# MusicBox: Personalized Music Recommendation based on Cubic Analysis of Social Tags

Alexandros Nanopoulos, Dimitrios Rafailidis, Panagiotis Symeonidis and Yannis Manolopoulos

***Abstract*—Social tagging is becoming increasingly popular in music information retrieval (MIR). It allows users to tag music items like songs, albums, or artists. Social tags are valuable to MIR, because they comprise a multifaced source of information about genre, style, mood, users' opinion, or instrumentation. In this paper, we examine the problem of personalized music recommendation based on social tags. We propose the modeling of social tagging data with 3-order tensors, which capture cubic (3-way) correlations between users-tags-music items. The discovery of latent structure in this model is performed with the Higher Order Singular Value Decomposition (HOSVD), which helps to provide accurate and personalized recommendations, i.e., adapted to the particular users' preferences. To address the sparsity that incurs in social tagging data and further improve the quality of recommendation, we propose to enhance the model with a tag-propagation scheme that uses similarity values computed between the music items based on audio features. As a result, the proposed model effectively combines both information about social tags and audio features. The performance of the proposed method is examined experimentally with real data from Last.fm. Our results indicate the superiority of the proposed approach compared to existing methods that suppress the cubic relationships that are inherent in social tagging data. Additionally, our results suggest that the combination of social tagging data with audio features is preferable than the sole use of the former.**

***Index Terms*—Social tags, Audio similarity, Music Recommendation, Tensors, HOSVD.**

## I. Introduction

Social tagging is the process through which users apply free text metadata (tags) to annotate items and to facilitate their retrieval. Social tagging is gaining increasing popularity in music information retrieval (MIR). Web sites like Last.fm, MyStrands, and Qloud, allow users to tag music items such as songs, albums, or artists. The power of social tags lies on the fact that they are shared among users. Social tags provide information about various features that are desirable in MIR, like the genre, style, mood, users' opinion, or instrumentation. Thus, conversely to a single piece of information, like the genre assigned from a taxonomy, social tags comprise a multifaced source of information about music content [7].

Social tags can be used in several ways. The most widely applied way is to consider them as query terms, e.g., find all songs tagged as "classic". However, there are some important challenges posed by the free nature of social tags, which impact their direct usage as query terms:

- The first challenge is that tags may have more than one meaning, a problem called *polysemy*. For instance, several users may refer to music composed in the 18th century with the tag "classic", but the tag "classic" may also be assigned by other users to rock songs from the 60's. Thus, when applying "classic" as a query term, users may unfortunately retrieve a mixture of music pieces from the aforementioned categories.
- A second challenge is the existence of different tags with similar meaning, a problem called *synonymy*. As a result, related music items may not be retrieved together, because they do not share any tags. For instance, some music pieces composed in 18th century may be tagged as "orchestral" and not be retrieved together with others tagged as "classic".
- A third challenge, not related to the meaning of tags, is that most of the music items are tagged poorly, a problem called *sparsity* (or *cold-start*). For instance, a new song is not being as frequently tagged as one that topped the charts. Therefore, users face difficulties when trying to retrieve less frequently tagged content.

The widely used technique of latent semantic analysis (LSA) [4] has recently been applied as a MIR method for addressing the problems of synonymy, polysemy, and noise in social tags [8]. LSA reveals latent structures in the data by using techniques like the Singular Value Decomposition (SVD). Therefore, similar music items can be retrieved and recommended to users based on their representation in the resulting latent space, even if they do not share any tags (not even the query tag)[1]. On the other hand, the problem of sparsity has been investigated in a number of ways. Tagging games, like Tag a Tune (www.gwap.com/gwap/gamesPreview/tagatune), Major Miner (majorminer.com), Listen Game (www.listengame.org), or Herd It (apps.facebook.com/herd-it) collect tags with human players that try to guess the tags of other players. Despite of their potential, music tagging games have not reached the scale of social tagging systems (like Last.fm) [7]. For this reason, research has been conducted to employ features extracted from audio to directly fight sparsity in social tagging systems. A recent experimental evaluation demonstrated that content-

The first author gratefully acknowledges the partial co-funding of his work through the European Commission FP7 project MyMedia (www.mymediaproject.org) under the grant agreement no. 215006.

A. Nanopoulos is with the Institute of Computer Science, University of Hildesheim, Germany, e-mail: nanopoulos@ismll.de

D. Rafailidis, P. Symeonidis, Y. Manolopoulos are with the Department of Informatics, Aristotle University of Thessaloniki, Greece. e-mail {draf,symeon,manolopo}@csd.auth.gr

Manuscript received; revised.

[1] As the results may not share the query tag, in the sequel we more generally refer to the problem as the recommendation of music items, because the retrieved results may contain items not explicitly requested.

based similarity can help to propose labels to yet unlabeled songs [12]. Autotagging [3] represents another approach that uses automated content analysis to predict social tags directly from audio.

### A. Background and Motivation

Approaches based on LSA [8], compute the SVD of a two dimensional matrix that represents 2-way relationships between music items and tags. Although, as previously described, this approach addresses synonymy and polysemy, it suppresses the 3-way (cubic) relationships originally contained in the social tagging data, i.e., between users-items-tags, to just 2-way relationships, i.e., between items-tags. However, music is an artistic concept and music items have a rich and complex view, which is only partially perceived by particular users, depending on their emotional and cultural perspective on music. Social tags are a powerful mechanism that project for each user his perception about a particular music item. For instance, when only items-tags relationships are considered, assume that a male user $U_1$ is fond of young female singers and has tagged Christina Aguilera as "sexy" and Beyonce as "sensual", whereas a female user $U_2$ likes male singers and has tagged Lenny Kravitz as "sexy". When intending to listen to "sexy" artists, the recommendation results for $U_1$ will include both Aguielera and Beyonce (i.e., synonymy between "sensual" and "sexy" is addressed), but Kravitz as well, since the personalized aspect of the tag "sexy" is ignored.

To overcome this problem, we have recently proposed [13] to extent LSA towards the consideration of cubic relationships between users-items-tags. This was attained by generalizing the SVD method to higher dimensions through the Higher-Order SVD (HOSVD), a technique that finds increasingly more applications in various scientific fields [6]. Based on HOSVD, substantially better personalized recommendations can be provided. Nevertheless, the method presented in our preliminary work did not address the problem of sparsity. Along the lines of [3], [12], sparsity in social tagging systems has to be confronted with features extracted from audio. However, the latter two works focused on automatic tag prediction (used, e.g., for tag autocompletion) and not on personalized recommendation of music items.

### B. Contribution and Layout

Here, we propose a method based on HOSVD to extend our previous work on recommending music items, by combining similarities extracted from audio features with social tags. In particular, we opt to address all described problems: (i) synonymy and polysemy, as HOSVD is effectively an extended LSA, (ii) consideration of the personalized aspect of tags, as HOSVD is able to reveal 3-way latent correlations, and (iii) sparsity, as similarities from audio features can account for the absence of social tags. After summarizing related work (Section II), we describe how the proposed method models the social tagging data with a 3-order *tensor* (three dimensional matrix), for which we briefly describe the necessary background information (Section III). Next (Section IV), we present how the proposed method applies HOSVD to recommendation. To ease comprehension, the proposed method is outlined with a motivating example. Since the model, i.e., the 3-order tensor, is highly sparse, we subsequently (Section V) describe a method to exploit audio features and reduce sparsity. Our experimental results on real-world social tagging data from Last.fm (Section VI) provide evidence about the effectiveness of the proposed method and how it addresses all the aforementioned problems. Finally, we furnish the basic conclusions of our study (Section VII).

## II. Related work

Music recommendation has been addressed in various works. In an early attempt, Logan [10] proposed a music recommendation method based solely on using acoustic-based similarity measure. Other approaches try to bridge the semantic gap and to employ hybrid music recommendation methods. Yoshii et al. [14] model collaborative filtering (CF) data and audio-content data together, and unobservable user preferences are statistically estimated. Li et al. [9] employ a probabilistic model estimation for CF. Celma [1] mines music information from the Web (album releases, MP3 blogs, etc.) and combines it with user profiling and audio-content descriptions. However, these existing works are different from our proposed method in the fact that they do not exploit social tags, whose large potential for MIR has been only recently recognized [7], [8].

The application of LSA for discovering the latent semantic structure in an items-tags space has been proposed by Levy and Sandler [8]. As described in Section I-A, methods based on LSA suppress the 3-way users-items-tags relationships and, thus, do not consider the personalized aspect of tags. This problem has been recently examined in [13]. However, differently from this preliminary work, our proposed method exploits not only social tags, but also features extracted from the audio to address the problem of sparsity.

An innovative combination of social tags and audio features has been proposed by Eck et al. [3], where new tags are predicted, while using audio features extracted from music and supervised learning. This method focuses on the task of tag autocompletion, by allowing the system to suggest tags to users, opting to a quick converge on a stable set of tags, whereas we focus on the problem of recommending music items, not tags. Sordo et al. [12] propose a way to annotate music collections by exploiting audio similarity as a way to propagate labels to untagged songs. We use a similar approach, but we follow a different technique and for a different purpose. We focus on how to propagate tags within the tensor model of the social tagging data in order to help HOSVD to discover better latent structure and to improve the quality of personalized recommendation of music items.

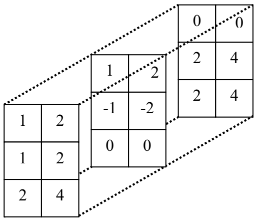
## III. Tensors and HOSVD

This section provides a concise introduction to the topic of tensors and their decomposition. A *tensor* is a multidimensional matrix. An $N$-order tensor $\mathcal{A}$ is denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \ldots I_N}$, with elements $a_{i_1,\ldots,i_N}$. In this paper, for the purposes of the proposed approach, only 3-order tensors are used. In the following, tensors are denoted by calligraphic uppercase letters (e.g., $\mathcal{A}, \mathcal{B}$), matrices by uppercase letters (e.g., $A, B$), and scalars by lowercase letters (e.g., $a, b$).

HOSVD generalizes SVD to multi-dimensional matrices [2]. To apply HOSVD on a 3-order tensor $\mathcal{A}$ we need the definition of the following three *matrix unfoldings*:

$$A_1 \in \mathbb{R}^{I_1 \times I_2 I_3}, \qquad A_2 \in \mathbb{R}^{I_2 \times I_1 I_3}, \qquad A_3 \in \mathbb{R}^{I_1 I_2 \times I_3}$$

Each $A_n,\ 1 \leq n \leq 3$, is called the $n$-mode matrix unfolding of $\mathcal{A}$ and is computed by arranging the corresponding fibers of $\mathcal{A}$ as columns of $A_n$. The left part of Figure 1 depicts an example tensor, whereas the right one shows the 1-mode matrix unfolding $A_1 \in \mathbb{R}^{I_1 \times I_2 I_3}$, where the columns (1-mode fibers) of $\mathcal{A}$ are being arranged as columns of $A_1$.



$$\begin{pmatrix} 1 & 2 & 1 & 2 & 0 & 0 \\ 1 & 2 & -1 & -2 & 2 & 4 \\ 2 & 4 & 0 & 0 & 2 & 4 \end{pmatrix}$$

Fig. 1. An example tensor $\mathcal{A}$ and its 1-mode matrix unfolding $A_1$.

Next, the $n$-mode product of an $N$-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ by a matrix $U \in \mathbb{R}^{J_n \times I_n}$ is defined, which is denoted as $\mathcal{A} \times_n U$. The result of the $n$-mode product is an $(I_1 \times I_2 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N)$-tensor, the entries of which are defined as follows:

$$(\mathcal{A} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} u_{j_n i_n} \qquad (1)$$

Since the focus is on 3-order tensors, $n \in \{1, 2, 3\}$, only 1-mode, 2-mode, and 3-mode products are being used. The HOSVD of a 3-order tensor $\mathcal{A}$ can be written as [2]:

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \qquad (2)$$

where $U^{(1)}, U^{(2)}, U^{(3)}$ contain the orthonormal vectors (called the 1-mode, 2-mode and 3-mode singular vectors, respectively) spanning the column space of the $A_1, A_2, A_3$ matrix unfoldings. $\mathcal{S}$ is the core tensor and has the property of all orthogonality. Figure 2 illustrates the result of HOSVD.

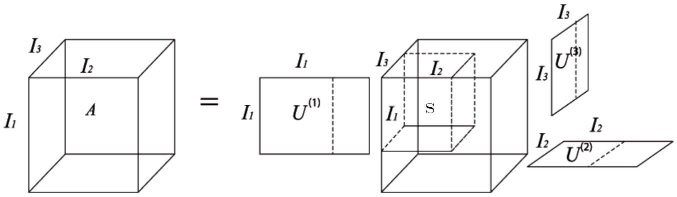


Fig. 2. Visualization of the result of HOSVD.

## IV. Recommendation based on HOSVD

First, we outline the proposed method, which bases the recommendation of music items on HOSVD, through a motivating example. Next, we analyze the steps of the proposed method. To ease the presentation, this section bases its discussion only on social tags. The combination with audio features is separately described in the following section.

### A. Outline

The data collection accumulated by the social tagging system is called *usage data*, which can be represented by a set of triplets $\langle u, t, i \rangle$. Each triple denotes that $u$ user labeled with the $t$ tag the $i$ item.

To outline how the proposed approach works, we use as running example the usage data illustrated in Figure 3, where 4 users tagged 4 different music items (artists). In this figure, the arrow lines and the numbers placed on top of them give the correspondence between the three types of entities. For example, user $U_1$ tagged Beyonce (item as $I_1$) as "sensual" (tag $T_1$). From Figure 3, we can see that users $U_1$ and $U_2$ have a common interest about female singers, while users $U_3$ and $U_4$ have a common interests about male singers.

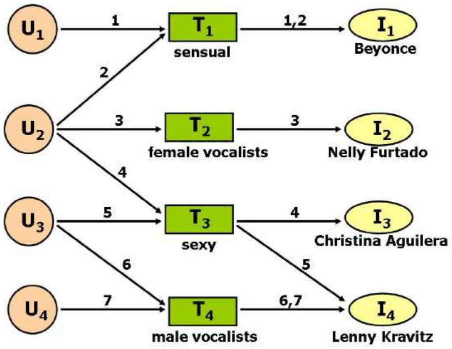


Fig. 3. Usage data of the example.

A 3-order tensor $\mathcal{A} \in \mathbb{R}^{4 \times 4 \times 4}$ can be constructed from these usage data, whose elements are given in Table I. Along with each element we associate a weight, initially set to 1 (the role of this weight is explained in the end of this section).

| Arrow Line | User | Tag | Item | Weight |
|---|---|---|---|---|
| 1 | $U_1$ | $T_1$ | $I_1$ | 1 |
| 2 | $U_2$ | $T_1$ | $I_1$ | 1 |
| 3 | $U_2$ | $T_2$ | $I_2$ | 1 |
| 4 | $U_2$ | $T_3$ | $I_3$ | 1 |
| 5 | $U_3$ | $T_3$ | $I_4$ | 1 |
| 6 | $U_3$ | $T_4$ | $I_4$ | 1 |
| 7 | $U_4$ | $T_4$ | $I_4$ | 1 |

TABLE I
The elements of the tensor from the example in Figure 3.

The proposed method applies HOSVD on the 3-order tensor $\mathcal{A}$ constructed from these usage data. Similarly to LSA for two dimensional matrices, we maintain only a number of the original dimensions in each of the three modes (this procedure is detailed in Section IV-B). This way, we get a reconstructed tensor $\hat{\mathcal{A}}$, which approximates $\mathcal{A}$. $\hat{\mathcal{A}}$ reveals the latent structure in $\mathcal{A}$ and contains a reduced amount of noise compared to $\mathcal{A}$. Table II gives the elements of the reconstructed tensor for the tensor of Table I. The graphical form of $\hat{\mathcal{A}}$ is depicted in Figure 4.

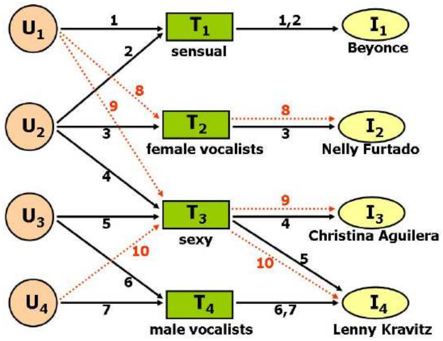


Fig. 4. Graphical form of the reconstructed tensor for the example.

| Arrow Line | User | Tag | Item | Weight |
|---|---|---|---|---|
| 1 | $U_1$ | $T_1$ | $I_1$ | 0.50 |
| 2 | $U_2$ | $T_1$ | $I_1$ | 1.20 |
| 3 | $U_2$ | $T_2$ | $I_2$ | 0.85 |
| 4 | $U_2$ | $T_3$ | $I_3$ | 0.85 |
| 5 | $U_3$ | $T_3$ | $I_4$ | 0.72 |
| 6 | $U_3$ | $T_4$ | $I_4$ | 1.17 |
| 7 | $U_4$ | $T_4$ | $I_4$ | 0.72 |
| **8** | $\mathbf{U_1}$ | $\mathbf{T_2}$ | $\mathbf{I_2}$ | **0.35** |
| **9** | $\mathbf{U_1}$ | $\mathbf{T_3}$ | $\mathbf{I_3}$ | **0.35** |
| **10** | $\mathbf{U_4}$ | $\mathbf{T_3}$ | $\mathbf{I_4}$ | **0.44** |

TABLE II
THE ELEMENTS OF THE RECONSTRUCTED TENSOR FROM THE EXAMPLE IN FIGURE 3.

The reconstructed tensor $\hat{\mathcal{A}}$ additionally contains latent associations discovered among the involved entities, which are typed with boldface in the last three rows of Table II (the corresponding arrows in Figure 4 are depicted with dotted lines). Each element of $\hat{\mathcal{A}}$ is represented as a quadruplet $\langle u, t, i, w \rangle$. The weight $w$ of each quadruplet corresponds to the likeness that user $u$ will tag with $t$ the $i$ item. Please notice that the reconstructed tensor $\hat{\mathcal{A}}$ has modified weights compared to the original tensor $\mathcal{A}$. Therefore, we can use the reconstructed tensor to reveal latent associations and recommend to a user $u$ items for a query tag $t$ according to the weights in the quadruplets that contain $u$ and $t$.

In our example, assume that both users $U_1$ and $U_4$ provide as query term the tag "sexy" ($T_3$). From the reconstructed tensor and the quadruplet $\langle U_1, T_3, I_3, 0.35 \rangle$, we can recommend Christina Aguilera to $U_1$, whereas from the quadruplet $\langle U_4, T_3, I_4, 0.44 \rangle$, we can recommend Lenny Kravitz to $U_4$. It is worth mentioning that neither $U_1$ nor $U_4$ have originally tagged any artist as "sexy". Thus, similarly to LSA [8], the latent relationships provide to $U_1$ and $U_4$ recommendations for this tag. Conversely to LSA [8], however, the consideration by HOSVD of all 3 modes (users-items-tags) helps to provide personalized recommendations to different users for the same query.

### B. Algorithm

In this section, we elaborate on the algorithm that applies HOSVD on tensors and recommends musical items according to the detected latent associations in the reconstructed tensor. The procedure can be decomposed in 6 steps presented as follows.

*1) The initial construction of tensor $\mathcal{A}$:* Based on the usage data, we construct an initial 3-order tensor $\mathcal{A} \in \mathbb{R}^{I_u \times I_t \times I_i}$, where $I_u, I_t, I_i$ are the numbers of users, tags and items, respectively. The initial weight assigned to each entry of $\mathcal{A}$ is equal to 1.

*2) Matrix unfolding of tensor $\mathcal{A}$:* As described in Section III, a tensor $\mathcal{A}$ can be unfolded i.e., transformed to a two dimensional matrix. In our approach, the initial tensor $\mathcal{A}$ is unfolded to all its three modes. Thus, after the unfolding of tensor $\mathcal{A}$, we create 3 new matrices $A_1, A_2, A_3$, as follows:

$$A_1 \in \mathbb{R}^{I_u \times I_t I_i}, \qquad A_2 \in \mathbb{R}^{I_t \times I_u I_i}, \qquad A_3 \in \mathbb{R}^{I_u I_t \times I_i}$$

*3) Application of SVD on each unfolded matrix:* Next, SVD is applied on the three matrix unfoldings $A_n$ ($1 \leq n \leq 3$), resulting to the following decomposition:

$$A_n = U^{(n)} \cdot \Sigma^{(n)} \cdot (V^{(n)})^T, \quad 1 \leq n \leq 3 \tag{3}$$

To reveal latent associations and reduce noise, the dimensionality of each array containing the left-singular vectors (i.e., matrices $U^{(1)}, U^{(2)}, U^{(3)}$) has to be reduced. Therefore, we maintain the dominant $c_n$ left singular vectors in each $U^{(n)}$, $1 \leq n \leq 3$ matrix (as will be shortly explained, only $U^{(n)}$ matrices are used in the following) based on the corresponding singular values in $\Sigma^{(n)}$. The resulting matrix is denoted as $U_{c_n}^{(n)}$. The value of $c_n$ parameters are usually chosen by preserving a percentage of information of the original in $\Sigma^{(n)}$. In our experiments this percentage was set to 60%, because we found that higher values increase the computation time without paying-off in terms of the accuracy of prediction.

*4) The core tensor $\mathcal{S}$ construction:* The core tensor S (see Equation 2) governs the interactions among the three examined modes. Its construction is implemented as follows:

$$\mathcal{S} = \mathcal{A} \times_1 \left(U_{c_1}^{(1)}\right)^T \times_2 \left(U_{c_2}^{(2)}\right)^T \times_3 \left(U_{c_3}^{(3)}\right)^T, \tag{4}$$

where $\mathcal{A}$ is the initial tensor and $\left(U_{c_n}^{(n)}\right)^T$ is the transpose of $U_{c_n}^{(n)}$. Notice that $\mathcal{S}$ is a $c_1 \times c_2 \times c_3$ tensor.

*5) The reconstructed tensor $\hat{\mathcal{A}}$:* Finally, the reconstructed tensor $\hat{\mathcal{A}}$ is computed by:

$$\hat{\mathcal{A}} = \mathcal{S} \times_1 U_{c_1}^{(1)} \times_2 U_{c_2}^{(2)} \times_3 U_{c_3}^{(3)} \tag{5}$$

where $\hat{\mathcal{A}}$ is a tensor with the same size as $\mathcal{A}$. $\hat{\mathcal{A}}$ is a good approximation of $\mathcal{A}$, in the sense that the Frobenius norm $||\mathcal{A} - \hat{\mathcal{A}}||_F^2$ (element-wise squared differences) is small [2]. Moreover, as described, $\hat{\mathcal{A}}$ contains less noise and contains additional, latent associations, resulting from keeping only a subset of the dominant left singular vectors in Step 3.

*6) The generation of the item recommendations:* The elements of the reconstructed tensor $\hat{\mathcal{A}}$ are represented as quadruplets $\langle u, t, i, w \rangle$, where the $w$ weight corresponds to the likeliness that user $u$ will tag item $i$ with tag $t$. If $N$ items have to be recommended to $u$ that queried with tag $t$, then (as exemplified in Section IV-A) the $N$ items are selected that have the highest weights from quadruplets that contain both $u$ and $t$.

## V. Exploiting Audio Similarity

Sparsity occurs in social tagging data, because users tend to provide a relatively small number of tags and, moreover, they tend to tag only a small subset of popular music items. As a consequence, when using a 3-order tensor to model the social tagging data, the resulting tensor can be very sparse. Sparsity affects negatively the HOSVD and the discovery of latent structure. Thus, the quality of the resulting recommendations reduces.

An effective way to address sparsity is to exploit an audio-based similarity measure, so that untagged items can inherit social tags from tagged items to which they are acoustically similar. This process is based on the assumption that similar sounding music will have similar tags [7]. Similarly to [12], the audio-based similarity can be considered as a "black box" (the computed audio similarities are normalized in the range 0–1). In our study, we focused on songs as music items and we computed the audio similarity $\mathrm{Sim}(i, j)$ between any two songs $i, j$ using the G1C algorithm[2]. G1C first applies 1-Gaussian Mixture Model (GMM), based on the Mel Frequency Cepstrum Coefficients (MFCC) and then it applies KL-divergence. The number of MFCC used in our experiments was 20. This similarity measure exploits mostly timbral and rhythmical features, but it does not take into account other mid-level features such as tonality.

The propagation of tags is performed as follows. For each user-tag $(u, t)$ pair in the original data, $S_1$ denotes the set of songs that have been tagged by user $u$ with the $t$ tag, and $S_2$ the set of songs that have not been tagged by user $u$ with the $t$ tag. Then, for each $s \in S_2$, a weight $w$ is measured as follows: $w = max_{\forall p \in S_1} \mathrm{Sim}(s, p)$. Given a parameter $a$, if $w \geq a$, then we assume that $u$ has tagged $s$ as $t$. We consider $w$ as a weight that represents the likelihood of this labeling. This process results to new quadruplets of the form $\langle u, t, s, w \rangle$, which are added to the original data (recall that each quadruplet in the original data has weight equal to 1).

As an example of the tag-propagation process, consider the data in Table I. Assume that the similarities between the 4 items are given in Table III (due to symmetry, only the upper diagonal is given). For the pair $(U_1, T_1)$, it follows that $S_1 = \{I_1\}$ (items tagged by $U_1$ with $T_1$), whereas $S_2 = \{I_2, I_3, I_4\}$ (items not tagged by $U_1$ with $T_1$). The computed weights for the items of $S_2$ are 0.3, 0.6, and 0.2, respectively. Assuming that $a = 0.5$, then an additional quadruplet $\langle U_1, T_1, I_3, 0.6 \rangle$ can be inserted in the tensor, i.e., for the $(U_1, T_1)$ pair, the $T_1$ tag is propagated to item $I_3$.

|  | I$_2$ | I$_3$ | I$_4$ |
|---|---|---|---|
| I$_1$ | 0.3 | 0.6 | 0.2 |
| I$_2$ |  | 0.5 | 0.1 |
| I$_3$ |  |  | 0.2 |

TABLE III
Example of similarities between the items of Table I.

Thus, the $a$ parameter acts as a threshold to decide whether or not to propagate the tags. With higher values of $a$, tags are propagated with higher confidence, since only very similar songs are taken into account. However, their number may be not adequate to address the sparsity. When $a$ is low, more tags can be propagated, but this time with lower confidence. Thus, the latter case has the danger of including noise that will affect the recommendation result. In our experimental evaluation we demonstrate the impact of $a$ and how tag propagation can help to improve the provided recommendations.

## VI. Experimental Evaluation

We experimentally compare the proposed method, denoted as MusicBox (MB)[3], against the following methods: (i) Recommendation based on HOSVD that does not consider audio features [13]. (ii) Recommendation based on LSA applied to items-tags (2-way) relationships, with recommendations being generated based on the Item-based (IB) algorithm [11]. In simpler terms, LSA combines SVD and vector cosine similarity (in the reduced space). Comparison against HOSVD will indicate how much the consideration of audio features helps to address sparsity, whereas comparison against LSA will indicate how much the consideration of 3-way relationships helps to provide personalized recommendations with higher quality. The experiments were conducted with a 64-bit, with an Intel Xeon CPU at 2 GHz with 4 cores and 8 GB RAM.

### A. Experimental configuration

All examined methods have been implemented in Matlab. For HOSVD and MB, we used the Multislice Projection (MP) toolbox[4] that scales to large tensors that cannot be maintained in memory.

A real data set has been crawled from Last.fm. The data was gathered during June 2008, using Last.fm web services. The musical items correspond to song titles. The preprocessing of tags involved their tokenization with a standard stop-list and filtering by removing tags that are either personal (e.g., "seen live", "I own it") or organizational (e.g., "check out"). The result was 64,025 triplets in the form user–tag–song, with 732 distinct users, 2,527 tags and 991 songs. Please notice that although in this work we considered a simple and not automatic process for discarding some tags, a more systematic approach can be followed in general, using the process proposed by Geleijnse et al. [5].

To examine the combination of audio features with tagging data, the USPOP'02[5] song collection is used, which contains

[2] Implemented in the MA Toolbox`www.ofai.at/∼elias.pampalk/ma`

[3] The name MusicBox stems from the cubic (3-mode) tensor that contains music audio-similarities.

[4] `www.apperceptual.com/multislice`

[5] `labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html`

8,764 tracks from 400 artists. In particular, the crawling of data from Last.fm was designed to detect songs that belong in the USPOP'02 collection. Thus, the crawling procedure identified the songs that both belong to USPOP'02 and have been tagged in Last.fm. Audio features have been computed by keeping 30 sec from the center of each song and extracting 20 MFCC from each frame. Based on the audio features, the G1C algorithm computed the similarity between any two songs.

The following evaluation protocol is performed. Originally we consider the input tensor (no propagated tags). For each user $u$, one of his triplets in this tensor is randomly selected. The set of all selected triplets forms the test data, whereas the remaining triplets form the training data used to build the models. Notice that for MB, the insertion of the additional triplets is performed only to the training data[6].

The objective of the recommendation task was to predict the items in the test data. We used two performance metrics: Recall and Reciprocal Rank (RR), which is defined as the inverse of the correct answer's rank. Recall characterizes the ability of a method to predict relevant items, whereas RR takes also into account the position of predicted relevant items in the recommendation list. We report mean values of Recall and RR (denoted as MRR) for all the test data. Other common measures, like precision or $F_1$, are omitted, because for each user/tag combination in the test data a constant number of items has to be predicted and only a prespecified number ($N$) of recommendations is taken into account. Therefore, for this kind of evaluation protocol, it is redundant to evaluate precision (thus $F_1$ too), because it is just the same as recall up to a multiplicative constant.

Regarding parameters, HOSVD in MusicBox maintains singular vectors by preserving the 60% of information (variance), as described in Section IV. For LSA, we have tried several values for the percentage of singular vectors preserved by SVD and kept the one that resulted to the best results. Moreover, for the IB algorithm used by LSA, the $k$ parameter (i.e., the number of nearest neighbor items) was varied from 10 to 300 by an interval of 10, and we finally kept the value that leaded to the best results ($k = 30$).

### B. Results

First, we examined the impact of $a$ parameter on the performance of MB. Table IV reports the percentage of propagated triplets relative to the original number of triplets (i.e., without propagation) for varying values of $a$.

| $a$ | 0.25 | 0.5 | 0.75 |
|---|---|---|---|
| perc. propagated triplets | 69.17% | 21.47% | 14.3% |

TABLE IV
PERCENTAGE OF PROPAGATED TRIPLETS.

Next, we compared all methods in terms of Recall (Figure 5) and MRR (Figure 6) for varying the number of recommended items $N$. For MB we consider 3 different $a$ values: 0.25, 0.5, 0.75. MB can be considered as a generalization of HOSVD, since by setting $a = 1$ MB reduces to HOSVD.

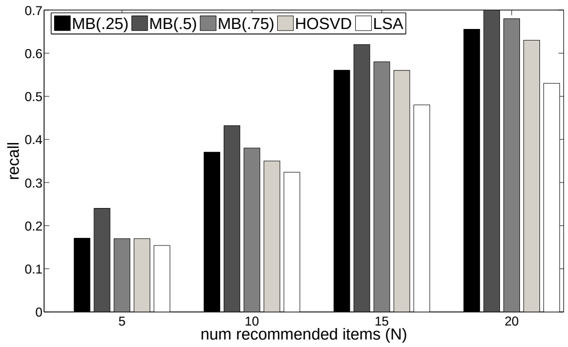


Fig. 5. Results on Recall.

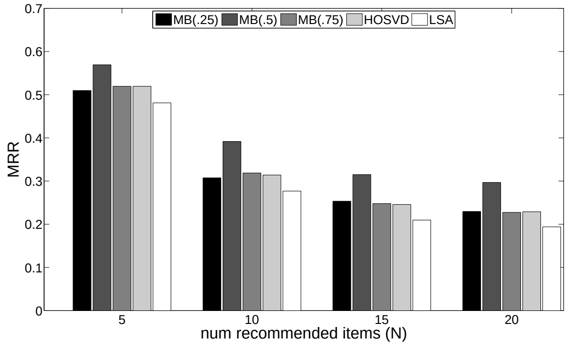


Fig. 6. Results on Mean Reciprocal Rank (MRR).

Along the lines of Section V, lower values of $a$ can incur noise and reduce the quality of recommendation of MB. Conversely, higher values of $a$ do not adequately address sparsity (see also Table IV). Values between the two extremes (like $a = 0.5$) can address sparsity without being significantly affected by noise, and thus compare favorably to HOSVD, since the latter is impacted by sparsity. MB outperforms LSA as well, mainly because it does not suppress the 3-way relationships between users-items-tags, and also because the tag propagation used in MB addresses the sparsity. To further understand the differences between the examined methods, we measured the (Spearman) correlation between the results of MB ($a = 0.5$), HOSDV, and LSA. The correlation between MB and HOSVD is 0.50, between MB and LSA 0.03, and between HOSVD and LSA -0.14. This shows that MB and HOSVD have a partial "agreement", which is expected as MB extends HOSVD, but both methods have different results than LSA.

Finally, we compared MB and LSA in terms of execution times (results for HOSVD are omitted because they are comparable to MB). Both algorithms consist of an offline part to build the model and an online part to generate the recommendations. Table V presents the execution times of both parts in terms of the percentage of preserved information (variance), which mainly affects the offline times. Because MB builds a more complex model than LSA, its offline times are higher.[7] Regard-

[6] The triplets of the test set are excluded from tag propagation.

[7] The presented offline times are for building the model in batch. Both LSA and MB can be fast incrementally updated when new data arrive.

ing online times, which affect users' experience, MB compares favorably against LSA, because MB needs just to sort a vector of predictions to generate recommendations. Conversely, since LSA is based on IB to generate recommendations, during its online part it first requires to find items' neighborhoods, then to compute aggregated frequencies from the neighborhoods, and finally to sort these frequencies.

| info. perc. | 10% | 20% | 30% | 40% | 50% | 60% |
|---|---|---|---|---|---|---|
| MB (offline) | 205.69 | 224.13 | 242.73 | 263.63 | 286.13 | 306.96 |
| LSA (offline) | 1.66 | 6.3 | 16.74 | 32.55 | 58.83 | 83.88 |
| MB (online) | 0.21 | 0.21 | 0.21 | 0.21 | 0.21 | 0.21 |
| LSA (online) | 13.39 | 13.63 | 13.63 | 13.71 | 13.71 | 13.69 |

TABLE V
EXECUTION TIMES (OFFLINE IN SEC AND ONLINE IN MSEC).

### C. Discussion

To summarize the aforementioned experimental results, they clearly indicate that by preserving the 3-way relationships originally existing in social tagging data, the quality of recommendation is improved against methods that suppress them. The reason is that due to the consideration of the 3-way relationships, the resulting recommendations can better match the personalized perspectives of each user. This conclusion is supported by the fact that all tensor-based methods (MB and HOSVD) perform favorably against LSA. Moreover, the proposed tag-propagation method, which exploits audio similarities, is effective in reducing the sparsity that is particularly pronounced in the case of the tensor-based methods.

For all these reasons, the proposed method outperforms all the other examined methods. However, there exist some issues that require further attention, because they can comprise limitations of the proposed approach. The first one is the computation overhead of the proposed method for the off-line part (model computation). Although the proposed method requires small execution times for the online part (that is most important to users), it is also demanding to reduce the requirements for the off-line part as well. We have to mention that we used a multi-sliced tensor decomposition algorithm (see Section VI-A) that reduces memory consumption to permit the use of large tensors, but increases the overall execution time due to the separate processing of several slices. In our future work, we plan to examine more efficient implementations of multi-sliced methods, that will present a balance between memory consumption and increased execution time for the off-line computation.

Another important observation is that the proposed tag-propagation process cannot be uncontrolled. By allowing extensive tag-propagation, the noise that incurs may affect the quality of recommendations. In this work, we have proposed a simple method to control the amount of propagated tags by using the $a$ parameter. In our future work, we plan to investigate further this issue, by examining the impact of different tag categories (like those that are related to time, genre, instrumentation, etc.) and to develop more advanced methods for the detection of noisy tags in order to avoid their propagation.

## VII. CONCLUSIONS

We have examined the problem of personalized music recommendation based on social tags. To capture the 3-way correlations between users-tags-music items, we modeled social tagging data with 3-order tensors. Recommendation is performed based on the discovery of latent structure in this model using HOSVD, which extends SVD to high dimensional matrixes (tensors). Moreover, we further improved the quality of recommendation by addressing the sparsity that incurs in social tagging data, by exploiting similarities between the music items that are computed based on audio features. The performance of the proposed method is examined experimentally with real social tagging data from Last.fm. Our experimental results indicate the superiority of the proposed method in terms of improving the recommendation quality.

As future work, we will extend the proposed method to the important task of using tags to explain the personalized recommendations, as described in [7]. Another point of future work, as mentioned in Section VI-C, is to investigate the automatic categorization of tags and to use this information for filtering the propagation of noisy tags and improving the recommendation quality. Moreover, such categorization can provide further insights about the type of tags that help more the recommendation process. Finally, we will examine additional audio-similarity measures, which will cope with other mid-level audio features, like tonality.

## REFERENCES

[1] O. Celma. Foafing the music: Bridging the semantic gap in music recommendation. In *Proc. 5th ISWC Conf.*, pages 927–934, 2006.

[2] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal of Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[3] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *Proc. 21st NIPS Conf.*, 2007.

[4] G. Furnas, S. Deerwester, and S. Dumais. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proc. 11th ACM SIGIR Conf.*, pages 465–480, 1988.

[5] G. Geleijnse, M. Schedl, and P. Knees. The quest for ground truth in musical artist tagging in the social web era. In *Proc. 8th ISMIR Conf.*, 2007.

[6] T. Kolda and T. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3), September 2009 (to appear).

[7] P. Lamere. Social tagging and music information retrieval. *Journal of New Music Research*, 37(2):101–114, 2008.

[8] M. Levy and M. Sandler. Learning latent semantic models for music from social tags. *Journal of New Music Research*, 37(2):137–150, 2008.

[9] Q. Li, S. Myaeng, D. Guan, and B. Kim. A probabilistic model for music recommendation considering audio features. In *Proc. AIRS Conf.*, pages 72–83, 2005.

[10] B. Logan. Music recommendation from song sets. In *Proc. 5th ISMIR Conf.*, pages 425–428, 2004.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. 10th WWW Conf.*, pages 285–295, 2001.

[12] M. Sordo, C. Laurier, and O. Celma. Annotating music collections: How content-based similarity helps to propagate labels. In *Proc. 8th ISMIR Conf.*, 2007.

[13] P. Symeonidis, M. Ruxanda, A. Nanopoulos, and Y. Manolopoulos. Ternary semantic analysis of social tags for personalized music recommendation. In *Proc. 9th ISMIR Conf.*, pages 219–224, 2008.

[14] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. Okuno. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *Proc. 7th ISMIR Conf.*, pages 296–301, 2006.