TERNARY SEMANTIC ANALYSIS OF SOCIAL TAGS FOR PERSONALIZED MUSIC RECOMMENDATION

Panagiotis Symeonidis[1] **Maria Ruxanda**[2] **Alexandros Nanopoulos**[1] **Yannis Manolopoulos**[1]

1. Department of Informatics
Aristotle Univ. of Thessaloniki, Greece

2. Department of Computer Science
Aalborg University, Denmark

ABSTRACT

Social tagging is the process by which many users add metadata in the form of keywords, to annotate information items. In case of music, the annotated items can be songs, artists, albums. Current music recommenders which employ social tagging to improve the music recommendation, fail to always provide appropriate item recommendations, because: (i) users may have different interests for a musical item, and (ii) musical items may have multiple facets. In this paper, we propose an approach that tackles the problem of the multimodal use of music. We develop a unified framework, represented by a 3-order tensor, to model altogether users, tags, and items. Then, we recommend musical items according to users multimodal perception of music, by performing latent semantic analysis and dimensionality reduction using the Higher Order Singular Value Decomposition technique. We experimentally evaluate the proposed method against two state-of-the-art recommendations algorithms using real Last.fm data. Our results show significant improvements in terms of effectiveness measured through recall/precision.

1 INTRODUCTION

Social tagging is the process by which many users add metadata in the form of keywords to annotate and categorize information items such as songs, pictures, products. In general, social tagging is associated to the "Web 2.0" technologies and has already become an important source of information for recommendation. In the music domain, popular web systems such as Last.fm and MyStrands provide possibility for users to tag with free text labels an item of interest - e.g., artist, song, album. Such systems can further exploit these social tags to improve the search mechanisms and the personalized music recommendation.

Recent research in the music field has also focused on exploiting the social tags in various ways. For example, a partial solution to the cold-start problem of music recommenders has been proposed in [4]: social tags are used for the automatic generation of new tags, which then can be used to label the untagged music. In [9], the social tags are investigated as a source of semantic metadata for music, which can be used to generate a psychologically-motivated search-space representing musical emotion.

However, the social tags carry useful information not only about the musical items they label, but also about the users who tagged. This aspect is not being fully exploited, neither by the music recommenders, neither in the research field. Music is an artistic concept, and the musical items (artists, songs, albums) 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 reveal 3-dimensional correlations between users–tags–items. This triplet information can project for each user his perception of a particular musical item. However, the current music recommender systems are commonly using collaborative filtering techniques, which traditionally exploit only pairs of 2-dimensional data. Thus, they are not capable of capturing well the multimodal use of music.

As a simple example, let us consider the social tagging system of artists in Last.fm. Assume two users. One is very fond of young female singers and therefore has tagged Christina Aguilera as "sexy" and Beyonce as "sensual". Another is fond of male singers and has tagged Lenny Kravitz as "sexy" and "male vocalists". When wanting to listen to "sexy" music, both users are recommended male and female singers, while the first user is expecting female singers and the other prefers the opposite.

Recent research has focused on developing recommendation algorithms [7, 14], which try to exploit tags given by users on specific items. However, the existing algorithms do not consider the 3 dimensions of the problem altogether, and therefore they miss a part of the semantics that is carried by the 3-dimensions.

In this paper, we address the problem of music recommendation by capturing the multimodal perception of music by particular users. We perform 3-dimensional analysis on the social tags data, attempting to discover the latent factors that govern the associations among the triplets user–tag–item. Consequently, the musical items (artists, songs or albums) can be recommended according to the captured associations. That is, given a user and a tag, the purpose is to predict whether and how much the user is likely to label with this tag a specific musical item.

Our strategy in dealing with the 3-dimensional social

THIS RESEARCH WAS SUPPORTED IN PART BY THE DANISH RESEARCH COUNCIL FOR TECHNOLOGY AND PRODUCTION SCIENCES PROJECT 26-04-0092 INTELLIGENT SOUND, AND BY GREEK SECRETARIAT FOR RESEARCH AND TECHNOLOGY ΠΑΒΕΤ (05ΠΑΒ216) PROJECT.

tagging data, is to develop a unified framework to model the three dimensions. Thus, user-tag-item data is represented by a 3-order tensor. Consequently, we have to deal with the data sparsity problem: the three-way data is highly sparse, especially that each user only tags a small number of items. Latent Semantic Indexing (LSI) has been proved useful to address the data sparseness in 2-dimensional data recommender systems, however, it is still an open problem for the 3-dimensional data case. Therefore, we perform 3-mode analysis, using the Higher Order Singular Value Decomposition (HOSVD) technique.

The contributions of our approach are as follow: **(1)** we provide a method to improve music recommendation by capturing users multimodal perception of music; **(2)** we develop a unified framework, represented by a 3-order tensor, to model the three types of entities that exist in social tagging data; **(3)** we apply dimensionality reduction in 3-order tensors to reveal the latent semantic associations between users, tags, and items; **(4)** we perform experimental comparison of the proposed method against two state-of-the-art recommendations algorithms, using Last.fm data; our results show significant improvements in terms of effectiveness measured through recall/precision.

The rest of this paper is organized as follows. Section 2 summarizes the related work, whereas Section 3 briefly reviews background techniques employed in our approach. A motivating example and the proposed approach are described in Section 4. Experimental results are given in Section 5. Finally, Section 6 concludes this paper.

2 RELATED WORK

Music recommendation has been addressed in various work. For example, in Logan [11] music recommendation is done based solely on using acoustic-based similarity measure. Other approaches try to bridge the semantic gap and employ hybrid music recommendation methods. Thus, Yoshii et al. [15] model collaborative filtering (CF) data and audio-content data together, and unobservable user preferences are statistically estimated. Li et al. [10] employ a probabilistic model estimation for CF, where musical items are clustered based on audio-content and user rating, and predictions are made considering the Gaussian distribution of ratings. Celma [2] mines music information from the Web (album releases, MP3 blogs, etc.) and is using it together with user profiling and audio-content descriptions.

The above work can be used to improve the music recommendation by addressing the cold-start problem and the bias of CF towards mainstream music. Along the same lines, an innovative use of social tags has been recently proposed in [4]. Eck et al. [4] predict new tags using audio features extracted from music and supervised learning. These automatically-generated tags resemble the characteristics of those generated by social taggers, and can be used to label new or poorly tagged music.

However, current music recommenders fail to always provide good recommendations, because they do not capture well the interest of particular users in musical items that have multiple facets. Recent research work [4] envisaged that musical items have multiple facets, but it did not address their multimodal perception by particular users. Since social tagging data carry simultaneous information about both the items and the users who tagged them, we propose to use such data as means to improve music recommendation by capturing the multimodal use of music.

The characteristics of social tagging systems have been already studied in the literature. Halpin et al. [6] claimed that there are three main entities in any tagging system: users, items, and tags. In contrast to the above ternary relation, recommender systems apply collaborative filtering on 2-dimensional spaces. For example, the approach of projecting the 3-dimensional space of social tagging into pair relations $\{$user, item$\}$, $\{$user, tag$\}$, $\{$tag, item$\}$, is applied in well-known recommendation algorithms such as Penalty-Reward and FolkRank. The Collaborative Tag Suggestions algorithm [14], also known as Penalty-Reward (PR), uses an authority score for each user, which measures how well each user has tagged in the past. This authority score can be computed via an iterative algorithm such as HITS [8]. FolkRank algorithm [7] is inspired by the seminal PageRank [12] algorithm. The key idea of FolkRank is that an item, which is tagged with important tags by important users, becomes important itself (and the same holds for tags and users).

However, the above state-of-art algorithms miss a part of the total interaction between the three dimensions of the social tagging space. In contrast, our approach develops a unified framework to concurrently model the three dimensions by employing a 3-order tensor, on which latent semantic analysis is performed using HOSVD technique [3]. The HOSVD technique has been successfully applied for computer vision problems. We also use in our approach the work proposed in Wang and Ahuja [13], which present a novel multi-linear algebra-based method to reduce the dimensionality representation of multi-dimensional data.

3 PRELIMINARIES - TENSORS AND HOSVD

In the following, we denote tensors by calligraphic uppercase letters (e.g., $\mathcal{A}$, $\mathcal{B}$), matrices by uppercase letters (e.g., A, B), scalars by lowercase letters (e.g., a, b), and vectors by bold lowercase letters (e.g., $\mathbf{a}$, $\mathbf{b}$).

SVD and Latent Semantic Indexing

The singular value decomposition (SVD) [1] of a matrix $F_{I_1 \times I_2}$ can be written as a product of three matrices, as shown in Equation 1:

$$F_{I_1 \times I_2} = U_{I_1 \times I_1} \cdot S_{I_1 \times I_2} \cdot V^T_{I_2 \times I_2}, \qquad (1)$$

where U is the matrix with the left singular vectors of F, V^T is the transpose of the matrix V with the right singular vectors of F, and S is the diagonal matrix of (ordered) singular values of F.

By preserving only the largest $c < \min\{I_1, I_2\}$ singular values of S, SVD results to matrix $\hat{F}$, which is an approximation of F. In Information Retrieval, this technique is used by Latent Semantic Indexing (LSI) [5], to deal with the latent semantic associations of terms in texts

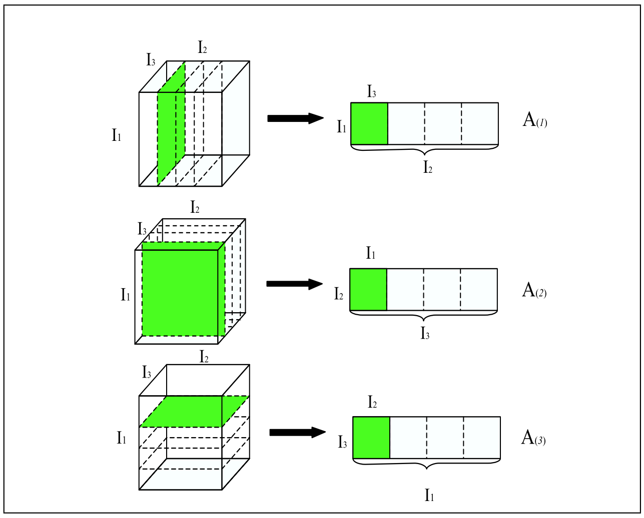

Figure 1. Visualization of the three unfoldings of a 3-order tensor.

and to reveal the major trends in F. The tuning of c is empirically determined by the information percentage that is preserved compared to the original matrix [3].

Tensors

A *tensor* is a multi-dimensional matrix. A N-order tensor $\mathcal{A}$ is denoted as $\mathcal{A} \in R^{I_1 \ldots I_N}$, with elements $a_{i_1,\ldots,i_N}$. In this paper, for the purposes of our approach, we only use 3-order tensors.

HOSVD

The high-order singular value decomposition [3] generalizes the SVD computation to multi-dimensional matrices. To apply HOSVD on a 3-order tensor $\mathcal{A}$, three *matrix unfolding* operations are defined as follows [3]:

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

where A_1, A_2, A_3 are called the 1-mode, 2-mode, 3-mode matrix unfoldings of $\mathcal{A}$, respectively. The unfoldings of $\mathcal{A}$ in the three modes is illustrated in Figure 1.

Example: Define a tensor $\mathcal{A} \in R^{3 \times 2 \times 3}$ by $a_{111} = a_{112} = a_{211} = -a_{212} = 1, a_{213} = a_{311} = a_{313} = a_{121} = a_{122} = a_{221} = -a_{222} = 2, a_{223} = a_{321} = a_{323} = 4, a_{113} = a_{312} = a_{123} = a_{322} = 0$. The tensor and its 1-mode matrix unfolding $A_1 \in R^{I_1 \times I_2 I_3}$ are illustrated in Figure 2.

Next, we define the n-mode product of an N-order tensor $\mathcal{A} \in R^{I_1 \times \ldots \times I_N}$ by a matrix $U \in R^{J_n \times I_n}$, 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:

$$(\mathcal{A} \times_n U)_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} u_{j_n i_n} \qquad (2)$$

Since we focus on 3-order tensors, $n \in \{1, 2, 3\}$, we use 1-mode, 2-mode, and 3-mode products.

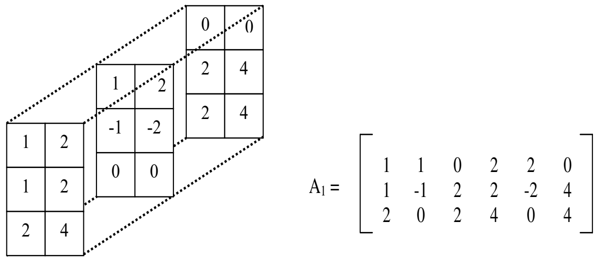

Figure 2. Visualization of tensor $\mathcal{A} \in R^{3 \times 2 \times 3}$ and its 1-mode matrix unfolding.

In terms of n-mode products, SVD on a regular two-dimensional matrix (i.e., 2-order tensor), can be rewritten as follows [3]:

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

where $U^{(1)} = (u_1^{(1)} u_2^{(1)} \ldots u_{I_1}^{(1)})$ is a *unitary* $(I_1 \times I_1)$-matrix, $U^{(2)} = (u_1^{(2)} u_2^{(2)} \ldots u_{I_1}^{(2)})$ is a *unitary* $(I_2 \times I_2)$-matrix, and S is a $(I_1 \times I_2)$-matrix with the properties of: (i) pseudo-diagonality: $S = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min\{I_1, I_2\}})$ (ii) ordering: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min\{I_1, I_2\}} \geq 0$.

By extending this form of SVD, HOSVD of 3-order tensor $\mathcal{A}$ can be written as follows [3]:

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \qquad (4)$$

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. S is the core tensor and has the property of all orthogonality.

4 THE PROPOSED APPROACH

We first provide the outline of our approach, which we name Tensor Reduction, through a motivating example. Next, we analyze the steps of the proposed algorithm.

4.1 Outline

When using a social tagging system, to be able to retrieve information items easily, a user u labels with a tag t an item i. After some time of usage, the tagging system accumulates a collection of data – hence, *usage data*, which can be represented by a set of triplets $\{u, t, i\}$.

Our Tensor Reduction approach applies HOSVD on the 3-order tensor constructed from these usage data. In accordance with the HOSVD technique introduced in Section 3, the Tensor Reduction algorithm receives as input the usage data of $\mathcal{A}$ and outputs the reconstructed tensor $\hat{\mathcal{A}}$. $\hat{\mathcal{A}}$ measures the associations among the users, tags, and items. The elements of $\hat{\mathcal{A}}$ can be represented by a quadruplet $\{u, t, i, p\}$, where p measures the likeliness that user u will label with tag t an item i . Therefore, items can be recommended to u according to their weights associated with $\{u, t\}$ pair.

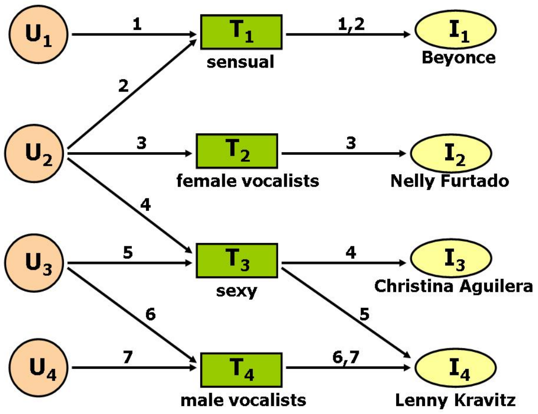

Figure 3. Usage data of the running example

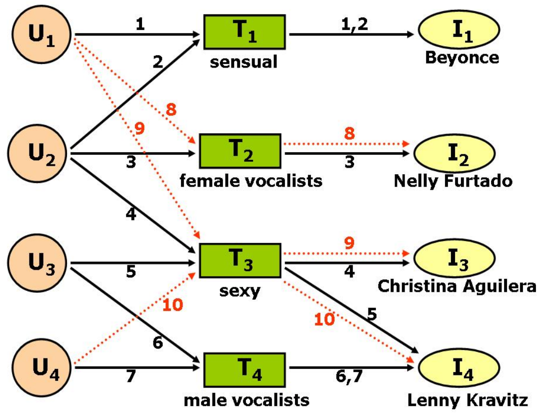

Figure 4. Illustration of the Tensor Reduction Algorithm output for the running example

In order to illustrate how our approach works, we apply the Tensor Reduction algorithm to a running example, which is illustrated in Figure 3. As it can be seen, 4 users tagged 4 different items. In the figure, the arrow lines and the numbers on them give the correspondence between the three types of entities. For example, user U_1 tagged with tag "sensual" (denoted as T_1) the item "Beyonce" (denoted as I_1). From Figure 3, we can see that users U_1 and U_2 have common interests on female singers, while users U_3 and U_4 have common interests in male singers.

A 3-order tensor $\mathcal{A} \in R^{4\times4\times4}$ can be constructed from these usage data. We use the co-occurrence frequency of user, tag and item as the elements of tensor $\mathcal{A}$, which are given in Table 1.

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 1. Tensor Constructed from the usage Data of the running example.

After performing the Tensor Reduction analysis (details are given in the section 4.2), we get the reconstructed tensor of $\hat{\mathcal{A}}$. Table 2 gives the output of the Tensor Reduction algorithm, which is also illustrated in Figure 4.

We can notice that the algorithm outputs new associations among the involved entities (see the last rows in the Table 2 and the dotted lines in Figure 4). Even though in the original data, user U_1 did not tag items I_2 and I_3, the algorithm is capable to infer that if U_1 would tag them, then U_1 would likely (likelihood 0.35) use tags "female vocalists", and respectively "sexy". As well, the algorithm can infer that if U_4 would tag item I_4 with another tag, then U_4 would likely (likelihood 0.44) use the tag "sexy".

We judge the obtained results as reasonable since U_1 appears to be concerned with female singers rather than with male singers, and viceversa for U_4 . That is, the Tensor Reduction approach is able to capture the latent associations among the multi-type data entities: users, tags and musical items. These associations can further be used to improve the recommendation procedure of musical items.

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 2. Tensor Constructed from the usage data of the running example.

4.2 The Tensor Reduction Algorithm

In this section, we elaborate on how HOSVD is applied on tensors and on how the recommendation of musical items is performed according to the detected latent associations.

Our Tensor Reduction algorithm initially constructs a tensor, based on usage data triplets $\{u, t, i\}$ of users, tags and items. The motivation is to use all three entities that interact inside a social tagging system. Consequently, we proceed to the unfolding of $\mathcal{A}$, where we build three new matrices. Then, we apply SVD in each new matrix. Finally, we build the core tensor $\mathcal{S}$ and the resulting tensor $\hat{\mathcal{A}}$. All these can be summarized in 6 steps, as follows.

Step 1. The initial construction of tensor $\mathcal{A}$

From the usage data triplets (user, tag, item), we construct an initial 3-order tensor $\mathcal{A} \in R^{u\times t\times i}$, where u, t, i are the numbers of users, tags and items, respectively. Each tensor element measures the preference of a (user u, tag t) pair on an item i.

Step 2. Matrix unfolding of tensor $\mathcal{A}$

As described in Section 3, a tensor $\mathcal{A}$ can be matricized i.e., to build matrix representations in which all the column (row) vectors are stacked one after the other. In our approach, the initial tensor $\mathcal{A}$ is matricized in all three modes. Thus, after the unfolding of tensor $\mathcal{A}$ for all three modes, we create 3 new matrices A_1, A_2, A_3, as follows:

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

Step 3. Application of SVD in each matrix

We apply SVD on the three matrix unfoldings A_1, A_2, A_3. We result to total 9 new matrices.

$$A_1 = U^{(1)} \cdot S_1 \cdot V_1^T \tag{5}$$

$$A_2 = U^{(2)} \cdot S_2 \cdot V_2^T \tag{6}$$

$$A_3 = U^{(3)} \cdot S_3 \cdot V_3^T \tag{7}$$

For tensor dimensionality reduction, there are three parameters to be determined. The numbers c_1, c_2, and c_3 of left singular vectors of matrices $U^{(1)}$, $U^{(2)}$, $U^{(3)}$ which are retained, are determinative for the final dimension of the core tensor S. Since each of the three diagonal singular matrices S_1, S_2, and S_3 are calculated by applying SVD on matrices A_1, A_2 and A_3 respectively, we use different c_1, c_2, and c_3 values for each matrix $U^{(1)}$, $U^{(2)}$, $U^{(3)}$.

The numbers c_1, c_2, and c_3 are empirically chosen by preserving a percentage of information of the original S_1, S_2, S_3 matrices after appropriate tuning (usually the percentage is set to 50% of the original matrix).

Step 4. The core tensor $\mathcal{S}$ construction

The core tensor S governs the interactions among user, item and tag entities. Since we have selected the dimensions of $U^{(1)}$, $U^{(2)}$, and $U^{(3)}$ matrices, we proceed to the construction of the core tensor S, as follows:

$$\mathcal{S} = \mathcal{A} \times_1 {U_{c_1}}^{(1)^T} \times_2 {U_{c_2}}^{(2)^T} \times_3 {U_{c_3}}^{(3)^T}, \tag{8}$$

where $\mathcal{A}$ is the initial tensor, ${U_{c_1}}^{(1)^T}$ is the transpose of the c_1-dimensionally reduced $U^{(1)}$ matrix, ${U_{c_2}}^{(2)^T}$ is the transpose of the c_2-dimensionally reduced $U^{(2)}$, and ${U_{c_3}}^{(3)^T}$ is the transpose of the c_3-dimensionally reduced $U^{(3)}$.

Step 5. The tensor $\hat{\mathcal{A}}$ construction

Finally, tensor $\hat{\mathcal{A}}$ is build by the product of the core tensor $\mathcal{S}$ and the mode products of the three matrices $U^{(1)}$, $U^{(2)}$ and $U^{(3)}$ as follows:

$$\hat{\mathcal{A}} = \mathcal{S} \times_1 {U_{c_1}}^{(1)} \times_2 {U_{c_2}}^{(2)} \times_3 {U_{c_3}}^{(3)}, \tag{9}$$

where $\mathcal{S}$ is the c_1, c_2, c_3 reduced core tensor, ${U_{c_1}}^{(1)}$ is the c_1-dimensionally reduced $U^{(1)}$ matrix, ${U_{c_2}}^{(2)}$ is the c_2-dimensionally reduced $U^{(2)}$ matrix, ${U_{c_3}}^{(3)}$ is the c_3-dimensionally reduced $U^{(3)}$ matrix.

Step 6. The generation of the item recommendations

The reconstructed tensor $\hat{\mathcal{A}}$ measures the associations among the users, tags and items, so that the elements of $\hat{\mathcal{A}}$ represent a quadruplet $\{u, t, i, p\}$ where p is the likeliness that user u will tag musical item i with tag t. Therefore, musical items can be recommended to u according to their weights associated with $\{u, t\}$ pair.

5 EXPERIMENTAL CONFIGURATION

In this section, we study the performance of our approach against two well-known recommendation algorithms: Collaborative Tag Suggestions [14] (known as Penalty-Reward) and FolkRank [7], denoted as PR and FolkRank, respectively. We denote our algorithm as Tensor Reduction.

5.1 Data Set

To evaluate the aforementioned algorithms we have chosen a real data set mined from Last.fm. The data was gathered during October 2007, by using Last.fm web services. The musical items correspond to artist names, which are already normalized by the system. There are 12,773 triplets in the form user–tag–artist. To these triplets correspond 4,442 users, 2,939 tags and 1,620 artists. (We only kept tags expressing positive preference.)

To get more dense data, we follow the approach of [7] and we adapt the notion of a p-core to tri-partite hypergraphs. The p-core of level k has the property, that each user, tag and item occurs in at least k posts. We use k = 5, and we finally retain 112 users, 567 tags, and 234 artists.

5.2 Evaluation Metrics

We perform 4-fold cross validation and the default size of the training set is 75% – we pick, for each user, 75% of his posts randomly. The task of each item recommendation algorithm is to predict the items of the user's 25% remaining posts. As performance measures we use precision and recall, which are standard in such scenarios.

For a test user that receives a list of N recommended items (top-N list), the following are defined:

– Precision: is the ratio of the number of relevant items in the top-N list (i.e., those items in the top-N list that are used/tagged by the test user) relative to N.

– Recall: is the ratio of the number of relevant items in the top-N list relative to the total number of relevant items (all items used/tagged by the test user).

5.3 Settings of the Algorithms

Tensor Reduction algorithm: The numbers c_1, c_2, and c_3 of left singular vectors of matrices $U^{(1)}$, $U^{(2)}$, $U^{(3)}$ after appropriate tuning are set to 40, 80 and 190. Due to lack of space we do not present experiments for the tuning of c_1, c_2, and c_3 parameters. The core tensor dimensions are fixed, based on the aforementioned c_1, c_2, and c_3 values.

FolkRank algorithm: We set the damping factor d = 0.7 and stop computation after 10 iterations or when the distance between two consecutive weight vectors is less than 10^{-6}. For the preference vector $\mathbf{p}$, we give higher weights to the user and the tag from the post which is chosen. While each user, tag and item gets a preference weight

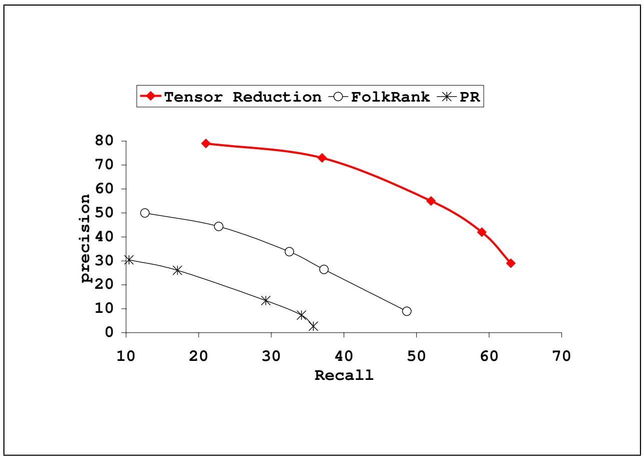

Figure 5. Comparison of the Tensor Reduction, FolkRank and the PR algorithms for the Last.fm data set

of 1, the user and tag from that particular post gets a preference weight of 1 + $|U|$ and 1 + $|T|$, respectively.

PR algorithm: Initially, we set the uniform authority score for each user equal to 1.0. Then, the authority score of users is computed via an iterative algorithm similar to HITS.

5.4 Comparison Results

We compare the Tensor Reduction algorithm with FolkRank and PR, in terms of precision and recall. This reveals the robustness of each algorithm in attaining high recall with minimal losses in terms of precision. We examine the top-N ranked list, which is recommended to a test user, starting from the top item. In this situation, the recall and precision vary as we proceed with the examination of the top-N list. For our data set, N is between [1..5].

In Figure 5, we plot a precision versus recall curve for all three algorithms. As it can be seen, Tensor Reduction algorithm attains the best performance. The reason is that Tensor Reduction exploits altogether the information that concerns the three entities (users, tags, items) and thus, it is able to provide more accurate recommendations.

6 CONCLUSIONS

We developed a unified framework to model the three types of entities that exist in social tagging systems: users, tags and items. This data is represented by a 3-order tensor, on which latent semantic analysis and dimensionality reduction is applied using the higher-order singular value decomposition technique. Our approach can be used to generate personalized recommendations of musical items, which can capture users multimodal perception of music. We performed thorough experimental comparison of our approach against two state-of-the-art recommendations algorithms on real Last.fm data. Our method achieves significant improvements measured through recall/precision.

As future work, we intend to apply different weighting schemes for the initial construction of a tensor. Also, we will adjust our Tensor Reduction framework so that the newly emerged users, tags or items of a social tagging system, can be handled online.

7 REFERENCES

[1] M. Berry, S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1994.

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

[3] 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.

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

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

[6] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *Proc. WWW Conf.*, pages 211–220, 2007.

[7] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, pages 411–426, 2006.

[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999.

[9] M. Levy and M. Sandler. A semantic space for music derived from social tags. In *Proc.ISMIR Conf.*, pages 411–416, 2007.

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

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

[12] L. Page, S. Brin, R. Motwani, and W. T. The pagerank citation ranking:bringing order to the web. In *Technical Report*, 1998.

[13] H. Wang and N. Ahuja. A tensor approximation approach to dimensionality reduction. *International Journal of Computer Vision*, 2007.

[14] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. *Collaborative Web Tagging Workshop at WWW2006*, 2006.

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