PERSONALIZED MUSIC TAGGING USING RANKING ON HYPERGRAPHS

Konstantinos Pliakos, Constantine Kotropoulos

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

ABSTRACT

Social tagging enables users of social media sharing platforms to annotate multimedia items by employing arbitrary keywords (i.e., tags), which describe better the multimedia content. Several applications, such as personalized multimedia recommendation or music genre classification, to name a few, benefit from tagging. Clearly, tagging aims at bridging the semantic gap between human concepts and content retrieval exploiting low-level features extracted from the multimedia. Here, the problem of personalized tag recommendation is addressed in a "query and ranking" manner on hypergraphs. This way, the relationships between the different object types, such as user friendships, user groups, music tracks and tags are captured and tags are recommended for a certain track to a user. Ranking on hypergraphs is studied by enforcing either ℓ_2 norm regularization or group sparsity. Experiments on a dataset collected from `Last.fm` demonstrate a promising tag recommendation accuracy.

Index Terms— Music Signal Processing, Tagging, Hypergraph, Group Sparse Optimization

1. INTRODUCTION

Social tagging systems of Web 2.0 applications, like Youtube[1] or Last.fm[2] have become increasingly popular in the last years, leading to a large amount of generated tags. Users are given the opportunity to annotate different types of items (i.e. web sites, media content, artists, products) with their own tags, which according to their personal opinion describe better the items. Evidence was reported recently, that the context revealed by user-ratings for music similarity, the network graph relationships in tagging, and features not derived from media content for recommendations, are more crucial than content descriptive features [1]. Such findings by no means de-emphasize the role of content analysis, but undoubtedly highlight that content analysis is difficult and perhaps underline the need for better feature analysers. Personalized tag recommendation aims at supporting large-scale retrieval and recommendation systems by expanding the set of tags annotating the system resources and improving the user satisfaction.

Various content-based automatic music tagging systems have been proposed [2–5]. Most of the aforementioned systems resort to the so-called bag-of-features approach, which models the audio signals by the long-term statistical distribution of their short-time spectral features. These features are then fed into machine learning algorithms that associate tags with audio features. For instance, audio tag prediction was treated as a set of binary classification problems, where standard classifiers, such as the Support Vector Machines [3] or Ada-Boost [2] were applied. Other methods attempted to infer the correlations or joint probabilities between the tags and the low-level acoustic features extracted from audio [5] or treated tagging as a multi-class classification problem [4]. Closely related to graph-based approaches are the tensor factorization models that were also employed for personalized tag recommendation [6, 7]. A graph-based ranking algorithm for different object types was proposed in [8], taking into consideration both the document relevance and the preferences of each user. The tag recommendation was addressed in a "query and ranking" manner, where objects and users were treated as part of the query.

Motivated by [8], here a generalization of graphs, called hypergraphs, is employed to model the high-level relationships between the different object types, i.e., user friendships, groups of users, music tracks and tags. A hypergraph is defined as a set of vertices and hyperedges linking more than two vertices. In this way, several relations between the objects are captured, avoiding any information loss. Moreover, a regularization framework is proposed, where tagging is treated as a ranking problem on hypergraphs subject to proper constraints, such as the ℓ_2 norm between the query vector and the vector of the ranking scores (used previously for music recommendation [9]) or group sparsity [10] in order to control how the individual object groups (i.e., user friendships, user groups, tracks, tags) affect the personalized music tagging. By enforcing group sparsity in the solution, one can take advantage of the hypergraph structure and examine how each object group and its associated relations affect the personalized tagging.

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 [11]. Each hyperedge $e \in E$ contains an arbitrary number of vertices $v \in V$ and the hyperedge degree $\delta(e) = |e|$ is its cardinality. 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 $\mathbf{A} = \mathbf{D}_u^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_u^{-1/2}$, then $\mathbf{L} = \mathbf{I} - \mathbf{A}$ is the positive semi-definite Laplacian matrix of the hypergraph. 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

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

978-1-4799-2890-3/14/$31.00 ©2014 IEEE

$\mathbf{f}$ to be strongly connected [12]. The just mentioned optimization problem was extended by including the ℓ_2 regularization norm between the ranking vector $\mathbf{f}$ and the query vector $\mathbf{y} \in \mathbb{R}^{|V|}$ for music recommendation in [9]. Clearly, personalized music tagging can be casted as the solution of the optimization problem:

$$\tilde{Q}(\mathbf{f}) = \Omega(\mathbf{f}) + \vartheta \, ||\mathbf{f} - \mathbf{y}||_2^2. \tag{1}$$

Then, personalized music tagging seeks for the ranking vector

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

where ϑ is a regularizing parameter. The solution of (2) is [9]:

$$\mathbf{f}^* = \frac{\vartheta}{1+\vartheta} \left(\mathbf{I} - \frac{1}{1+\vartheta}\mathbf{A}\right)^{-1} \mathbf{y}. \tag{3}$$

The vertex set V in the hypergraph is made by the concatenation of sets of objects of different type, such as user friendships, user groups, tracks and tags. Let each set of objects define a group. Clearly, each object group contributes differently to the ranking procedure. Accordingly, a Group Lasso regularizing term is more appropriate than the ℓ_2 norm [13]. If the hypergraph vertices are split into S non-overlapping object groups (users, user groups, tags, tracks) the ranking recommendation should be optimized by assigning different weights γ_s, $s = 1, 2, \ldots, S$ to each object group, yielding the following objective function to be minimized:

$$Q(\mathbf{f}) = \Omega(\mathbf{f}) + \vartheta \sum_{s=1}^{S} \sqrt{\gamma_s \, (\mathbf{f} - \mathbf{y})^T \mathbf{K}_s (\mathbf{f} - \mathbf{y})}. \tag{4}$$

In (4), ϑ is also a 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 object group. The personalized tag recommendation problem is now expressed as:

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}} \, Q(\mathbf{f}). \tag{5}$$

Let $\mathbf{x} = \mathbf{f} - \mathbf{y}$. By introducing the auxiliary variable $\mathbf{z} = \mathbf{x}$, (5) can be rewritten as:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2}(\mathbf{x}+\mathbf{y})^T \mathbf{L}(\mathbf{x}+\mathbf{y}) + \vartheta \sum_{s=1}^{S} \sqrt{\gamma_s \, \mathbf{z}^T \mathbf{K}_s \mathbf{z}} \quad \text{s.t. } \mathbf{z} = \mathbf{x}. \tag{6}$$

The solution of (6) can be obtained by minimizing the augmented Lagrangian function

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

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 $\mathbf{x} = \mathbf{z}$. (7) can be solved by the Alternating Directions Method [14] as shown in Algorithm 1. Solving for $\mathbf{x}^{t+1}$ in line 3 of Algorithm 1 yields

$$\mathbf{x}^{t+1} = (\mathbf{L} + \mu^t \mathbf{I})^{-1}(\boldsymbol{\lambda}^t + \mu^t \mathbf{z}^t - \mathbf{L}\mathbf{y}). \tag{8}$$

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

$$\min_{\mathbf{z}} \mu^t \left\{ \frac{\vartheta}{\mu^t} \sum_{s=1}^{S} \sqrt{\gamma_s} \sqrt{\mathbf{z}^T \mathbf{K}_s \mathbf{z}} + \frac{1}{2}\|\mathbf{z} - (\mathbf{x}^{t+1} - \frac{1}{\mu^t}\boldsymbol{\lambda}^t)\|_2^2 \right\}. \tag{9}$$

Algorithm 1 Alternating Directions Method

1: Given $\mathbf{x}^t$,$\mathbf{z}^t$ and $\boldsymbol{\lambda}^t$.
2: Set tolerance ϵ and initialize μ.
3: $\mathbf{x}^{t+1} \leftarrow \underset{\mathbf{x}}{\operatorname{argmin}} \mathcal{L}(\mathbf{x}, \mathbf{z}^t, \boldsymbol{\lambda}^t)$
4: $\mathbf{z}^{t+1} \leftarrow \underset{\mathbf{z}}{\operatorname{argmin}} \mathcal{L}(\mathbf{x}^{t+1}, \mathbf{z}, \boldsymbol{\lambda}^t)$
5: **if** $\|\mathbf{z} - \mathbf{x}\|_2^2 > \epsilon$ **then**
6: $\quad \boldsymbol{\lambda}^{t+1} \leftarrow \boldsymbol{\lambda}^t + \mu^t(\mathbf{z}^{t+1} - \mathbf{x}^{t+1})$
7: $\quad \mu^{t+1} = \min(1.1\mu^t, 10^6)$
8: **else**
9: $\quad$ return $\mathbf{x}^{t+1}$, $\mathbf{z}^{t+1}$.
10: $\quad \mathbf{f} = \mathbf{x}^{t+1} + \mathbf{y}$
11: **end if**

By applying the soft-thresholding operator [15], we get

$$z_j = \frac{r_j}{||\mathbf{r}_s||_2} \max\left(0, ||\mathbf{r}_s||_2 - \vartheta\mu^t \frac{1}{\sqrt{\gamma_s}}\right) \tag{10}$$

where $r_j = x_j^{t+1} - \frac{1}{\mu^t}\lambda_j^t$, s is the object group where j-th vertex belongs, and $\mathbf{r}_s$ is the segment of $\mathbf{r}$ corresponding to the s-th object group.

3. PERSONALIZED TAG RECOMMENDATION

3.1. Dataset description

The dataset had to be abundant in tagging and listening relations for evaluation purposes. It was created by collecting real data from `Last.fm`. First of all, in order to create the list of users, the 450 top artists were selected and their top 50 user fans were concatenated in a user set. This set was then reduced taking into account 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, the tagging relations of each user were collected and 1711 unique tags were retained. At this point, we have to mention that this choice was guided by the fact that tagging relations had to be essentially triples (user, tag, track), so only the tags that were connected to the previously obtained tracks were kept. By using Porter's stemming algorithm [16] and calculating the edit distance [17] between the tag pairs, various morphological variants of the tags were eliminated. The size of all dataset objects is summarized in Table 1.

Table 1. Dataset objects, notations, and size.

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

3.2. Audio-track similarities

The 20 mel frequency cepstral coefficients (MFCCs) were used to encode the timbral properties of the music signal. Frames of duration 23ms with a hop size of 11.5 ms and a 42-band filter bank were used for their calculation. A Gaussian Mixture Model (GMM) was created for each track with 30 components trained using the Expectation-Maximization (EM) algorithm as in [18]. The distances between the GMM's were computed by using the Earth Movers' Distance [19], yielding the audio-track (content) similarities.

3.3. Hypergraph construction

The vertex set of the hypergraph can be defined as $V = U \cup G \cup Ta \cup Tr$. The incidence matrix of the unified hypergraph $\mathbf{H}$ has a size of 4875×147296 elements and its structure is shown in Table 2. In particular:

- $E^{(1)}$: This hyperedge represents a pairwise friendship relation between users and its weight value is set to 1. The incidence matrix of the hypergraph $UE^{(1)}$ has a size of 1389×12296 elements.
- $E^{(2)}$: This hyperedge represents a group of users and it contains all the vertices of the users participating into the group, as well as the vertex corresponding to this group. Its weight value is also set to 1. The incidence matrix of the hypergraph $UE^{(2)} - GrE^{(2)}$ has a size of 1399×10 elements.
- $E^{(3)}$: This 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:

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

  and further scaled as $w(e_{ij}^{(3)})^* = \frac{w(e_{ij}^{(3)})'}{ave(w(e_{ij}^{(3)})')}$, where $ave(w(e_{ij}^{(3)'}))$ is the average of the normalized weights for the particular user u_i. The incidence matrix of the hypergraph $UE^{(3)} - T_rE^{(3)}$ has a size of 3154×68774 elements.
- $E^{(4)}$: This hyperedge contains three vertices: a user, a tag and a music track, representing a tagging relation. Its weight is set to 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. Its weight $w(e_{ij}^{(5)})$ is the similarity between tracks tr_i and tr_j, normalized as follows to eliminate the bias: $w(e_{ij}^{(5)})' = \frac{w(e_{ij}^{(5)})}{max(w(e^{(5)}))}$. Furthermore, a tuning parameter c is included in this point, in order to adjust the importance between audio content similarity and social media information. The final weight is $w(e_{ij}^{(5)}) = cw(e_{ij}^{(5)})'$. The incidence matrix of the hypergraph $TrE^{(5)}$ was computed by selecting only the K most similar tracks (nearest neighbors) with respect to the content similarity as it is described in Section 3.2. Considering our dataset size and structure we set K to 10 and the size of $TrE^{(5)}$ is 1765×17650.

Having constructed $\mathbf{H}$, $\mathbf{W}$,$\mathbf{D}_u$,$\mathbf{D}_e$ and $\mathbf{A}$ can be computed as described in Section 2. At this point, it has to be mentioned that each element $A(u, v)$ of $\mathbf{A}$ is a relatedness measure between objects u and v. The ranking vector $\mathbf{f}^*$ is derived by solving either (2) or (5). The query vector $\mathbf{y}$ is initialized by setting the entry corresponding to the target user u to 1 and all others objects connected to the specific user (Gr, Ta, Tr), to $A(u, v)$. The query vector $\mathbf{y}$ has a length of 4875 elements.

Table 2. 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
Gr	0	$GrE^{(2)}$	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 the same size and structure with the query vector $\mathbf{y}$. The values corresponding to tags are used for personalized tag recommendation with the top ranked tags for a certain track that is left out being recommended to the user.

3.4. Experiments

Let us refer by Music Recommendation on Hypergraph (MRH) to the ranking obtained by (3). The ranking obtained by solving (5) is denoted as Query Group Sparse Optimization (QGSO). The averaged Recall-Precision and the F_1 measure are used as figures of merit. Precision is defined as the number of correctly recommended tags divided by the number of all recommended tags. Recall is defined as the number of correctly recommended tags divided by the number of all tags the user actually used for a track. The F_1 measure is the weighted harmonic mean of precision and recall, measuring the effectiveness of recommendation, when treating precision and recall as equally important.

To assess the tagging performance, the "leave-one-out" (LOO) scheme has been used. Each music track listened and tagged by each user has been left out from training in turn and the 10 top ranked tags are being recommended for the track left out and this particular user. In Fig. 1, the Averaged Recall-Precision curves are plotted by averaging the Recall-Precision curves over 500 randomly selected users with high tagging activity for both the MRH and the QGSO.

The MRH algorithm models the high-order relations between U, Gr, Ta, Tr and thus achieves satisfactory results. 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. The QGSO inherits the advantages of the MRH, but it exploits also the group structure of the hypergraph by assigning unique weights γ_s to each object group (U, Gr, Ta, Tr).

As it is reflected in Fig. 1, by solving the ranking problem and enforcing group sparsity, better results are obtained than the MRH. This way, we exploit the group structure of the hypergraph enhancing the accuracy of our recommendation method. The weights for the 4 different object groups (U, Gr, Ta, Tr) were set 0.9, 0.2, 0.9, 0.6 respectively. This choice was made empirically after examining carefully the results of several experiments.

QGSO yields a slightly higher average precision for average recall rate than the MRH. This finding is also supported by studying the F_1-measure for various ranking positions summarized in Table 3.

4. CONCLUSIONS AND FUTURE WORK

Personalized tag recommendation was addressed as a ranking problem on a hypergraph and solved the ranking problem by enforcing either ℓ_2 norm regularization or group sparsity. The experimental

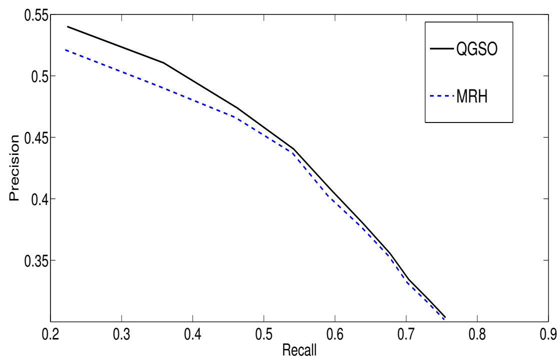

Fig. 1. Averaged Recall-Precision curves for the QGSO and the MRH.

Table 3. Averaged F_1 measures for the QGSO and the MRH at ranking positions 1, 2, 4, ,6, 8 and 10.

	$F_1@1$	$F_1@2$	$F_1@4$	$F_1@6$	$F_1@8$	$F_1@10$
QGSO	0.316	0.422	0.486	0.477	0.453	0.433
MRH	0.310	0.410	0.483	0.473	0.450	0.430

results indicated that by using the unified hypergraph, a promising tag recommendation accuracy has been achieved that can be further improved thanks to the group structure of the hypergraph.

By using non-overlapping object groups, we assume that each object group affects the recommendation process separately. However, certain groups contain mutual and highly correlated information, therefore overlapping groups could be exploited to further improve the recommendation results in the future. Finally, this approach can be possibly extended by automatically calculating the group-weights, based on an optimization scheme.

5. ACKNOWLEDGMENTS

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] M. Slaney, “Web-scale multimedia analysis: Does content matter?,” *IEEE Multimedia*, vol. 11, no. 2, pp. 12–15, April-June 2011.

[2] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere, “Autotagger: A model for predicting social tags from acoustic features on large music databases,” *Journal of New Music Research*, vol. 37, no. 2, pp. 115–135, 2008.

[3] S. R. Ness, A. Theocharis, G. Tzanetakis, and L. G. Martins, “Improving automatic music tag annotation using stacked generalization of probabilistic SVM outputs,” in *Proc. 17th ACM Int. Conf. Multimedia*, Beijing, China, 2009, pp. 705–708.

[4] Y. Panagakis, C. Kotropoulos, and G. R. Arce, “Sparse multi-label linear embedding within nonnegative tensor factorization applied to music tagging,” in *Proc. 11th Int. Conf. Music Information Retrieval*, Utrecht, The Netherlands, 2010, pp. 393–398.

[5] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, “Semantic annotation and retrieval of music and sound effects,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 467–476, 2008.

[6] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, “Tag recommendation based on tensor dimensionality reduction,” in *Proc. 2nd ACM Conf. Recommender Systems*, Lausanne, Switzerland, 2008, pp. 43–50.

[7] S. Rendle, B. L. Marinho, A. Nanopoulos, and L. Schmidt-Thieme, “Learning optimal ranking with tensor factorization for tag recommendation,” in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 727–736.

[8] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang, “Personalized tag recommendation using graph-based ranking on multi-type interrelated objects,” in *Proc. 32nd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, New York, NY, USA, 2009, pp. 540–547.

[9] 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.

[10] A. Theodoridis, C. Kotropoulos, and Y. Panagakis, “Music recommendation using hypergraphs and group sparsity,” in *Proc. IEEE Int. Conf. Audio, Speech, and Signal Processing*, 2013, pp. 56–60.

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

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

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

[14] 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.

[15] Z. Qin and K. Scheinberg, “Efficient block-coordinate descent algorithms for the group lasso,” *Industrial Engineering*, pp. 1–21, 2010.

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

[17] 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.

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

[19] 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.