MUSIC RECOMMENDATION USING HYPERGRAPHS AND GROUP SPARSITY

Antonis Theodoridis, Constantine Kotropoulos, Yannis Panagakis

Department of Informatics
Aristotle University of Thessaloniki
Box 451, Thessaloniki 54124, Greece
Email: antochat@csd.auth.gr,{costas,panagakis}@aiia.csd.auth.gr

ABSTRACT

A challenging problem in multimedia recommendation is to model a variety of relations, such as social, friend, listening, or tagging ones in a unified framework and to exploit all these sources of information. In this paper, music recommendation problem is expressed as a hypergraph ranking problem, introducing group sparsity constraints. By doing so, one can control how the different data groups (i.e., sets of hypergraph vertices) affect the recommendation process. Experiments on a dataset collected from `Last.fm` demonstrate that the accuracy is significantly increased by exploiting the group structure of the data. Preliminary results are also presented for Greek folk music recommendation.

***Index Terms*—** music recommendation, social media information, hypergraph, group sparse optimization, music signal processing

1. INTRODUCTION

Recommender systems assist users to identify content, which may be of their interest from a potentially overwhelming set of choices [1]. Combined with the growth of the digital music market, the deployment of an effective music recommendation system has become a challenging research topic. The various music recommendation methods developed so far, can be classified into three basic categories: collaborative-filtering ones, content-based ones, and hybrid recommendation ones. Collaborative filtering methods are popular, as a result of the growth of social networking [2–4]. Items are recommended by exploiting user relations, assuming that similar users will have similar preferences. Content-based recommendation methods exploit any meta-data information of the items (i.e., the song writer, singer, musical genre or style) [5] and acoustic features extracted from the audio content [6–9]. Due to the semantic gap between the low level acoustic features and the high level music concepts, content-based methods lead to inefficient recommendations. Hybrid methods attempt to combine content-based recommendation with collaborative filtering techniques [10, 11].

Here, we are interested in a hybrid method treating music recommendation as a hypergraph ranking problem. That is, different kinds of objects (e.g., users, tracks, tags) are viewed as different types of hypergraph vertices, which participate in the same hyperedge. Hyperedges containing such multi-type objects can model high-order relations between the just mentioned objects. A novel recommendation method is proposed extending the framework proposed by Bu *et al*. [12] in order to incorporate group sparsity constraints, enabling thus the exploitation of the group structure of the data. A proper optimization problem is defined and solved by employing the Linearized Alternating Direction Method [13]. The F_1 measure of the proposed method at various ranking positions is demonstrated to be higher than that of the method in [12] and collaborative filtering.

The outline of the paper is as follows. Ranking on a hypergraph with group sparsity regularization is detailed in Section 2. In Section 3, music recommendation on `Last.fm` is explained. In particular, the dataset, the calculation of the audio track similarities, the construction of the hypergraph, and recommendation results are described. Preliminary results for Greek folk music recommendation are included in Section 4. Conclusions and future work directions are summarized in Section 5.

2. GROUP SPARSE REGULARIZATION FOR RANKING ON A HYPERGRAPH

A hypergraph $G(V, E, w)$ is defined as a set of vertices V and hyperedges E, to which a weight function $w : E \rightarrow \mathbb{R}$ is assigned [14]. Each hyperedge $e \in E$ contains an arbitrary number of vertices $v \in V$, which is defined as the hyperedge degree $\delta(e) = |e|$, with the set cardinality denoted as $|.|$. Ordinary graphs could be described as hypergraphs with a hyperedge degree equal to 2. Similarly, the degree of a vertex v can be defined as $\delta(v) = \sum_{e \in E | v \in e} w(e)$. Let $\mathbf{H} \in \mathbb{R}^{|V| \times |E|}$ be the vertex to hyperedge incidence matrix, having elements $H(v, e) = 1$ if $v \in e$ and 0 otherwise. The following diagonal matrices are defined: the vertex degree matrix $\mathbf{D}_u$, the hyperedge degree matrix $\mathbf{D}_e$ of size $|V| \times |V|$ and $|E| \times |E|$, respectively as well as the $|E| \times |E|$ matrix $\mathbf{W}$ containing the hyperedge weights. The ℓ_2 norm of a vector is denoted by $\|.\|_2$ and $\mathbf{I}$ is the identity matrix of compatible dimensions.

Let $\mathbf{A} = \mathbf{D}_u^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_u^{-1/2}$ and $\mathbf{L} = \mathbf{I} - \mathbf{A}$ be the positive semi-definite hypergraph Laplacian matrix. For a real valued ranking vector $\mathbf{f} \in \mathbb{R}^{|V|}$, one seeks to minimize $\Omega(\mathbf{f}) = \frac{1}{2}\mathbf{f}^T\mathbf{L}\mathbf{f}$ requiring all vertices with the same value in ranking vector $\mathbf{f}$ to be strongly connected [15]. Vertices participating in many common hyperedges should have similar ranking values. The just mentioned optimization problem was extended by including the ℓ_2 regularization norm between the ranking vector $\mathbf{f}$ and the query vector $\mathbf{y} \in \mathbb{R}^{|V|}$ in [12].

The vertex set V in the hypergraph is made by the concatenation of sets of objects of different kind, such as users, user groups, tracks, or tags. Let each set of objects define a group. Clearly, each group contributes differently to the ranking procedure. Accordingly, a Group Lasso regularizing term is more appropriate [16] than the ℓ_2 norm one. The replacement of the ℓ_2 norm regularization by the Group Lasso term is proposed here. If the hypergraph vertices are grouped into S non-overlapping groups, the ranking recommendation should be optimized by assigning different weights γ_s, $s = 1, 2, \ldots, S$ to each group, yielding the following objective function to be minimized:

$$Q(\mathbf{f}) = \Omega(\mathbf{f}) + \vartheta \sum_{s=1}^{S} \sqrt{\gamma_s\, (\mathbf{f}-\mathbf{y})^T \mathbf{K}_s (\mathbf{f}-\mathbf{y})}. \tag{1}$$

In (1), ϑ is the regularizing parameter and $\mathbf{K}_s$ is the $|V| \times |V|$ diagonal matrix with elements admitting the value 1 for the vertices, which belong to the s-th group. The recommendation problem can now be expressed as:

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}}\, Q(\mathbf{f}). \tag{2}$$

Let $\mathbf{x} = \mathbf{f} - \mathbf{y}$. By introducing the auxiliary variable $\mathbf{z} = \mathbf{x}$, (2) can be rewritten as:

$$\begin{aligned} &\underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2}(\mathbf{x}+\mathbf{y})^T \mathbf{L}(\mathbf{x}+\mathbf{y}) + \vartheta \sum_{s=1}^{S} \sqrt{\gamma_s\, \mathbf{z}^T \mathbf{K}_s \mathbf{z}} \\ &\text{s.t. } \mathbf{z} = \mathbf{x}. \end{aligned} \tag{3}$$

The solution of (3) can be obtained by minimizing the augmented Lagrangian function

$$\begin{aligned} \mathcal{L}(\mathbf{x},\mathbf{z},\boldsymbol{\lambda}) = &\frac{1}{2}(\mathbf{x}+\mathbf{y})^T \mathbf{L}(\mathbf{x}+\mathbf{y}) + \vartheta \sum_{s=1}^{S} \sqrt{\gamma_s \mathbf{z}^T \mathbf{K}_s \mathbf{z}} \\ &+ \boldsymbol{\lambda}^T(\mathbf{z}-\mathbf{x}) + \frac{\mu}{2}\|\mathbf{z}-\mathbf{x}\|^2. \end{aligned} \tag{4}$$

where $\boldsymbol{\lambda}$ is the vector of the Lagrange multipliers, which is updated at each iteration and μ is a parameter regularizing the violation of the constraint $\mathbf{x} = \mathbf{z}$. (4) can be solved by the Alternating Directions Method [13] as shown in Algorithm 1. Solving for $\mathbf{x}^{t+1}$ in line 3 of Algorithm 1 is a convex minimization, requiring the gradient of $\mathcal{L}(\mathbf{x},\mathbf{z},\boldsymbol{\lambda})$ with respect to $\mathbf{x}$ is vanishing:

Algorithm 1 Alternating Directions Method

```
1: Given x^t, z^t and λ^t.
2: Set tolerance ε and initialize μ.
3: x^{t+1} ← argmin_x L(x^t, z^t, λ^t)
4: z^{t+1} ← argmin_z L(x^{t+1}, z, λ^t)
5: if ||z − x||^2 > ε then
6:     λ^{t+1} ← λ^t + μ(z^{t+1} − x^{t+1})
7:     μ^{t+1} = min(1.1μ^t, 10^6)
8: else
9:     return x^{t+1}, z^{t+1}.
10:    f = x^{t+1} + y
11: end if
```

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = (\mathbf{I}-\mathbf{A})(\mathbf{x}+\mathbf{y}) - \boldsymbol{\lambda} + \mu(\mathbf{x}-\mathbf{z}) = \mathbf{0} \tag{5}$$

(5) yields

$$\mathbf{x}^{t+1} = (\mathbf{L} + \mu\mathbf{I})^{-1}(\boldsymbol{\lambda}^t + \mu\mathbf{z}^t - \mathbf{L}\mathbf{y}). \tag{6}$$

The minimization problem described in line 4 of Algorithm 1 can be expressed as

$$\min_{\mathbf{z}} \mu \Big\{ \frac{\vartheta}{\mu} \sum_{s=1}^{S} \sqrt{\gamma_s} \sqrt{\mathbf{z}^T \mathbf{K}_s \mathbf{z}} + \frac{1}{2}\|\mathbf{z} - (\mathbf{x}^{t+1} - \frac{1}{\mu}\boldsymbol{\lambda}^t)\|_2^2 \Big\}. \tag{7}$$

For $\mathbf{K}_s$ a diagonal matrix of ones, the term $\sqrt{\gamma_s}\|\mathbf{K}_s^{1/2}\mathbf{z}\|_2$ can be rewritten as:

$$\sqrt{\gamma_s}\|\mathbf{K}_s^{1/2}\mathbf{z}\|_2 = \sqrt{\sum_{j=1}^{|V|} \gamma_s\, K(s,j)\, z_j^2}. \tag{8}$$

It is easily seen that (7) for the jth element of $\mathbf{z}$ yields

$$\frac{1}{2}\big(z_j - (x_j^{t+1} - \frac{1}{\mu}\lambda_j^t)\big)^2 + \frac{\vartheta}{\mu}\sum_{s=1}^{S}\sqrt{\gamma_s}\, K(s,j)\, |z_j| \tag{9}$$

which is solved by applying the soft-thresholding operator [17]

$$z_j = \frac{r_j}{|r_j|} \max\Big(0, |r_j| - \frac{\vartheta}{\mu}\sum_{s=1}^{S}\sqrt{\gamma_s}\, K(s,j)\Big) \tag{10}$$

where $r_j = x_j^{t+1} - \frac{1}{\mu}\lambda_j^t$.

3. WESTERN MUSIC RECOMMENDATION ON A HYPERGRAPH

3.1. Dataset description

A dataset was created by collecting real data from `Last.fm`. To create the list of users, the 450 top artists were selected and their top 50 user fans where concatenated in a user set. This user set was later reduced based on the track and tag count of each user, yielding a final set of 1389 users. To create the track set, the 500 top played tracks for each user were concatenated in a list, from which 1765 unique tracks were selected based on their popularity among the users. Finally, we collected the tagging relations of each user and retained the 1711 most frequent unique tags. By using Porter's stemming algorithm [18] and calculating next the edit distance [19] between the tag pairs, all synonyms have been removed from the tags vocabulary (i.e., pairs, such as hardrock and hard rock or 90s and 1990s were merged). The final size of all sets and the various relations is described in Tables 1 and 2, respectively.

Table 1. Dataset objects, notations, and counts.

Objects	Notations	Count
Users	U	1389
Groups	Gr	10
Tracks	Tr	1765
Tags	Ta	1711

Table 2. Dataset relations between the objects.

Relations	Notations	Count
Listening relations	R_1	68774
Tagging relations	R_2	48566
Friendship relations	R_3	13890
Similarities between tracks	R_4	-

3.2. Audio-track similarities

Audio content similarities are captured thanks to 20 mel frequency cepstral coefficients (MFCCs) used to encode the timbral properties of the music signal. They are calculated by employing frames of duration 23ms with a hop size of 11.5 ms and a 42-band filter bank. A Gaussian Mixture Model (GMM) is created for each track with 30 components trained using the Expectation-Maximization (EM) algorithm as in [20]. The distances between the GMM's are computed by using the Earth Movers' Distance [21], yielding the audio-track similarities.

3.3. Hypergraph construction

The object set includes the users, the user groups, the tags, and the tracks, forming the vertex set $V = U \cup G \cup Ta \cup Tr$ of the hypergraph. The cardinality of these sets can be seen in Table 1. The set of hyperedges is defined as in [12]. The incidence matrix of the unified hypergraph $\mathbf{H}$ is shown in Table 3. Its size is 4875×146885 elements. In particular:

- $E^{(1)}$: The hyperedge represents a pairwise friendship relation between users. In this case its weight value is set to 1. The incidence matrix of the hypergraph $UE^{(1)}$ has a size of 1389×13890 elements.
- $E^{(2)}$: The hyperedge represents a group of users. It contains all the vertices of the corresponding users, as well as the vertice corresponding to the group object. In this case its weight value is also set to 1. The incidence matrix of the hypergraph $UE^{(2)} - GE^{(2)}$ has a size of 1399×13890 elements.
- $E^{(3)}$: The hyperedge contains a user and a music track, representing a user-track listening relation. The hyperedge weight $w(e_{ij}^{(3)})$ is defined as the number of times the particular user u_i has listened to the track tr_j, normalized as follows to eliminate the bias:

$$w(e_{ij}^{(3)})' = \frac{w(e_{ij}^{(3)})}{\sqrt{\sum_{k=1}^{|Tr|} w(e_{ik}^{(3)})}\sqrt{\sum_{l=1}^{|U|} w(e_{lj}^{(3)})}} \qquad (11)$$

  and further scaled as $w(e_{ij}^{(3)})^* = \frac{w(e_{ij}^{(3)})'}{ave(w(e_{ij}^{(3)})')}$, where $ave(w(e_{ij}^{(3)'}))$ the average of normalized weights of the particular user u_i. The incidence matrix of the hypergraph $UE^{(3)} - T_rE^{(3)}$ has a size of 3154×68774 elements.
- $E^{(4)}$: The hyperedge contains three vertices, a user, a tag and a music track, representing a tagging relation. In this case, its weight is set to be 1. The incidence matrix of the hypergraph $UE^{(4)} - T_aE^{(4)} - T_rE^{(4)}$ has a size of 4865×48566 elements.
- $E^{(5)}$: The hyperedge contains two vertices which represent two music tracks, with its weight $w(e_{ij}^{(5)})$ set as the similarity between tracks tr_i and track tr_j, normalized as follows to eliminate the bias $w(e_{ij}^{(5)})' = \frac{w(e_{ij}^{(5)})}{max(w(e^{(5)}))}$.

Based on this incidence matrix $\mathbf{H}$, the weight matrix $\mathbf{W}$, the vertex degree matrix $\mathbf{D}_u$ and the hyperedge degree matrix $\mathbf{D}_e$ are computed. The ranking vector $\mathbf{f}^*$ is derived by solving (2), after setting the values of the query vector $\mathbf{y}$, the group weights γ_s, and the regularization parameter ϑ. The query vector $\mathbf{y}$ is initialized by setting the entries corresponding to the target user and all objects (users, user groups, tags, and tracks), which are connected to this user to 1. The query vector $\mathbf{y}$ has a length of 4875 elements. If for example, the query user has listened to a certain track, the value of the element of $\mathbf{y}$ corresponding to this track is set to 1. The group weights γ_s are initially set to 1, treating all groups equally and then are empirically adjusted in order to examine the contribution of each group separately. The group weight values vary in the range $[0, 1]$ with a step of 0.1. The regularization parameter ϑ is a constant and does not affect the ranking results.

Table 3. The structure of the hypergraph incidence matrix $\mathbf{H}$ and its sub-matrices.

	$E^{(1)}$	$E^{(2)}$	$E^{(3)}$	$E^{(4)}$	$E^{(5)}$
U	$UE^{(1)}$	$UE^{(2)}$	$UE^{(3)}$	$UE^{(4)}$	0
G	0	$GE^{(2)}$	0	0	0
T_a	0	0	0	$T_aE^{(4)}$	0
T_r	0	0	$T_rE^{(3)}$	$T_rE^{(4)}$	$T_rE^{(5)}$

The resulting ranking vector $\mathbf{f}^*$ has a size of 4875 elements and the same structure with the query vector. Only the values corresponding to music tracks are used for music recommendation with the top ranked tracks being recommended to the user.

3.4. Experiments

The proposed algorithm is compared to a user-based Collaborative Filtering (CF) algorithm [22] and the Music Recommendation on Hypergraph (MRH) algorithm [12], including listening relations (R_1), tagging relations (R_2), friends relations (R_3), and music similarity relations (R_4). The Recall-Precision and the F_1 measure are used as figures of merit. Recall is defined as the fraction of relevant items that are retrieved (here, recommended), while precision is the fraction of retrieved (here, recommended) items that are relevant. The F_1 measure is the weighted harmonic mean of precision and recall, which measures the effectiveness of recommendation when treating precision and recall as equally important.

A test set is defined, containing 20% of the listening relations and the remaining 80% of the listening relations create the training set. Before applying any recommendation method for a certain user, all his/her relations with the tracks of the training set are removed.

A set of 100 random users is selected. In Fig. 1, the Averaged Recall-Precision curves are plotted, by averaging the Recall-Precision curves over these 100 users, for all compared methods. The user-based CF algorithm has the worst performance, as expected, due to the high sparsity of the user and track data and the fact that it does not exploit any acoustic similarity between the tracks. The MRH algorithm models the high-order relations between the users, the tracks, the tags, and the acoustic similarities and thus achieves better results than the CF. In the MRH, the problems associated with data sparsity, like the cold start problem or the user bias, are alleviated thanks to the additional information on acoustic similarity, user friendship relations, and tagging relations exploited.

The proposed method, denoted as Query Group Sparse (QGS), inherits the advantages of the MRH, but it exploits furthermore the group structure of the hypergraph by assigning unique weights γ_s to each group (users, user groups, tags, and tracks). The averaged recall-precision curves for three different weight settings are presented in Fig. 1. The QGS-(1) curve treats all groups equally by assigning equal weights γ_s to the groups. The QGS-(2) curve sets

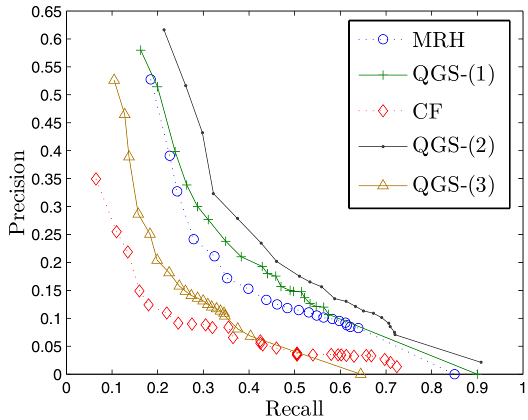

Fig. 1. Averaged Recall-Precision curves of 100 users for all compared algorithms.

larger weights to users and tracks than user groups and tags The weights for users, user groups, tags, and tracks are set to 0.8, 0.2, 0.2, and 0.9, respectively. The QGS-(3) curve weights more heavily the users and user groups than the tags and tracks. In this case, the respective weights are set to 0.9, 0.6, 0.1, and 0.1. Clearly, the weights have been empirically set here in order to derive qualitative conclusions. In most cases, the larger weights are set for users and tracks, the better recommendation performance is observed. The superiority of the QGS-(2) method indicates that the information on tagging contributes less to the music recommendation process. These conclusions can be verified by studying the F_1-measure for various ranking positions summarized in Table 4.

Table 4. F_1 measures at ranking positions 1, 5, 10, and 20. The best results for each setting are marked with *.

	$F_1@1$	$F_1@5$	$F_1@10$	$F_1@20$
CF	0.106	0.155	0.131	0.085
MRH	0.273	0.256	0.196	0.180
QGS(1)	0.253	0.284	0.251	0.197^*
QGS(2)	0.315^*	0.286^*	0.262^*	0.184
QGS(3)	0.173	0.274	0.185	0.196

4. GREEK FOLK MUSIC RECOMMENDATION

A dataset of Greek folk music recordings, coined as THALIS-ERASITECHNIS dataset, was collected from `erasitechnis` site[1]. This very primitive dataset contains 227 tracks, 235 tags, and 41 users. Preliminary averaged recall-precision curves for 25 users are shown in Figure 2. Three groups were included: tags, tracks, and users. Training and testing was performed similarly to the experiments conducted in `Last.fm`. The QGS-THALIS(1) curve sets larger weights to tags or tracks than the users. The weights for users, tags, and tracks are set to 0.3, 0.9, and 0.9, respectively. The QGS-THALIS(2) curve weighs more heavily the users than the tags or tracks by setting the respective weights to 0.9, 0.4, and 0.4. The reduced accuracy of the QGS-THALIS(2) curve verifies that the sparse user information does not contribute efficiently to the recommendation process. By adding more hyper-relations to the hypergraph, the recommendation accuracy could be further improved.

[1] `http://erasitechnis-aiia.web.auth.gr/music`

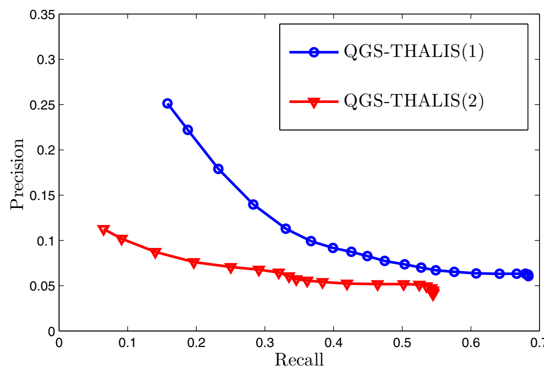

Fig. 2. Averaged Recall-Precision curves of 25 users of the THALIS-ERASITECHNIS database.

5. CONCLUSION AND FUTURE WORK

We addressed music recommendation as a ranking problem on a unified hypergraph by modelling all available types of information and the high-order relations among them. We solved the ranking problem by using a group sparse optimization approach. The experimental results indicated that by exploiting the group structure of the hypergraph, the recommendation accuracy was significantly improved.

Further experiments could include more groups in the unified hypergraph, i.e., the information on genre, artists, albums, or social networks. Such information is now easily accessible through online music platforms. By using non-overlapping groups, we assume that each group affects the recommendation process separately. However, certain groups contain mutual and highly correlated information. Accordingly, future work will investigate how overlapping groups could be exploited in music recommendation. Finally, this approach can be possibly extended for friend or group recommendation, using the ranking values corresponding to the users or user groups, respectively.

Acknowledgment

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operation Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS-UOA-ERASITECHNIS MIS 375435.

6. REFERENCES

[1] Y. Song, S. Dixon, and M. Pearce, "A survey of music recommendation systems and future perspectives," in *Proc. 9th Int. Symp. Computer Music Modelling and Retrieval*, 2012.

[2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th ACM Int. Conf. World Wide Web*, 2001, pp. 285–295.

[3] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances Artificial Intelligence*, pp. 4:2–4:2, 2009.

[4] K. Tso-Sutter, L. Marinho, and L. Schmidt-Thieme, "Tag-aware recommender systems by fusion of collaborative filtering algorithms," in *Proc. ACM Symp. Applied Computing*, 2008, pp. 1995–1999.

[5] R. Cai, C. Zhang, L. Zhang, and W. Ma, "Scalable music recommendation by search," in *Proc. 15th ACM Int. Conf. Multimedia*, 2007, pp. 1065–1074.

[6] Q. Li, B. Kim, D. Guan, and D. Oh, "A music recommender based on audio features," in *Proc. 27th ACM Int. Conf. Research and Development in Information Retrieval*, 2004, pp. 532–533.

[7] D. Bogdanov, J. Serrá, N. Wack, P. Herrera, and X. Serra, "Unifying low-level and high-level music similarity measures," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 687–701, 2011.

[8] P. Cano, M. Koppenberger, and N. Wack, "Content-based music audio recommendation," in *Proc. 13th ACM Int. Conf. Multimedia*, 2005, pp. 211–212.

[9] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[10] S. Strickroth and N. Pinkwart, "High quality recommendations for small communities: the case of a regional parent network," in *Proc. 6th ACM Conf. Recommender Systems*, 2012, pp. 107–114.

[11] X. Li and T. Murata, "Multidimensional clustering based collaborative filtering approach for diversified recommendation," in *Proc. 7th Int. Conf. Computer Science Education*, 2012, pp. 905–910.

[12] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, Z. Lijun, and X. He, "Music recommendation by unified hypergraph: combining social media information and music content," in *Proc. ACM Conf. Multimedia*, 2010, pp. 391–400.

[13] Z. Lin, R. Lui, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Neural Information Processing Systems*, 2011, pp. 612–620.

[14] C. Berge, *Graphs and Hypergraphs*, Elsevier Science Ltd, 1985.

[15] S. Agarwal, K. Branson, and S. Belongie, "Higher order learning with graphs," in *Proc. 23rd Int. Conf. Machine Learning*, 2006, pp. 17–24.

[16] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[17] Z. Qin and K. Scheinberg, "Efficient block-coordinate descent algorithms for the group Lasso," *Industrial Engineering*, pp. 1–21, 2010.

[18] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[19] E.S. Ristad and P. Yianilos, "Learning string-edit distance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, 1998.

[20] E. Pampalk, "A MATLAB toolbox to compute music similarity from audio," in *Proc. 5th Int. Conf. Music Information Retrieval*, 2004, pp. 254–257.

[21] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[22] I. Konstas, V. Stathopoulos, and J. Jose, "On social networks and collaborative recommendation," in *Proc. 32nd Int. Conf. ACM SIGIR Research and Development in Information Retrieval*, 2009, pp. 195–202.