Using Rich Social Media Information for Music Recommendation via Hypergraph Model

SHULONG TAN, JIAJUN BU, CHUN CHEN, BIN XU, CAN WANG, and XIAOFEI HE, Zhejiang University

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 main 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. 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.

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

Additional Key Words and Phrases: Recommender system, music recommendation, hypergraph, social media information

ACM Reference Format:

Tan, S., Bu, J., Chen, C., Xu, B., Wang, C., and He, X. 2011. Using rich social media information for music recommendation via hypergraph model. ACM Trans. Multimedia Comput. Commun. Appl. 7S, 1, Article 22 (October 2011), 22 pages.
DOI = 10.1145/2037676.2037679 http://doi.acm.org/10.1145/2037676.2037679

1. INTRODUCTION

With the recent advances of social media communities (e.g., Last.fm[1] Flickr[2] and YouTube[3]), there is an emerging presence of social media information, for example, user collective actions, implicit social networking structure and relations among media objects. This information not only facilitates users

[1]http://www.last.fm.
[2]http://www.flickr.com.
[3]http://www.youtube.com.

This work was supported by the China National Key Technology R&D Program (2008BAH26B00, 2007BAH11B06) and the National Natural Science Foundation of China (60875044, 90920303).
Authors' addresses: S. Tan, J. Bu, C. Chen, B. Xu, and C. Wang, Zhejiang Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou, China, 310027; email: {shulongtan, bjj, chenc, xbzju, wcan}@zju.edu.cn; X. He, State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, China, 310027; email: xiaofeihe@cad.zju.edu.cn.

DOI 10.1145/2037676.2037679 http://doi.acm.org/10.1145/2037676.2037679

in communication and organizing online resources, but is also valuable in many research tasks, such as social networks analysis and information retrieval. In particular, these kinds of social media information are important sources of information for recommender systems [Konstas et al. 2009; Lin et al. 2009].

Among these kinds of social media information, explicit feedback (e.g., in terms of ratings or use frequencies) from users is the most important for recommendation. Traditional recommender systems use techniques such as Collaborative Filtering (CF) [Resnick et al. 1994; Harpale and Yang 2008; Liu and Yang 2008], which only apply user-item explicit feedback matrix. As a kind of user collective action, explicit feedback presents collective information among users. Based on explicit feedback, recommendation can be done among similar users or items. Another type of collective action is social tagging, for example, Last.fm allows users to tag artists, albums or music tracks and Del.icio.us[4] allows users to tag webpages. Social tags carry useful information not only about the tagged items, but also about the preference of users who make the tags. Several algorithms have been proposed to exploit social tagging information for recommender systems [Diederich and Iofciu 2006; Tso-Sutterr et al. 2008; Guan et al. 2010].

In social media communities, users can make friends with other users or join some interest groups. These actions build a implicit social networking structure. This social networking structure is useful for predicting users' preferences, because the users' interests may be affected by their friends or neighbors in interest groups. There have been some papers already in utilizing friendship relations for recommendation [Ma et al. 2009; Konstas et al. 2009]. But no previous works exploit membership information about interest groups in recommendation.

Moreover, relations among media objects (e.g., inclusion relations among music tracks, albums and artists in Last.fm, inclusion relations between collections and photos in Flickr) not only can be used to organize resource items, but are also valuable in recommendation. We found that this information greatly improves the recommendation performance (see Section 6.5). But to the best of our knowledge, no emphasis has been placed on recommendation based on this kind of information.

Figure 1 shows an example of social media information in online music social community Last.fm. This information includes friendships, memberships, listening histories, tagging relations, inclusion relations among resources and similarities between music tracks which can be computed based on music content.

1.1 Motivation

We focus on music recommendation here. For the task of music recommendation, the most common approach is to directly analyze the audio signal. These methods are called acoustic-based music recommendation [Logan 2004; Cano et al. 2005; Cai et al. 2007; Rho et al. 2009]. Due to the semantic gap between low level acoustic features and high level music concepts [Celma 2006], the results of acoustic-based music recommendation are not satisfactory. It is necessary to consider more information in music recommendation [Celma and Lamere 2008]. Some researchers try to utilize the user rating information by applying collaborative filtering methods [Yoshii et al. 2006; Li et al. 2007; Tiemann and Pauws 2007; Yoshii and Goto 2009]. There are also works which exploit the information in the meta data (e.g., genre) associated with music tracks [Aucouturier and Pachet 2002; Ragno et al. 2005; Pauws et al. 2006]. However, all these approaches only utilize limited kinds of information, without considering rich social media information.

The various social media information mentioned above is very useful for music recommendation. However, there are several challenges to exploit all this information. First, it is difficult to take in

[4] http://delicious.com.

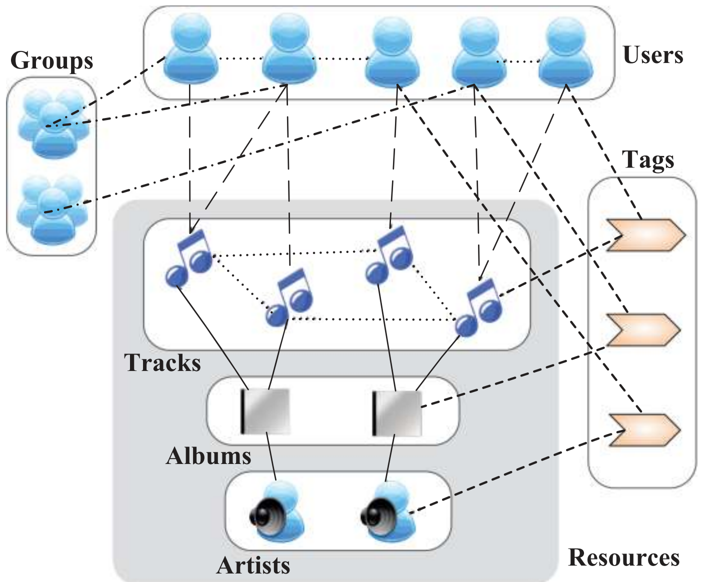


Fig. 1. Various types of objects and relations in the music social community Last.fm. The relations include friendship relations, membership relations, listening histories, tagging relations, inclusion relations among resources (e.g., tracks and albums) and similarity relations between music tracks.

account all types of media objects and relations in a unified framework simultaneously. It is difficult for traditional methods such as k-nearest neighbor (kNN) Collaborative Filtering and matrix or tensor factorization (MF/TF) to expand and utilize more kinds of social information. Second, in social media communities, some relations are beyond pairwise and are high-order relations. For example, multiple items belong to the same sets, or a user use a tag to bookmark a resource. Traditional methods that deal with pairwise relations can not properly model these high-order relations. Third, because most social media communities do not allow for free access to all user profiles, such as friend lists or interest group lists, there is not a concrete dataset yet that includes all social media information mentioned above.

Recently, there has been considerable interest in making use of social media information to enhance the recommendation performance [Tso-Sutterr et al. 2008; Symeonidis et al. 2008; Konstas et al. 2009; Ma et al. 2009; Sen et al. 2009; Zhang et al. 2009]. For example, some previous works employed ordinary graphs to model tagging data for recommendation problems [Konstas et al. 2009; Zhang et al. 2009]. Figure 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. Figure 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 precisely. For example, from Figure 2(a), it is unclear whether u_2 bookmarks r_1, r_2, or both.

1.2 Contributions

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

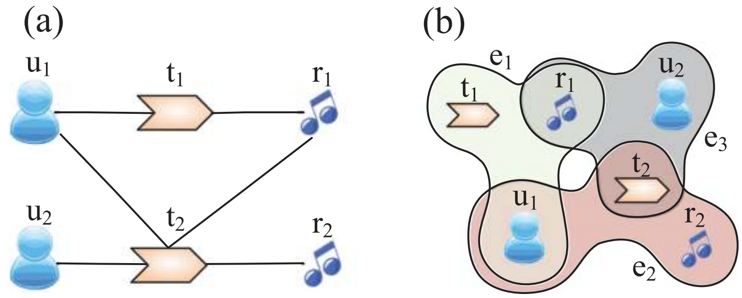


Fig. 2. Tagging relations represented in two models: (a) ordinary graph model, and (b) our unified hypergraph model. This hypergraph contains six vertices and three hyperedges, that is, (u_1, t_1, r_1), (u_1, t_2, r_2), and (u_2, t_2, r_1).

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 nonempty subsets of the vertex set [Agarwal et al. 2006]. Each vertex of the hypergraph corresponds to an object of any type. The hyperedges are used to model high-order relations, as shown in Figure 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 each user desires.

The following points highlight the contributions of this work.

(1) Multisource information fusion. We integrate multisource 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 hypergraphs instead of traditional graphs. 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.

This work is an extended and improved follow-up to our earlier paper [Bu et al. 2010]. In comparison, we add a substantially theoretical analysis about the background of ranking on graph data. The computational complexity of our algorithm is discussed and some speed up strategies are introduced additionally. We also extend the experiments here, such as exploring the parameter α setting and representing recommendation examples.

2. RELATED WORK

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 [Yoshii et al. 2006; Li et al. 2007; Tiemann and Pauws 2007; Donaldson 2007; Yoshii and Goto 2009]. Yoshii et al. [2006] and Yoshii and Goto [2009] integrate both rating and music content information by using probabilistic models. Unobservable user preferences are directly represented by introducing latent variables. Li et al. [2007] propose an item-based probabilistic model utilizing audio features to capture accurate similarities among items (i.e., music). Tiemann et al. [2007] 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 [2007] 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 of these 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. [2008] 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 [Adomavicius and Tuzhilin 2005], respectively. Diederich et al. [2006] 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. [2009] propose a recommendation algorithm by integrating diffusion on user-tag-item tripartite graphs. Ma et al. [2009] propose a probabilistic factor analysis framework which naturally fuses the users' preferences and their trusted friends' favors together. To utilize both friendship and tagging relations, Konstas et al. [2009] create a collaborative recommender system that constructs a social graph over users, tags and resources. Sen et al. [2009] 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 clickthrough and search information, in the step of tag preferences generation. Knees et al. [2006] 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 these 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. 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 [Zhou et al. 2003b, 2006; Agarwal 2006; Agarwal et al. 2006; Chen et al. 2007; Sun et al. 2008; Bulò and Pelillo 2009].

Zhou et al. [2003b] propose a manifold ranking algorithm which ranks data objects with respect to the intrinsic geometrical structure in the data. 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 [2006] 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. [2009] propose a graph-based ranking algorithm for interrelated multi-type objects.

Recently, there has been a lot of interest in learning with hypergraph [Agarwal et al. 2006; Zhou et al. 2006; Chen et al. 2007; Sun et al. 2008; Bulò and Pelillo 2009]. Bulò and Pelillo [2009] introduce a hypergraph clustering algorithm to extract maximally coherent groups from a set of objects using high-order (rather than pairwise) similarities. Zhou et al. [2006] develop a general framework which is applicable to classification, clustering and embedding on hypergraph data. 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 unified hypergraph, we consider music recommendation as a ranking problem on unified hypergraph.

3. BACKGROUND OF RANKING ON GRAPH DATA

Let $G(V, E, w)$ denote an ordinary graph where $V = \{v_1, \ldots, v_{|V|}\}$ is the set of vertices, E is the set of the pairwise edges, and w is a weight function defined as $w : E \to \mathbb{R}$, high weight indicates that two vertices are near. The *weighted adjacency matrix* of the ordinary graph is the matrix $\mathbf{W} = (w_{ij})_{i,j=1,\ldots,|V|}$. The degree of a vertex $v_i \in V$ is defined as

$$d_i = \sum_{j=1}^{|V|} w_{ij}. \tag{1}$$

The *vertex degree matrix* $\mathbf{D}$ of the ordinary graph is defined as the diagonal matrix with the degrees $d_1, \ldots, d_{|V|}$ on the diagonal.

The problem of ranking on graph data is addressed in a "query and ranking" manner as follows. Given some query vertices from V, rank the other vertices on the graph according to their relevance to the queries. Let $\mathbf{y} = [y_1, y_2, \ldots, y_{|V|}]^T$ denote the query vector and y_i denotes the initial score of the ith vertex. Similarly, let $\mathbf{f} = [f_1, f_2, \ldots, f_{|V|}]^T$ denote the ranking results.

3.1 Regularization Framework for Ranking on Graph

The cost function of the regularization framework for ranking on graph data is as follows [Zhou et al. 2003b; Guan et al. 2009]

$$Q(\mathbf{f}) = \frac{1}{2} \sum_{i,j=1}^{|V|} W_{ij} \left\| \frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}} \right\|^2 + \mu \sum_{i=1}^{|V|} \| f_i - y_i \|^2, \tag{2}$$

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

$$\mathbf{f}^* = \arg\min_{\mathbf{f}} Q(\mathbf{f}). \tag{3}$$

The first term of the right-hand side in Eq. (2) is the smoothness constraint, which means that vertices should have similar ranking scores if they are near. The second term measures the difference between the obtained ranking scores and the pre-given labels which needs to be minimized. The parameter μ controls the relative importance of these two terms.

We define a matrix

$$\mathbf{S} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}. \tag{4}$$

Then, we can rewrite the cost function (2) in the matrix-vector form:

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

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

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

Following some simple algebraic steps, we have

$$\mathbf{f}^* = \frac{\mu}{1+\mu}\Big(\mathbf{I} - \frac{1}{1+\mu}\mathbf{S}\Big)^{-1}\mathbf{y}. \tag{5}$$

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

$$\mathbf{f}^* = (\mathbf{I} - \alpha\mathbf{S})^{-1}\mathbf{y}. \tag{6}$$

3.2 Random Walks with Restarts Model

In the view of random walks with restarts theory [Lovász 1993; Konstas et al. 2009], we can model ranking on graph as follows. Starting from a particular vertex in the starting vertex set V^*, the model is performed by following a edge to another vertex or restarting from one vertex in V^* at each step. In every step there is a probability α to walk to neighbors of the current vertex and a probability $1-\alpha$ to restart from the starting vertex set V^*. If the current vertex is v_i and the model walks to the neighbors, there is a probability $p_{ij} = w_{ij}/D_{ii}$ to the vertex v_j. Let $\mathbf{p}^{(t)}$ be a column vector where $p_i^{(t)}$ denotes the probability that the random walk at step t is at node v_i. $\mathbf{q}$ is a column vector of zeros with 1s corresponding to vertices in the starting vertex set (i.e., $q_i = 1$, if $v_i \in V^*$). The transition probability matrix of the graph is $\mathbf{T} = \mathbf{D}^{-1}\mathbf{W}$. The stationary probabilities for each vertex can be obtained by recursively applying Eq. (7) until convergence,

$$\mathbf{p}^{(t+1)} = \alpha\mathbf{T}\mathbf{p}^{(t)} + (1-\alpha)\mathbf{q} \tag{7}$$

The stationary probabilities present the long term visit rate of each vertex given a bias towards the starting vertex set V^*. Therefore, each stationary probability corresponding to a vertex v_i can be considered as a measure of relatedness between v_i and the starting vertex set.

To find $\mathbf{p}^c$, where c is the state after convergence, we set $\mathbf{p}^{(t+1)} = \mathbf{p}^{(t)} = \mathbf{p}^c$. Then, we can get this equation:

$$\mathbf{p}^c = (1-\alpha)(\mathbf{I} - \alpha\mathbf{T})^{-1}\mathbf{q}. \tag{8}$$

Since $1-\alpha$ does not change the ranking results, we can rewrite $\mathbf{p}^c$ as follows:

$$\mathbf{p}^c = (\mathbf{I} - \alpha\mathbf{T})^{-1}\mathbf{q}. \tag{9}$$

We find that this expression is similar to the ranking result deduced by the regularization framework.

4. 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.

4.1 Notation and Problem Definition

Let $G(V, E_h, w)$ denote a hypergraph where E^h is the set of hyperedges. Different from ordinary graphs, each hyperedge $e \in E_h$ 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 $d(v) = \sum_{e\in E_h|v\in e} w(e)$. 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 $v_1, e_1, v_2, e_2, \ldots, e_{k-1}, v_k$, 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 [Zhou et al. 2006]. We define a vertex-hyperedge incidence matrix $\mathbf{H} \in \mathbb{R}^{|V|\times|E_h|}$ whose entry $h(v, e)$ is 1 if $v \in e$ and 0 otherwise. Then, we have:

$$d(v) = \sum_{e \in E_h} w(e)h(v, e), \tag{10}$$

$$\delta(e) = \sum_{v \in V} h(v, e). \tag{11}$$

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

In the following, we define *unified hypergraph* that will be used to model the high-order relations among different types of objects. A unified hypergraph is a hypergraph that has multitype vertices and hyperedges. Suppose a unified hypergraph has R types of vertices and S types of hyperedges. The vertex set of the rth type is denoted by $V^{(r)}$ and the hyperedge set of the sth type is denoted by $E_h^{(s)}$. We define $V = \bigcup_{r=1}^{R} V^{(r)}$ and $E_h = \bigcup_{s=1}^{S} E_h^{(s)}$. 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 hypergraphs is similar to ranking on ordinary graphs. Given some query vertices from V, rank the other vertices on the unified hypergraph according to their relevance to the queries. Let $\mathbf{y} = [y_1, y_2, \ldots, y_{|V|}]^T$ denote the query vector and $y_i, i = 1, \ldots, |V|$, denote the initial score of the ith vertex. We will discuss how to set the query vector in detail in Section 5.4. Similarly, let $\mathbf{f} = [f_1, f_2, \ldots, f_{|V|}]^T$ be the vector of ranking scores.

4.2 Regularization Framework for Ranking on Unified Hypergraph

There are many existing algorithms for learning on hypergraph [Agarwal et al. 2006; Zhou et al. 2006; Chen et al. 2007; Sun et al. 2008; Bulò and Pelillo 2009]. However, most of them focus on classification, clustering, and Euclidean embedding. In this section, we discuss how to perform ranking on unified hypergraph by using an idea similar to [Zhou et al. 2006].

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

$$Q(\mathbf{f}) = \frac{1}{2} \sum_{i,j=1}^{|V|} \sum_{e \in E_h} \frac{w(e)h(v_i, e)h(v_j, e)}{\delta(e)} \left\| \frac{f_i}{\sqrt{d(v_i)}} - \frac{f_j}{\sqrt{d(v_j)}} \right\|^2 + \mu \sum_{i=1}^{|V|} \| f_i - y_i \|^2, \tag{12}$$

where $\mu > 0$ is the regularization parameter. This function is similar to Eq. (2). The optimal ranking result is achieved when $Q(\mathbf{f})$ is minimized:

$$\mathbf{f}^* = \arg\min_{\mathbf{f}} Q(\mathbf{f}). \tag{13}$$

The first term of the right-hand side in Eq. (12) is a smoothness constraint too. 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. 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 second term and the parameter μ play the same roles as in Eq. (2).

The first term of the right-hand side in the cost function (12) can be rewritten as follows:

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{|V|}\sum_{e\in E_h}\frac{w(e)h(v_i,e)h(v_j,e)}{\delta(e)}\left\|\frac{f_i}{\sqrt{d(v_i)}}-\frac{f_j}{\sqrt{d(v_j)}}\right\|^2\\
&=\sum_{i,j=1}^{|V|}\sum_{e\in E_h}\frac{w(e)h(v_i,e)h(v_j,e)}{\delta(e)}\left(\frac{f_i^2}{d(v_i)}-\frac{f_i f_j}{\sqrt{d(v_i)d(v_j)}}\right)\\
&=\sum_{i=1}^{|V|}f_i^2\sum_{e\in E_h}\frac{w(e)h(v_i,e)}{d(v_i)}\sum_{j=1}^{|V|}\frac{h(v_j,e)}{\delta(e)}\\
&\quad-\sum_{i,j=1}^{|V|}\sum_{e\in E_h}\frac{f_i w(e)h(v_i,e)h(v_j,e)f_j}{\sqrt{d(v_i)d(v_j)}\delta(e)}\\
&=\sum_{i=1}^{|V|}f_i^2-\sum_{i,j=1}^{|V|}\sum_{e\in E_h}\frac{f_i w(e)h(v_i,e)h(v_j,e)f_j}{\sqrt{d(v_i)d(v_j)}\delta(e)}\\
&=\mathbf{f}^T\mathbf{f}-\mathbf{f}^T\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}_h\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{f}.
\end{aligned}\tag{14}$$

We define a matrix

$$\mathbf{A}=\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}_h\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}.\tag{15}$$

Then we can rewrite the cost function (12) in the matrix-vector form:

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

The following formal deductions are the same as the ways in Section 3.1 and we can get the ranking result as follows.

$$\mathbf{f}^*=(\mathbf{I}-\alpha\mathbf{A})^{-1}\mathbf{y}.\tag{16}$$

There is a variant of the results: $\mathbf{f}^*=(\mathbf{I}-\alpha\mathbf{A})^{-1}\mathbf{y}$ and $\mathbf{A}=\mathbf{D}_v^{-1}\mathbf{H}\mathbf{W}_h\mathbf{D}_e^{-1}\mathbf{H}^T$, which corresponds to Random Walks with Restarts model. We will compare this variant with our algorithm in experiments.

5. MUSIC RECOMMENDATION VIA HYPERGRAPH

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

5.1 Data Collection

To evaluate our algorithm, we have collected data from Last.fm in December 2009. First, 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 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, that is, track set,

Table I. 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 II. 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	–

album set and artist set, denoted by Tr, Al, and Ar, respectively. We collected the tagging relations which are essentially triples, that is, (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 I and Table II, respectively. Similarities between music tracks are computed based on music content.

5.2 Acoustic-Based Music Similarity

Acoustic measures of music similarity have been extensively studied in recent years [Logan and Salomon 2001; Tao et al. 2004; Berenzweig et al. 2004; McKay and Fujinaga 2008]. 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, we derive features from Mel-frequency cepstral coefficients (MFCCs) Berenzweig et al. [2004]. 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 Logan and Salomon [2001], 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) [Rubner et al. 2000].

5.3 Unified Hypergraph Construction

We take into account six types of objects and nine types of relations in the data set previously mentioned. 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, that is, 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 II. We denote the hyperedge sets as $E_h^{(s)}$ corresponding to R_s, $s = 1, \ldots, 9$. The construction of the nine types of hyperedges is listed as follows.

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

—$E_h^{(2)}$. For each group, we build a hyperedge that 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.

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

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

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

$$w\big(e_{ij}^{(3)}\big)' = \frac{w\big(e_{ij}^{(3)}\big)}{\sqrt{\sum_{k=1}^{|Tr|} w(e_{ik}^{(3)})}\sqrt{\sum_{l=1}^{|U|} w\big(e_{lj}^{(3)}\big)}}. \tag{17}$$

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

$$w\big(e_{ij}^{(3)}\big)^* = \frac{w\big(e_{ij}^{(3)}\big)'}{ave\big(w\big(e_{i.}^{(3)}\big)'\big)}, \tag{18}$$

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

—$E_h^{(4)}/E_h^{(5)}/E_h^{(6)}$: For tagging relations, there are two choices to build hyperedges: (1) Each hyperedge contains three vertices (corresponding to a user, a tag and a resource). (2) Each hyperedge contains vertices corresponding to a user, a resource and all tags used by the user for the resource. Customarily, the tagging relations are treated as triples, so we choose the first approach in this article. The weight is set to be 1.

—$E_h^{(7)}/E_h^{(8)}$: 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.

—$E_h^{(9)}$: 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

$$w\big(e_{ij}^{(9)}\big)' = \frac{w\big(e_{ij}^{(9)}\big)}{max(w(e^{(9)}))}. \tag{19}$$

Table III. The Incidence Matrix **H** of the Unified Hypergraph and the Submatrices

	$E_h^{(1)}$	$E_h^{(2)}$	$E_h^{(3)}$	$E_h^{(4)}$	$E_h^{(5)}$	$E_h^{(6)}$	$E_h^{(7)}$	$E_h^{(8)}$	$E_h^{(9)}$
U	$UE_h^{(1)}$	$UE_h^{(2)}$	$UE_h^{(3)}$	$UE_h^{(4)}$	$UE_h^{(5)}$	$UE_h^{(6)}$	0	0	0
G	0	$GE_h^{(2)}$	0	0	0	0	0	0	0
Ta	0	0	0	$TaE_h^{(4)}$	$TaE_h^{(5)}$	$TaE_h^{(6)}$	0	0	0
Tr	0	0	$TrE_h^{(3)}$	$TrE_h^{(4)}$	0	0	$TrE_h^{(7)}$	0	$TrE_h^{(9)}$
Al	0	0	0	0	$AlE_h^{(5)}$	0	$AlE_h^{(7)}$	$AlE_h^{(8)}$	0
Ar	0	0	0	0	0	$ArE_h^{(6)}$	0	$ArE_h^{(8)}$	0

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

$$w\left(e_{ij}^{(9)}\right)^* = c * w\left(e_{ij}^{(9)}\right)'. \tag{20}$$

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

5.4 Methodology

Our music recommendation algorithm MRH contains two phases, offline training and online recommendation. In the offline training phase, we first construct the unified hypergraph as previously described and get the vertex-hyperedge incidence matrix $\mathbf{H}$ and the weight matrix $\mathbf{W}_h$. Then, the vertex degree matrix $\mathbf{D}_v$ and the hyperedge degree matrix $\mathbf{D}_e$ are computed based on $\mathbf{H}$ and $\mathbf{W}_h$. Finally, we calculate $(\mathbf{I} - \alpha\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}_h\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2})^{-1}$, 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 work), 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 [Song et al. 2008; Guan et al. 2009], 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 previously. Finally, we can recommend to the user the top-ranked tracks that he/she has not listened to before.

5.5 Computational Complexity Analysis and Speed Up Strategy

In this section, we analyze the computational cost of MRH. Let m denote the number of vertices and n denote the number of hyperedges in the unified hypergraph. Let p be the density of the matrix $\mathbf{H}$, that is, the probability of nonzero entries. To calculate matrix $\mathbf{A}$, it requires $O(p^2nm^2)$ operations and $(1-(1-p^2)^n)m^2$ memory, where $1-(1-p^2)^n$ is the density of matrix $\mathbf{A}$. If p is very small (e.g., p is $7.3*10^{-5}$ in our data), $\mathbf{A}$ and $\mathbf{I}-\alpha\mathbf{A}$ are highly sparse. Computing the inverse of matrix $\mathbf{I}-\alpha\mathbf{A}$ requires $O(m^3)$ operations and m^2 memory. Since $\mathbf{I}-\alpha\mathbf{A}$ is sparse, the computation of matrix inversion will be efficient [Svizhenko et al. 2009].

In real-world social media communities, the size of matrix $(\mathbf{I}-\alpha\mathbf{A})$, despite its sparsity, may be potentially huge. By analyzing the MRH recommendation method, we can find the most time is consumed on computing the inverse of matrix $(\mathbf{I}-\alpha\mathbf{A})$. If the size of this matrix is very large, it is time consuming. Moreover, for real recommender systems, the update of matrix $\mathbf{A}$ is performed periodically. For example, a user accesses a resource recently or new resources are added. So there is a great amount of computation. Fortunately, some approximation approaches can be used to speed up the algorithm. We describe them in the following part.

Similar to the Random Walks with Restarts model represented in the background part, we can formulate an iterative approach for our model as follows:

$$\mathbf{f}^{(t+1)} = \alpha\mathbf{A}\mathbf{f}^{(t)} + (1-\alpha)\mathbf{y}, \tag{21}$$

where t is the iteration step index. Obviously, it requires $O(m^2)$ operations for one iteration step. Since A is highly sparse, it is very fast for one iteration. Generally, we need to repeat the iteration until convergence. For example, the stop condition can be defined as $\|\mathbf{f}^{(t+1)}-\mathbf{f}^{(t)}\| < \varepsilon$, where ε is a very small threshold. However, we find that in the first several steps, $\mathbf{f}$ changes rapidly, while in the rest steps before stop, $\mathbf{f}$ is comparatively very stable. That means, we don't have to find the convergence state. In experiments, we find that using only 20 iterations achieves a good performance. By such a strategy, the computation can be speeded up a lot, besides it saves memory since we need not to store a dense matrix in m^2 size.

Another approach is the approximation of matrix decomposition. If $\mathbf{A}$ can be approximately written as $\mathbf{A}=\mathbf{H}\mathbf{G}\mathbf{H}^T$, where $\mathbf{H}\in\mathbb{R}^{m\times k}$, $\mathbf{G}\in\mathbb{R}^{k\times k}$ and $k \ll m$, then the well-known Woodbury formula can be used to accelerate the inverse computation. That is, $(\mathbf{I}-\alpha\mathbf{H}\mathbf{G}\mathbf{H}^T)^{-1} = (\mathbf{I}-\mathbf{H}(\mathbf{H}^T\mathbf{H}-\frac{1}{\alpha}\mathbf{G}^{-1})^{-1}\mathbf{H}^T)$. With this equation, the inverse cost is reduced to $O(k^3)$. But it requires $O(m^3)$ computations to decompose $\mathbf{A}$, such as the SVD decomposition ($\mathbf{A}=\mathbf{U}\Sigma\mathbf{U}^T$, since $\mathbf{A}$ is symmetric). So we need some fast decomposition strategies. We can arrange $\mathbf{A}$ to be

$$\mathbf{A} = \begin{pmatrix} \mathbf{W} & \mathbf{A}_{21}^T \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \quad \text{and} \quad \mathbf{C} = \begin{pmatrix} \mathbf{W} \\ \mathbf{A}_{21} \end{pmatrix}.$$

Two sampling-based methods can be used to approximate the SVD decomposition of $\mathbf{A}$. The Nyström method [Williams and Williams 2001] uses a $l\times l$ submatrix $\mathbf{W}$ in $\mathbf{A}$, and the approximate singular values and singular vectors of $\mathbf{A}$ are: $\Sigma_i = (\frac{m}{l})\Sigma_w$ and $\mathbf{U}_i = \sqrt{\frac{l}{m}}\mathbf{C}\mathbf{U}_w\Sigma_w^{-1}$. When k singular vectors are used, the run cost of the Nyström method is $O(l^3+mlk)$. Another alternative method is the Column-sampling method [Frieze et al. 2004]. It approximates the spectral decomposition of $\mathbf{A}$ by using the SVD decomposition on $\mathbf{C}$ directly: $\Sigma_{c;i} = \sqrt{\frac{m}{l}}\Sigma_c$ and $\mathbf{U}_{c;i} = \mathbf{U}_c$. Then, the run cost of the Column-sampling method is $O(ml^2)$.

6. EXPERIMENTS

6.1 Compared Algorithms

We compare our MRH algorithm with other four recommendation algorithms. The first one is a user-based Collaborative Filtering (CF) method [Resnick et al. 1994; Konstas et al. 2009] that only uses listening relations. We choose user-based CF algorithm because 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 Konstas et al. [2009]

$$r_{u_i,tr_p} = \overline{w}\big(e_{i.}^{(3)}\big)^* + \frac{\sum_{j=1}^{k}\big(w\big(e_{jp}^{(3)}\big)^* - \overline{w}\big(e_{j.}^{(3)}\big)^*\big)s_{u_i,u_j}}{\sum_{j=1}^{k} s_{u_i,u_j}}, \tag{22}$$

where

$$\overline{w}\big(e_{i.}^{(3)}\big)^* = \frac{\sum_{p=1}^{|Tr|} w\big(e_{ip}^{(3)}\big)^*}{\big|\big\{tr_p | tr_p \in Tr \text{ and } w\big(e_{ip}^{(3)}\big)^* \neq 0\big\}\big|} \tag{23}$$

and 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 [Breese et al. 1998; Sarwar et al. 2001] to compute the similarities between users:

$$s_{u_i,u_j} = \frac{\sum_{p=1}^{|Tr|} w\big(e_{ip}^{(3)}\big)^* w\big(e_{jp}^{(3)}\big)^*}{\sqrt{\sum_{p=1}^{|Tr|}\big(w\big(e_{ip}^{(3)}\big)^*\big)^2}\sqrt{\sum_{p=1}^{|Tr|}\big(w\big(e_{jp}^{(3)}\big)^*\big)^2}}. \tag{24}$$

Based on the obtained similarities, we use the significance weighting method proposed in Herlocker et al. [1999] 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 [Tiemann and Pauws 2007], which uses listening relations and music similarity relations. It 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 Figure 2(a), and model the membership and inclusion relations by tree structure as shown in Figure 3. The graph ranking algorithm described in Zhou et al. [2003b] is applied to compute the optimal ranking scores. We call this algorithm *Recommendation on Unified Graph* (*RUG*).

The fourth compared algorithm is the variant of our proposed MRH method mentioned in Section 4.2, which is named as *MRH-variant*.

We also compare the performances of our proposed method on different subsets of information. The first one 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 second one 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*.

6.2 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

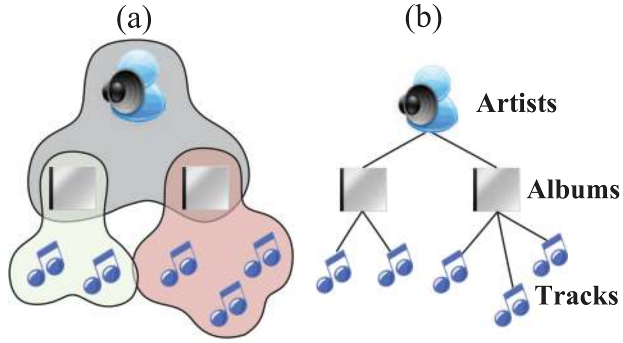


Fig. 3. Inclusion relations represented in two models: (a) our unified hypergraph model, and (b) ordinary graph model.

to a certain 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:

$$\mathrm{AP} = \frac{\sum_i^N \mathrm{Precision@}i * \mathrm{corr}_i}{\text{Number of correctly recommended items}}, \tag{25}$$

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:

$$\mathrm{NDCG@}n = \frac{1}{\mathrm{IDCG}} \times \sum_{i=1}^{n} \frac{2^{r_i} - 1}{log_2(i+1)}, \tag{26}$$

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.

6.3 Performance Comparison

We use all evaluation metrics mentioned in Section 6.2 to measure the performance of each recommendation algorithm. Figure 4 shows the recall-precision curves for all the methods. We report the performance of all algorithms in terms of MAP, F1 and NDCG in Table IV (MAP and F1) and Table V (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

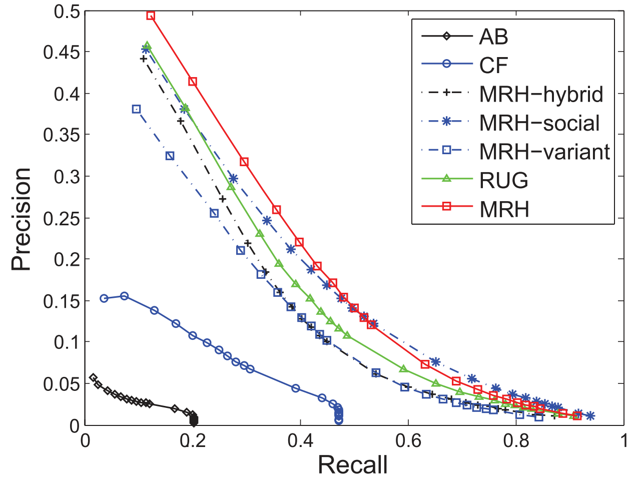


Fig. 4. Recall-Precision curves for all the methods.

Table IV. Comparison of Recommendation Algorithms in Terms of MAP and F1

	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-variant	0.2380	0.1442	0.1973	0.2285	0.2275	0.2079	0.1864	0.1599	0.1093
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

Bold typeface indicates the best performance. * indicates statistical significance at $p < 0.001$ compared to the second best.

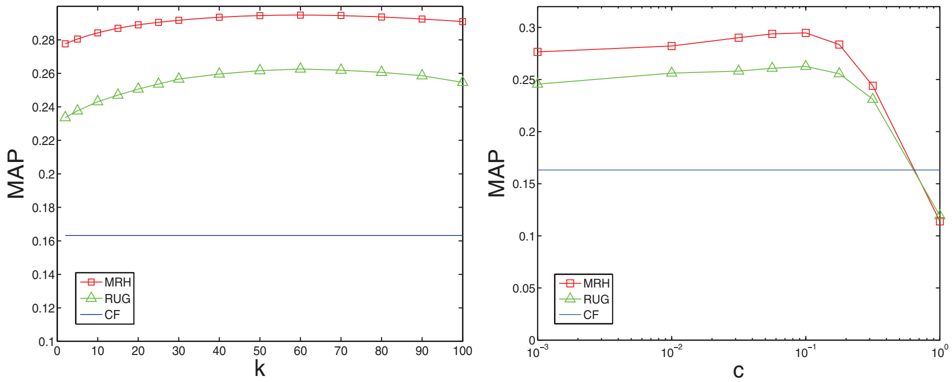
Table V. Comparison of Recommendation Algorithms in Terms of NDCG

	NDCG@5	NDCG@10	NDCG@30	NDCG@50	NDCG@100	NDCG@200
CF	0.1522	0.1713	0.2519	0.2987	0.3579	0.4120
AB	0.0733	0.0820	0.1241	0.1532	0.2027	0.2556
RUG	0.4849	0.4318	0.3826	0.4109	0.4587	0.5037
MRH-variant	0.3970	0.3626	0.3482	0.3820	0.4297	0.4715
MRH-hybrid	0.4587	0.4091	0.3640	0.3911	0.4346	0.4753
MRH-social	0.4759	0.4268	0.3866	0.4197	0.4763	0.5264
MRH	**0.5192***	**0.4650***	**0.4174***	**0.4484***	**0.4987***	**0.5419***

Bold typeface indicates the best performance. * indicates statistical significance at $p < 0.001$ compared to the second best.

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 [Celma 2006]. CF algorithm does not work well either. This is probably because the user-track matrix in our data set is highly sparse, with only about 0.6% nonzero 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-based information can improve the recommendation result, especially when recall is small. This is because acoustic-based information can alleviate some well-known problems associated with data sparseness in collaborative recommender systems, for example, user bias, nonassociation and cold-start problems [Li et al. 2007].



(a) k vs. MAP (b) c vs. MAP

Fig. 5. The parameter settings of k and c for music similarity relations. First, 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.

The superiority of MRH over MRH-variant indicates the normalized form of MRH is better, which is consistent with previous findings [Zhou et al. 2003a].

6.4 Exploring Parameter Settings

There are three parameters in our algorithm, that is, the number of nearest neighbors k mentioned in Section 5.3, c in Eq. (20) and α in Eq. (16).

To explore the influence of the parameters k and c, we use MAP as the evaluation metric and fix α to be 0.98. Figure 5 shows the results. First, we fix c at 0.1 empirically and let k vary. Figure 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. Figure 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 MRH algorithm and RUG [Zhou et al. 2003b]. In our performance comparision experiments, we just set α to be 0.98 for MRH, MRH-hybrid, MRH-social, MRH-variant and RUG empirically. We explore the optimal value of α here. Figure 6 shows the results by MAP. The MRH algorithm obtains the best performance when α is close to 0.97 and RUG obtains best performance when α is close to 0.96. It also can be seen, MRH outperforms RUG and CF algorithms in a wide range of parameter variation. When $\alpha > 0.98$, the performance drop dramatically. And when $\alpha = 0.999$ (i.e., the restart probability is close to 0), the performance becomes very bad. This is because the larger the value of α, the smaller the effect of the query. Restart probability 0 means that there is no relationship between the ranking results and the query.

6.5 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 listening 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 VI, we can see that inclusion relations significantly improve the recommendation performance. By using inclusion relations among

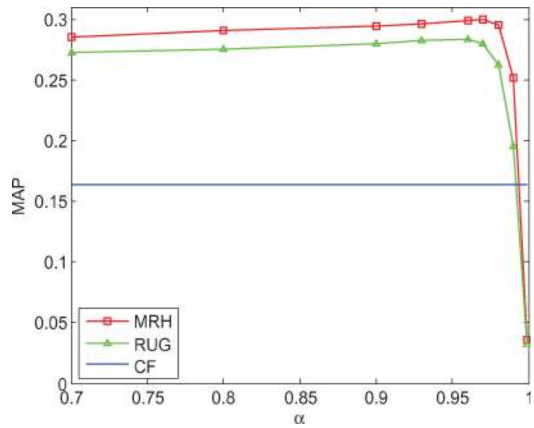


Fig. 6. Exploring the influence of the parameter α setting for MRH and RUG algorithms. We use MAP evaluation metric here.

Table VI. Comparison of MRH on Different Subsets of Social Information in Terms of MAP and F1

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

Bold typeface 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.

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' preferences 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.

6.6 Recommendation Examples

Table VII shows some recommendation examples for a few users. For each user, we list top 5 recommended music tracks by MRH, RUG and CF. As can be seen, the precision of MRH is higher than RUG and CF, but the top 5 recommended tracks of MRH and RUG are only from one or two artists centrally. That may be because of the contribution of inclusion relations. In practical applications, the diversity of recommended results should be considered. To solve this problem, the results of MRH should be re-treated. For example, choosing N tracks from different artists in top-ranked results for recommendation or suggesting users top N artists who has more tracks in top ranked results.

Table VIII shows that recommendations vary in different social groups. For each group, we list top 10 recommended music tracks by MRH. These are statistical results based on recommendations for all users in a particular group. As shown, for some special interest groups, top recommended results meet the users' tastes. For example, *Lupe Fiasco*, *2Pac*, *Wu-Tang Clan* et al. are all hip hop groups and *Iron Maiden*, *Slayer*, *Judas Priest* et al. are heavy metal or thrash metal bands. But for some

Table VII. Top 5 Recommended Tracks by MRH, RUG and CF, for Three Users

UserID	MRH	RUG	CF
136	(1)Lady GaGa-Poker Face	(1)Lady GaGa-Poker Face	(0)Lady GaGa-LoveGame
	(0)Miley Cyrus-Fly on the Wall	(1)Lady GaGa-Bad Romance	(0)Bon Jovi-Thank You For Loving Me
	(1)Lady GaGa-Bad Romance	(0)Miley Cyrus-Fly on the Wall	(1)Lady GaGa-Poker Face
	(1)Miley Cyrus-When I Look At You	(1)Miley Cyrus-When I Look At You	(0)Britney Spears-I Run Away
	(1)Lady GaGa-I Like It Rough	(0)Britney Spears-Womanizer	(1)Lady GaGa-Bad Romance
698	(1)3 Doors Down-It's Not Me	(1)Miley Cyrus-7 Things	(0)Bon Jovi-Born To Be My Baby
	(1)Simple Plan-Me Against the World	(1)Simple Plan-Me Against the World	(0)blink-182-Another Girl Another Planet
	(1)Simple Plan-Untitled	(0)Lady GaGa-Paparazzi	(1)Simple Plan-You Don't Mean Anything
	(1)Simple Plan-No Love	(1)3 Doors Down-It's Not Me	(1)Simple Plan-No Love
	(1)Miley Cyrus-7 Things	(0)Lady GaGa-Poker Face	(1)Simple Plan-Save You
1401	(1)In Flames-Only for the Weak	(1)In Flames-Only for the Weak	(0)Breaking Benjamin-Crawl
	(1)In Flames-Evil in a Closet	(1)In Flames-Evil in a Closet	(0)Slipknot-Vermilion Pt. 2
	(1)Seether-Pride	(1)In Flames-My Sweet Shadow	(0)Dropkick Murphys-Caps and Bottles
	(1)In Flames-My Sweet Shadow	(0)In Flames-The Mirror's Truth	(1)In Flames-My Sweet Shadow
	(1)In Flames-Like You Better Dead	(1)In Flames-Like You Better Dead	(0)Dropkick Murphys-(F)lannigan's Ball

1 indicates the track is in the test data (i.e., the user has listened to the track actually) and 0 otherwise.

Table VIII. Top 10 Recommended Music Tracks for a Few Social Groups by MRH

Group Name	Top 10 Recommended Music		
90's Hip-Hop	Lupe Fiasco-I Gotcha	2Pac-Tattoo Tears	Nas-It Ain't Hard to Tell
	Wu-Tang Clan-Method Man	Snoop Dogg-Gin And Juice	Wu-Tang Clan-Ain't Nuthing ...
	2Pac-How Do You Want It	2Pac-Never Had a Friend Like Me	2Pac-Panther Power
	2Pac-Toss It Up		
Thrash and Speed Metal	Iron Maiden-Sanctuary	Iron Maiden-Man on the Edge	Iron Maiden-Bring Your Daughter...
	Iron Maiden-Run To The Hills	Iron Maiden-2 Minutes to Midnight	Slayer-Divine Intervention
	Iron Maiden-Flight of Icarus	Lady GaGa-Poker Face	Judas Priest-Hell Bent for Leather
	Dire Straits-News		
1993-Born	Lady GaGa-Paparazz	Lady GaGa-Poker Face	Britney Spears-Womanizer
	Britney Spears-Gimme More	Cascada-Ready Or Not	Black Eyed Peas-Like That
	Slipknot-Pulse of the Maggots	Nas-Life's a Bitch	Slipknot-Three Nil
	The Prodigy-Mindfields		

social groups not based on special interests, such as *1993-Born*, the top recommended tracks are those general popular, such as tracks from *Lady GaGa* and *Britney Spears*.

7. CONCLUSIONS AND FUTURE WORK

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 via hypergraph (MRH). MRH constructs a hypergraph to model the multitype 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[5] and LinkedIn,[6] the preferences 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), 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 5.4, 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.

REFERENCES

ADOMAVICIUS, G. AND TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng. 17,* 6, 734–749.

AGARWAL, S. 2006. Ranking on graph data. In *Proceedings of the 23rd International Conference on Machine Learning*.

AGARWAL, S., BRANSON, K., AND BELONGIE, S. 2006. Higher order learning with graphs. In *Proceedings of the 23rd International Conference on Machine Learning*.

AUCOUTURIER, J.-J. AND PACHET, F. 2002. Scaling up music playlist generation. In *Proceedings of the IEEE International Conference on Multimedia and Expo*.

BERENZWEIG, A., LOGAN, B., W.ELLIS, D. P., AND WHITMAN, B. 2004. A large-scale evaluation of acoustic and subjective music-similarity measures. *Comput. Music J. 28,* 2, 63–76.

BREESE, J., HECKERMAN, D., AND KADIE, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*.

BU, J., TAN, S., CHEN, C., WANG, C., WU, H., AND HE, X. 2010. Music recommendation by unified hypergraph: Combining social media information and music content. In *Proceedings of the 18th ACM International Conference on Multimedia*.

BULÒ, S. R. AND PELILLO, M. 2009. A game-theoretic approach to hypergraph clustering. *Adv. Neural Inf. Proc. Syst. 22*.

CAI, R., ZHANG, C., ZHANG, L., AND MA, W.-Y. 2007. Scalable music recommendation by search. In *Proceedings of the 15th International Conference on Multimedia*.

CANO, P., KOPPENBERGER, M., AND WACK, N. 2005. Content-based music audio recommendation. In *Proceedings of the 13th ACM International Conference on Multimedia*.

CELMA, Ò. 2006. Foafing the music: Bridging the semantic gap in music recommendation. In *Proceedings of the 5th International Semantic Web Conference*.

CELMA, Ò. AND LAMERE, P. 2008. If you like the Beatles you might like...: A tutorial on music recommendation. In *Proceedings of the 16th ACM International Conference on Multimedia*.

CHEN, S., WANG, F., AND ZHANG, C. 2007. Simultaneous heterogeneous data clustering based on higher order relationships. In *Proceedings of the 7th IEEE International Conference on Data Mining Workshops*.

DIEDERICH, J. AND IOFCIU, T. 2006. Finding communities of practice from user profiles based on folksonomies. In *Proceedings of the 1st International Workshop on Building Technology Enhanced Learning Solutions for Communities of Practice*.

DONALDSON, J. 2007. A hybrid social-acoustic recommendation system for popular music. In *Proceedings of the ACM Conference on Recommender Systems*.

FRIEZE, A., KANNAN, R., AND VEMPALA, S. 2004. Fast Monte-Carlo algorithms for finding low-rank approximations. *J. ACM 51,* 6, 1025–1041.

GUAN, Z., BU, J., MEI, Q., CHEN, C., AND WANG, C. 2009. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *Proceedings of the 32nd ACM SIGIR Conference on Research and Development in Information Retrieval*.

[5]http://www.facebook.com.
[6]http://www.linkedin.com.

GUAN, Z., WANG, C., BU, J., CHEN, C., YANG, K., CAI, D., AND HE, X. 2010. Document recommendation in social tagging services. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*.

HARPALE, A. S. AND YANG, Y. 2008. Personalized active learning for collaborative filtering. In *Proceedings of the 31st ACM SIGIR Conference on Research and Development in Information Retrieval*.

HERLOCKER, J. L., KONSTAN, J. A., BORCHERS, A., AND RIEDL, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*.

KNEES, P., POHLE, T., SCHEDL, M., AND WIDMER, G. 2006. Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*.

KONSTAS, I., STATHOPOULOS, V., AND JOSE, J. M. 2009. On social networks and collaborative recommendation. In *Proceedings of the 32nd ACM SIGIR Conference on Research and Development in Information Retrieval*.

LI, Q., MYAENG, S. H., AND KIM, B. M. 2007. A probabilistic music recommender considering user opinions and audio features. *Infor. Process. Manag. 43,* 2, 473–487.

LIN, Y., SUN, J., CASTRO, P., KONURU, R., SUNDARAM, H., AND KELLIHER, A. 2009. Metafac: community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

LIU, N. N. AND YANG, Q. 2008. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st ACM SIGIR Conference on Research and Development in Information Retrieval*.

LOGAN, B. 2004. Music recommendation from song sets. In *Proceedings of the 5th International Conference on Music Information Retrieval*.

LOGAN, B. AND SALOMON, A. 2001. Music similarity function based on signal analysis. In *Proceedings of IEEE International Conference on Multimedia and Expo*.

LOVÁSZ, L. 1993. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty 2,* 1, 1–46.

MA, H., KING, I., AND LYU, M. R. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd ACM SIGIR Conference on Research and Development in Information Retrieval*.

MCKAY, C. AND FUJINAGA, I. 2008. Combining features extracted from audio, symbolic and cultural sources. In *Proceedings of the 9th International Conference on Music Information Retrieval*.

PAUWS, S., VERHAEGH, W., AND VOSSEN, M. 2006. Fast generation of optimal music playlists using local search. In *Proceedings of the 7th International Conference on Music Information Retrieval*.

RAGNO, R. J., BURGES, C. J. C., AND HERLEY, C. 2005. Inferring similarity between music objects with application to playlist generation. In *Proceedings of the 7th ACM SIGMM Workshop on Multimedia Information Retrieval*.

RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P., AND RIEDL, J. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*.

RHO, S., JUN HAN, B., AND HWANG, E. 2009. SVR-based music mood classification and context-based music recommendation. In *Proceedings of the 17th ACM International Conference on Multimedia*.

RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. 2000. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis. 40,* 2, 99–121.

SARWAR, B., KARYPIS, G., KONSTAN, J., AND REIDL, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*.

SEN, S., VIG, J., AND RIEDL, J. 2009. Tagommenders: Connecting users to items through tags. In *Proceedings of the 18th International Conference on World Wide Web*.

SONG, Y., ZHUANG, Z., LI, H., ZHAO, Q., LI, J., LEE, W.-C., AND GILES, C. L. 2008. Real-time automatic tag recommendation. In *Proceedings of the 31st ACM SIGIR Conference on Research and Development in Information Retrieval*.

SUN, L., JI, S., AND YE, J. 2008. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

SVIZHENKO, A., ANANTRAM, M. P., GOVINDAN, T. R., BIEGEL, B., AND VENUGOPAL, R. 2009. Two-dimensional quantum mechanical modeling of nanotransistors. *J. Appl. Phys. 91,* 4, 2343–2354.

SYMEONIDIS, P., RUXANDA, M., NANOPOULOS, A., AND MANOLOPOULOS, Y. 2008. Ternary semantic analysis of social tags for personalized music recommendation. In *Proceedings of the 9th International Conference on Music Information Retrieval*.

TAO, D., LIU, H., AND TANG, X. 2004. K-box: a query-by-singing based music retrieval system. In *Proceedings of the 12th ACM International Conference on Multimedia*.

TIEMANN, M. AND PAUWS, S. 2007. Towards ensemble learning for hybrid music recommendation. In *Proceedings of ACM Conference on Recommender Systems*.

TSO-SUTTERR, K. H. L., MARINHO, L. B., AND SCHMIDT-THIEME, L. 2008. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the ACM Symposium on Applied Computing*.

WILLIAMS, C. AND WILLIAMS, C. 2001. Using the nyström method to speed up kernel machines. In *Adv. Neural Inf. Proc. Syst. 13*.

YOSHII, K. AND GOTO, M. 2009. Continuous pLSI and smoothing techniques for hybrid music recommendation. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*.

YOSHII, K., GOTO, M., KOMATANI, K., OGATA, T., AND OKUNO, H. G. 2006. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *Proceedings of the 7th International Conference on Music Information Retrieval*.

ZHANG, Z., ZHOU, T., AND ZHANG, Y. 2009. Personalized recommendation via integrated diffusion on user-item-tag tripartite graphs. *Phys. A: Stat. Mech. Its Appl. 389,* 1, 179–186.

ZHOU, D., BOUSQUET, O., LAL, T. N., WESTON, J., AND SCHÖLKOPF, B. 2003a. Learning with local and global consistency. In *Adv. Neural Inf. Proc. Syst. 16*.

ZHOU, D., HUANG, J., AND SCHÖLKOPF, B. 2006. Learning with hypergraphs: Clustering, classification, and embedding. In *Adv. Neural Inf. Proc. Syst. 19*.

ZHOU, D., WESTON, J., GRETTON, A., BOUSQUET, O., AND SCHÖLKOPF, B. 2003b. Ranking on data manifolds. In *Adv. Neural Inf. Proc. Syst. 16*.

Received January 2011; revised May 2011; accepted August 2011