay and I. Fujinaga. Combining features extracted from audio, symbolic and cultural sources. In *Proc. the 9th International Conference on Music Information Retrieval*, Philadelphia, PA, 2008.

[26] H.-S. Park, J.-O. Yoo, and S.-B. Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. *Fuzzy Systems and Knowledge Discovery*, 4223:970–979, 2006.

[27] S. Pauws, W. Verhaegh, and M. Vossen. Fast generation of optimal music playlists using local search. In *Proc. the 7th International Conference on Music Information Retrieval*, Victoria, Canada, 2006.

[28] R. J. Ragno, C. J. C. Burges, and C. Herley. Inferring similarity between music objects with application to playlist generation. In *Proc. the 7th ACM SIGMM Workshop on Multimedia Information Retrieval*, Hilton, Singapore, 2005.

[29] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. the ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, 1994.

[30] S. Rho, B. jun Han, and E. Hwang. Svr-based music mood classification and context-based music recommendation. In *Proc. the 17th ACM International Conference on Multimedia*, Beijing, China, 2009.

[31] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[32] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proc. the 10th International Conference on World Wide Web*, Hong Kong, China, 2001.

[33] S. Sen, J. Vig, and J. Riedl. Tagommenders: connecting users to items through tags. In *Proc. the 18th International Conference on World Wide Web*, Madrid, Spain, 2009.

[34] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, and C. L. Giles. Real-time automatic tag recommendation. In *Proc. the 31st ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, 2008.

[35] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *Proc. the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, 2008.

[36] P. Symeonidis, M. Ruxanda, A. Nanopoulos, and Y. Manolopoulos. Ternary semantic analysis of social tags for personalized music recommendation. In *Proc. the 9th International Conference on Music Information Retrieval*, Pennsylvania, USA, 2008.

[37] D. Tao, H. Liu, and X. Tang. K-box: a query-by-singing based music retrieval system. In *Proc. the 12th ACM International Conference on Multimedia*, New York, NY, 2004.

[38] M. Tiemann and S. Pauws. Towards ensemble learning for hybrid music recommendation. In *Proc. ACM Conference on Recommender Systems*, New York, NY, 2007.

[39] K. H. L. TsoSutter, L. B. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proc. the 2008 ACM Symposium on Applied Computing*, Fortaleza, Brazil, 2008.

[40] K. Yoshii and M. Goto. Continuous pLSI and smoothing techniques for hybrid music recommendation. In *Proc. the 10th International Society for Music Information Retrieval Conference*, Kobe, Japan, 2009.

[41] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *Proc. the 7th International Conference on Music Information Retrieval*, Victoria, Canada, 2006.

[42] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proc. the 30th ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, 2007.

[43] Z. Zhang, T. Zhou, and Y. Zhang. Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. *Physica A: Statistical Mechanics and its Applications*, 389(1):179–186, 2009.

[44] D. Zhou, J. Huang, and B. Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems 19*, Vancouver, Canada, 2006.

[45] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2003.