

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221141007>

Tag recommendations based on tensor dimensionality reduction

Conference Paper · October 2008

DOI: 10.1145/1454008.1454017 · Source: DBLP

CITATIONS

234

READS

314

3 authors, including:



Panagiotis Symeonidis

Aristotle University of Thessaloniki

94 PUBLICATIONS 2,014 CITATIONS

[SEE PROFILE](#)



Yannis Manolopoulos

Aristotle University of Thessaloniki

535 PUBLICATIONS 10,537 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Data Structures and Parallel Algorithms on Solid State Drives [View project](#)



Bibliometrics by DELAB [View project](#)

Tag Recommendations based on Tensor Dimensionality Reduction

Panagiotis Symeonidis Alexandros Nanopoulos Yannis Manolopoulos

Aristotle University, Department of Informatics, Thessaloniki 54124, Greece
{symeon, ananopou, manolopo}@csd.auth.gr

ABSTRACT

Social tagging is the process by which many users add meta-data in the form of keywords, to annotate and categorize information items (songs, pictures, web links, products etc.). Collaborative tagging systems recommend tags to users based on what tags other users have used for the same items, aiming to develop a common consensus about which tags best describe an item. However, they fail to provide appropriate tag recommendations, because: (i) users may have different interests for an information item and (ii) information items may have multiple facets. In contrast to the current tag recommendation algorithms, our approach develops a unified framework to model the three types of entities that exist in a social tagging system: users, items and tags. These data is represented by a 3-order tensor, on which latent semantic analysis and dimensionality reduction is performed using the Higher Order Singular Value Decomposition (HOSVD) technique. We perform experimental comparison of the proposed method against two state-of-the-art tag recommendations algorithms with two real data sets (Last.fm and BibSonomy). Our results show significant improvements in terms of effectiveness measured through recall/precision.

Categories and Subject Descriptors

H.3.3 [Information Search-Retrieval]: Information Filtering

1. INTRODUCTION

Social tagging is the process by which many users add metadata in the form of keywords to annotate and categorize information items (songs, pictures, web links, products etc.). Many systems, such as Last.fm, Flickr, BibSonomy and Amazon use social tagging to categorize their information items and help users to share them. Moreover, based on social tagging, these systems are able to form tag categories and communities of users.

Many social tagging systems recommend tags to users based on what tags other users have used for the same items. Tag recommendations can expose different facets of an information item and relieve users of the tedious task of manually entering useful tags. Thus, tag recommendation can reduce the problem of data sparsity in social tagging systems, which results by the unwillingness of users to provide an adequate number of tags. Moreover, tag recommendation confronts the vocabulary divergence problem and the ambiguity in the meaning of tags. The basic concern for a tagging system is to become relatively “stable” with time and use. By “stable”, we mean to indicate that users have developed some consensus about which tags best describe an information item. Thus, by offering stability, tag recommendation helps users to find, share and integrate information items more easily.

Social tagging systems often fail to provide the appropriate tag recommendations, which affects the retrieval of items. The reasons are the following: (i) users may have different interests for an information item and (ii) information items have multiple facets. As a simple example, consider a social tagging system for images. Assume two users, one is a computer fan and the other likes fruits. The former selects an image of a Macintosh computer and the latter an image of a green apple. If they do not receive any tag recommendation, then they may both tag the images as “apple”. Thus, after a while, when they provide the tag “apple” to retrieve relevant images, they will receive both images (Mac and fruit). If the former user is recommended the tag “apple computer” and the latter “apple fruit”, the ambiguity could be resolved.

For the above reasons, recent research has focused on developing tag recommendation algorithms [8, 18], which try to exploit tags given by users on specific items (henceforth, these data is called *usage data*). However, existing tag recommendation algorithms do not consider the 3 dimensions of the problem (i.e., users, items, tags) altogether. In contrast, they split the 3-dimensional usage data into three 2-dimensional (pairwise) relations: {user, item}, {user, tag} and {tag, item}. Thus, they miss a part of the semantics that is carried by the 3-dimensions. A more severe disadvantage is presented from regular recommender systems, like Collaborative Filtering (CF), [2, 6, 7, 11], which are applied only to 2-dimensional data (e.g., users and items).

In this paper, we perform 3-dimensional analysis on the usage data, attempting to discover the latent factors that govern the associations among these multi-type objects. Consequently, tags can be recommended according to the cap-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RecSys'08, October 23–25, 2008, Lausanne, Switzerland.
Copyright 2008 ACM 978-1-60558-093-7/08/10 ...\$5.00.

tured associations. This approach faces two challenges:

(i) The relations among users, items and tags are complicated. There exist intra-relations among objects of the same type, as well as inter-relations among objects of different types. For personalized tag recommendations, what we are concerned about are the 3-order relations among them. That is, given a user and an item, the purpose is to predict whether and how much the user is likely to tag this item with a specific tag. Therefore, a unified framework is needed to model the multi-type objects and the multi-type relations among them.

(ii) The three-way data is highly sparse. As we know, most CF algorithms are susceptible to data sparsity. For usage data, the sparseness problem becomes more serious because each user only tags a small number of items. Latent Semantic Indexing (LSI) [3] has been proved useful to address the data sparseness problem in 2-dimensional data recommender systems, however, it is still an open problem for the 3-dimensional data case.

To address the first challenge, we need an approach dealing with 3-dimensional usage data. In this paper, we develop a unified framework to model the three types of dimensions: user, items and tags. The 3-dimensional data is represented by a 3-order tensor. To address the second challenge, we perform 3-mode analysis, using the Higher Order Singular Value Decomposition (HOSVD) technique [13].

The contributions of our approach are summarized as follows:

- We develop a unified framework to model the three types of entities that exist in social tagging systems: user, items and tags. The 3-dimensional data is represented by a 3-order tensor.
- We apply dimensionality reduction in 3-order tensors to reveal the latent semantic associations between users, items and tags.
- We perform experimental comparison of the proposed method against two state-of-the-art tag recommendations algorithms, using two real data sets (Last.fm and BibSonomy). 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 contains the description about Singular Value Decomposition (SVD) and HOSVD. The proposed approach is described in Section 4. Experimental results are given in Section 5. Finally, Section 6 concludes this paper.

2. RELATED WORK

In this section we briefly present some of the research literature related to social tagging, CF recommendation algorithms and tag recommendation algorithms.

Social tagging is the process by which many users add metadata in the form of keywords to share content. So far, the literature has studied the strengths and the weaknesses of social tagging systems. In particular, Golder and Huberman [4] analyzed the structure of collaborative tagging systems as well as their dynamic aspects. Moreover, Halpin

et al. [5] produced a generative model of collaborative tagging in order to understand the dynamics behind it. They claimed that there are three main entities in any tagging system: users, items and tags.

In contrast to the above ternary relation, many recommender systems use Collaborative Filtering (CF) to recommend items based on preferences of similar users, by exploiting a two-way relation of users and items [2]. Because of the ternary relational nature of social tagging, CF cannot be applied directly, unless the ternary relation is reduced to a lower dimensional space. Jaschke et al. [10], considered two alternative 2-dimensional projections to model the relationship among users, items and tags, in order to apply CF in social tagging. These projections preserve the user information and lead to log-based like recommender systems based on occurrence or non-occurrence of items, or tags, respectively, with the users.

Other tag recommendation algorithms are based on conceptual structures similar to the hyperlink structures used in Search Engines. For example, Collaborative Tag Suggestions algorithm [18], also known as Penalty-Reward algorithm (PR) uses an authority score for each user. The authority score measures how well each user has tagged in the past. This authority score can be computed via an iterative algorithm similar to HITs [12]. Moreover, the PR algorithm “rewards” the high correlation among tags, whereas it “penalizes” the overlap of concepts among the recommended tags to allow high coverage of multiple facets for an item.

Another state-of-the-art tag recommendation algorithm is FolkRank [8]. FolkRank exploits the conceptual structures created by people inside the social tagging systems. Their method is inspired by the seminal PageRank [14] algorithm, which reflects the idea that a web page is important if there are many pages linking to it, and if those pages are important themselves. FolkRank employs the same underlying principle for Web Search and Ranking in social tagging. The key idea of FolkRank algorithm is that an item which is tagged with important tags by important users become important itself. The same holds, for tags and users, thus we have a tripartite graph of vertices which mutually reinforces each other by spreading their weights. FolkRank is like the Personalized PageRank, which is a modification of global PageRank and was first proposed for personalized Web search in [14].

As described, all the aforementioned algorithms split the 3-dimensional space into pair relations {user, item}, {user, tag} and {tag, item}, that are only 2-dimensional. Thus, they miss a part of the total interaction between the three dimensions. In contrast, our approach develops a unified framework to concurrently model the three dimensions. Thus, usage data is represented by a 3-order tensor, on which latent semantic analysis is performed using the Higher Order Singular Value Decomposition (HOSVD) technique.

The higher-order singular value decomposition technique was proposed in [13]. It is a generalization of singular value decomposition and has been successfully applied for computer vision problems. In particular, Wang and Ahuja [16] present a novel multi-linear algebra based approach to reduced dimensionality representation of multidimensional data, such as image ensembles, video sequences and volume data. Moreover, in the area of Personalized Web Search, Sun et al. proposed CubeSVD [15] to improve Web Search. They claimed that as the competition of Web Search increases,

there is a high demand for personalized Web search. Therefore based on their CubeSVD analysis, which also uses HOSVD technique, web Search activities can be carried out more efficiently. Their experiments were applied in MSN search engine data. Finally, Xu et al. [17] used HOSVD to provide item recommendations. Thus, they compared their work with a standard Collaborative Filtering algorithm, without focusing in tag recommendations. In the next section, we provide more information on HOSVD.

3. TENSORS AND HOSVD

Since our Tensor Reduction approach is based on HOSVD [13], we first briefly review this technique. Henceforth, tensors are denoted by calligraphic upper-case letters ($\mathcal{A}, \mathcal{B}, \dots$), matrices by uppercase letters (A, B, \dots), vectors by bold lower case letters ($\mathbf{a}, \mathbf{b}, \dots$), scalars by lower case letters (a, b, \dots).

HOSVD generalizes SVD [1] to multi-dimensional matrices, which are called *tensors*. The SVD of a matrix $F_{I_1 \times I_2}$ can be written as a product of three matrices, as shown in Equation 1 and Figure 1:

$$F_{I_1 \times I_2} = U_{I_1 \times I_1} \cdot S_{I_1 \times I_2} \cdot V_{I_2 \times I_2}^T, \quad (1)$$

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 .

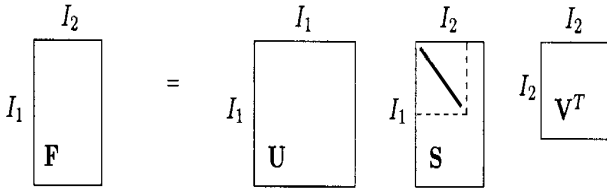


Figure 1: Visualization of SVD on matrix F .

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) [3], to deal with the latent semantic associations of terms in texts 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 [13].

A N -order tensor \mathcal{A} is denoted as $\mathcal{A} \in R^{I_1 \times \dots \times I_N}$, with elements real numbers of the form a_{i_1, \dots, i_N} . In the following, for the purposes of our approach, we only use 3-order tensors (the three dimensions are: users, items and tags).

To apply HOSVD on a 3-order tensor \mathcal{A} , three *matrix unfolding* operations are defined as follows [13]:

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

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

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

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 2.

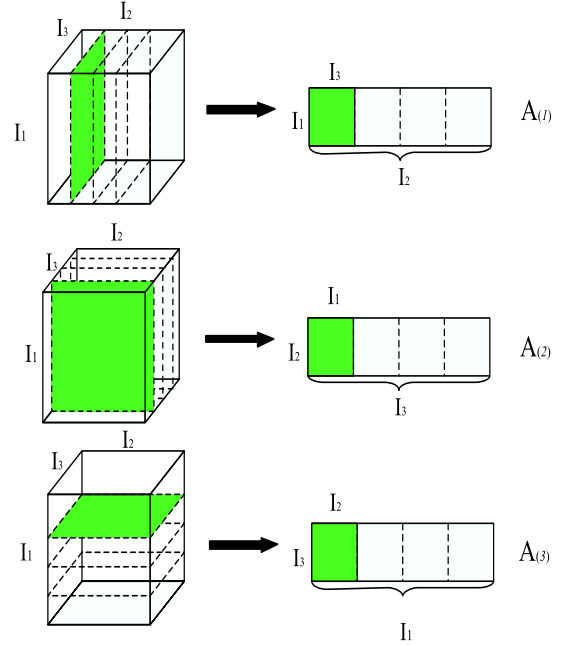


Figure 2: Visualization of the three unfoldings of a 3-order tensor.

Example: Assume a tensor $\mathcal{A} \in R^{3 \times 2 \times 3}$ with $a_{111} = a_{112} = a_{211} = -a_{212} = 1, a_{213} = a_{311} = a_{313} = a_{121} = a_{122} = a_{221} = -a_{222} = 2, a_{223} = a_{321} = a_{323} = 4, a_{113} = a_{312} = a_{123} = a_{322} = 0$. Figure 3 illustrates tensor \mathcal{A} .

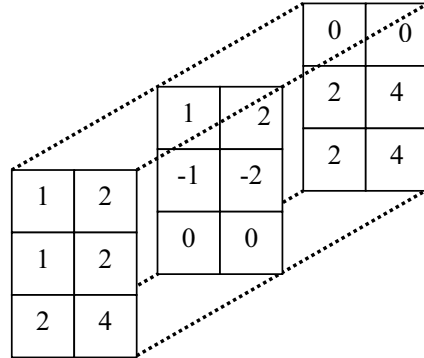


Figure 3: Visualization of a tensor $\mathcal{A} \in R^{3 \times 2 \times 3}$.

$A_1 \in R^{I_1 \times I_2 I_3}$, i.e., the 1-mode matrix unfolding of \mathcal{A} , is given in Figure 4.

Next, we define the n -mode product of an N -order tensor $\mathcal{A} \in R^{I_1 \times \dots \times I_N}$ 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 $(I_1 \times I_2 \times \dots \times I_{n-1} \times J_n \times I_{n+1} \times \dots \times I_N)$ -tensor, the entries of which are defined as follows:

$$A_1 = \begin{bmatrix} 1 & 1 & 0 & 2 & 2 & 0 \\ 1 & -1 & 2 & 2 & -2 & 4 \\ 2 & 0 & 2 & 4 & 0 & 4 \end{bmatrix}$$

Figure 4: Visualization of 1-mode matrix unfolding of tensor \mathcal{A} , which results to matrix A_1 .

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

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

In terms of n -mode products, SVD (Equation 1) on a regular two-dimensional matrix (i.e., 2-order tensor), can be rewritten as follows [13]:

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

where $U^{(1)} = (u_1^{(1)} u_2^{(1)} \dots u_{I_1}^{(1)})$ is a *unitary* ($I_1 \times I_1$)-matrix, $U^{(2)} = (u_1^{(2)} u_2^{(2)} \dots u_{I_2}^{(2)})$ is a *unitary* ($I_2 \times I_2$)-matrix and S is an ($I_1 \times I_2$)-matrix with the following properties:

- (i) pseudodiagonality ($S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{\min\{I_1, I_2\}})$),
- (ii) ordering ($\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min\{I_1, I_2\}} \geq 0$).

By extending this form of SVD, the HOSVD of 3-order tensor \mathcal{A} can be written as follows [13]:

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

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. This process is illustrated in Figure 5.

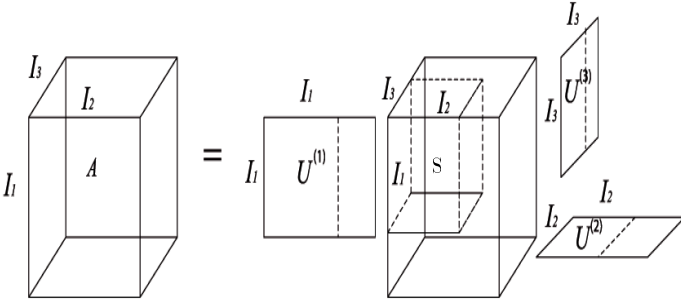


Figure 5: Visualization of HOSVD.

4. THE PROPOSED APPROACH

In this section we present our proposed approach. First, we provide the outline of the approach through a motivating example, to grasp the essential characteristics of the approach. Then, we analyze the steps of the proposed algorithm.

4.1 Outline

When using a social tagging system, a user u tags an item i with a tag t , in order to be able to retrieve information items easily. Thus, the tagging system accumulates a collection of usage data, which can be represented by a set of triplets $\{u, i, t\}$.

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 uses 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, items and tags. The elements of $\hat{\mathcal{A}}$ can be represented by a quadruplet $\{u, i, t, p\}$, where p measures the likelihood that user u will tag item i with tag t . Therefore, tags can be recommended to u according to their weights associated with $\{u, i\}$ pair.

In this subsection, in order to illustrate how our approach works, we apply the Tensor Reduction algorithm to a running example. As illustrated in Figure 6, 3 users tagged 3 different items. In Figure 6, the part of an arrow line (sequence of arrows with the same annotation) between a user and an item represents that the user tagged the corresponding item and the part between an item and a tag indicates that the user tagged this item with the corresponding tag. Thus, the annotated numbers on the arrow lines gives the correspondence between the three types of objects. For example, user U_1 tagged item I_1 with tag “IBM Computer”, denoted as T_1 . The remaining tags are “Apple Computer”, denoted as T_2 , “Apple Fruit”, denoted as T_3 .

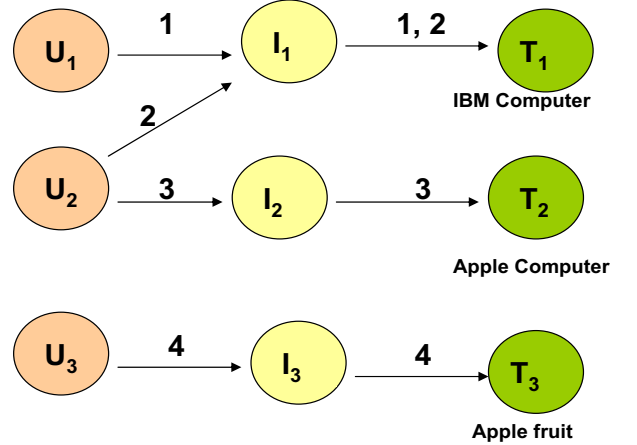


Figure 6: Usage data of the running example

From Figure 6, we can see that users U_1 and U_2 have common interests on computers, while user U_3 is interested in apple fruits. A 3-order tensor $\mathcal{A} \in R^{3 \times 3 \times 3}$, can be constructed from the usage data. We use the co-occurrence frequency (denoted as weight) of each triplet user, item and tag as the elements of tensor \mathcal{A} , which are given in Table 1.

After performing the Tensor Reduction analysis (details of how to do this are given in the following section), we can get the reconstructed tensor of $\hat{\mathcal{A}}$, which is presented in Table 2, whereas Figure 7 depicts the contents of $\hat{\mathcal{A}}$ graphically (the weights are omitted). As shown in Table 2 and

Arrow Line	User	Item	Tag	Weight
1	U_1	I_1	T_1	1
2	U_2	I_1	T_1	1
3	U_2	I_2	T_2	1
4	U_3	I_3	T_3	1

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

Figure 7, the output of the Tensor Reduction algorithm for the running example is interesting, because a new association among these objects is revealed. The new association is between U_1 , I_2 and T_2 . This association is represented with the last (bold faced) row in Table 2 and with the dashed arrow line in Figure 7).

If we have to recommend to U_1 a tag for I_2 , then there is no direct indication for this task in the original tensor \mathcal{A} . However, we see that in Table 2 the element of $\hat{\mathcal{A}}$ associated with $\{U_1, I_2, T_2\}$ is 0.44, whereas for U_1 there is no other element associating other tags with I_2 . Thus, we recommend to user U_1 the tag T_2 for item I_2 .

Arrow Line	User	Item	Tag	Weight
1	U_1	I_1	T_1	0.72
2	U_2	I_1	T_1	1.17
3	U_2	I_2	T_2	0.72
4	U_3	I_3	T_3	1
5	U_1	I_2	T_2	0.44

Table 2: Tensor Constructed from the usage Data of the running example.

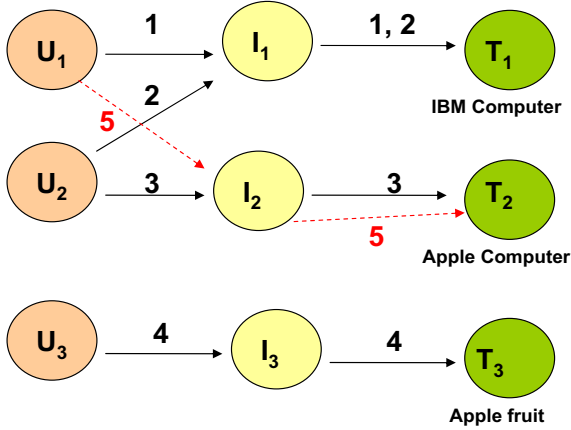


Figure 7: Illustration of the Tensor Reduction Algorithm output for the running example

The resulting recommendation is reasonable, because U_1 is interested in computers rather than fruits. That is, the Tensor Reduction approach is able to capture the latent associations among the multi-type data objects: user, item and tags. The associations can then be used to improve the tag recommendation procedure, as will be verified by our experimental results.

4.2 The Tensor Reduction Algorithm

In this section we provide details on how HOSVD is applied on tensors and how tag recommendation is performed based on the detected latent associations.

Our Tensor Reduction approach initially constructs a tensor, based on usage data triplets $\{u, i, t\}$ of users, items and tags. The motivation is to use all three objects 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 $\hat{\mathcal{A}}$. The 6 steps of the proposed approach are summarized as follows:

- *Step 1:* The initial tensor \mathcal{A} construction, which is based on usage data triplets (user, item, tag).
- *Step 2:* The matrix unfoldings of tensor \mathcal{A} , where we matricize the tensor in all three modes, creating three new matrices (one for each mode).
- *Step 3:* The application of SVD in all three new matrices, where we keep the c -most important singular values for each matrix.
- *Step 4:* The construction of the core tensor \mathcal{S} , that reduces the dimensionality.
- *Step 5:* The construction of the $\hat{\mathcal{A}}$ tensor, that is an approximation of tensor \mathcal{A} .
- *Step 6:* Based on the weights of the elements of the reconstructed tensor $\hat{\mathcal{A}}$, we recommend tags to the target user u .

Steps 1 – 5 build a model and can be performed off-line. The recommendation in Step 5 is performed on-line, i.e., each time we have to recommend a tag to a user, based on the built model. In the following, we provide more details on each step.

4.3 The initial construction of tensor \mathcal{A}

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

4.4 Matrix unfolding of tensor \mathcal{A}

As described in Section 3, a tensor \mathcal{A} can be unfolded (matricized), i.e., we build matrix representations of tensor \mathcal{A} in which all the column (row) vectors are stacked one after the other. 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:

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

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

$$A_3 \in R^{I_t \times I_u I_i}$$

4.5 Application of SVD on each matrix

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

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

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

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

For Tensor Dimensionality Reduction, there are three dimensional parameters to be determined. The numbers c_1, c_2 and c_3 of left singular vectors of matrices $U^{(1)}, U^{(2)}, U^{(3)}$, respectively, that are preserved. They will determine the final dimensionality 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 numbers of principal components for each matrix $U^{(1)}, U^{(2)}, U^{(3)}$. The numbers c_1, c_2 and c_3 of singular vectors are chosen by preserving a percentage of information of the original S_1, S_2, S_3 matrices after appropriate tuning (the default percentage is set to 50% of the original matrix).

4.6 The construction of the core tensor S

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

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

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

4.7 The construction of tensor $\hat{\mathcal{A}}$

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

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

S is the 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.

4.8 The generation of the Tag Recommendation list

Tensor $\hat{\mathcal{A}}$ measures the associations among the users, items and tags and acts as a model that is used during the recommendation.

Each element of $\hat{\mathcal{A}}$ represents a quadruplet $\{u, i, t, p\}$, where p is the likeliness that user u will tag item i with tag t .

Therefore, for a user u and an item i , tags can be recommended according to their weights associated with $\{u, i\}$ pair. If we want to recommend to u N tags for item i , then we select the N corresponding tags with the highest weights.

5. EXPERIMENTAL RESULTS

In this section, we present experimental results for the performance of our approach, compared to two state-of-the-art tag-recommendation algorithms. Henceforth, our proposed approach is denoted as Tensor Reduction. We use FolkRank [8] and the Collaborative Tag Suggestions [18] algorithm (known as Penalty-Reward algorithm), denoted as FolkRank and PR, respectively as comparisons to our approach.

5.1 Data Sets

To evaluate the examined algorithms, we have chosen real data sets from two different social tagging systems: BibSonomy and Last.fm, which have been used as benchmarks in past works [8].

BibSonomy: We used a snapshot of all users, items (both publication references and bookmarks) and tags publicly available on April 30, 2007. From this snapshot, we excluded the posts from the DBLP computer science bibliography since they are automatically inserted and all owned by one user and all tagged with the same tag (dblp). The number of users, items and tags are 1,037, 28,648 and 86,563, respectively.

Last.fm: The data for Last.fm was gathered during October 2007, partly through the web services API (collecting user nicknames), partly crawling the Last.fm site. Here the items correspond to artist names, which are already normalized by the system. There are 12,773 triplets in the form user-artist-tag. These triplets correspond to 4,442 users, 1,620 artists and 2,939 tags.

Following the approach of [8] to get more dense data, 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 has/occurs in at least k posts. For both data sets we used $k = 5$. Thus, for the Bibsonomy data set there are 105 users, 246 items and 591 tags, whereas for the Last.fm data set there are 112 users, 234 items and 567 tags.

5.2 Experimental Configuration

We performed 4-fold cross validation, thus each time we divide the data set into a training set and a test set with sizes 75% and 25% of the original set, respectively. All algorithms had the task to predict the tags of the users' postings in the test set.

Based on the approach of [9, 7], a more realistic evaluation of recommendation should consider the division of tags of each test user into two sets: (i) the *past* tags of the test user and, (ii) the *future* tags of the test user. Therefore, for a test user we generate the recommendations based only on the tags in his past set. This simulates the real-world applications, where users gradually tag items and receive recommendations before they provide all their tags. As most existing works ignore this division, their reported performance corresponds to the best case, because they indirectly exploit

apriori known information (the tags in the future set). For example, assume a user in the test set, which has posted 10 tags for a particular item. According to the approach in previous works, all the 10 tags are used to calculate his similarity with training users. In contrast, if we consider 7 tags to belong in the past set, then we compute similarities based only on the past 7 tags and we have to predict the remaining 3 tags (those in the future set). With the division into past and future sets, the accuracy is expected to decrease compared to the best case when the two sets are identical. However, the reported performance is more indicative for real-world applications. The default sizes of the past and future sets are 50% and 50%, respectively, of the number of tags posted by each test user.

As performance measures we use the classic metrics of precision and recall. For a test user that receives a list of N recommended tags (top- N list), precision and recall are defined as follows:

- *Precision* is the ratio of the number of relevant tags in the top- N list (i.e., those in the top- N list that belong in the future set of tags posted by the test user) to N .
- *Recall* is the ratio of the number of relevant tags in the top- N list to the total number of relevant tags (all tags in the future set posted by the test user).

5.3 Algorithms' Settings

For each of the algorithms of our evaluation we will now describe briefly the specific settings used to run them:

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 60, 105 and 225 for the BibSonomy data set, whereas are set to 40, 80 and 190 for the Last.fm data set. 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 for each data set, respectively.

FolkRank algorithm: We set the damping factor $d = 0.7$ and stopped computation after 100 iterations or when the distance between two consecutive weight vectors was less than 10^{-6} . For the preference vector \mathbf{p} , we gave higher weights to the user and the item from the post which was chosen. While each user, tag and resource got a preference weight of 1, the user and resource from that particular post got a preference weight of $1 + |U|$ and $1 + |I|$, respectively.

PR algorithm: Initially, we set the uniform authority score for each user equal to 1.0. The authority score $a(u)$ is computed via an iterative algorithm similar to HITS.

5.4 Results

In this section, we proceed with the comparison of Tensor Reduction 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 the BibSonomy data set (N is between [1..5]), in Figure 8, we plot a precision versus recall curve for all three algorithms. As shown, the precision of each algorithm falls as N increases. In contrast, as N increases, recall for all five algorithms increases too. Tensor Reduction algorithm attains 68% precision, when we recommend a top-1 list of tags. In contrast, FolkRank gets a precision of 42%. Moreover, Tensor Reduction is more effective than FolkRank getting a maximum recall of 44%, while the latter's is 36%. This experiment shows that Tensor Reduction is more robust in finding relevant tags for the test user. The reason is that Tensor Reduction exploits all information that concerns the three objects (users, items, tags) and through HOSVD, it addressed sparsity and finds latent associations.

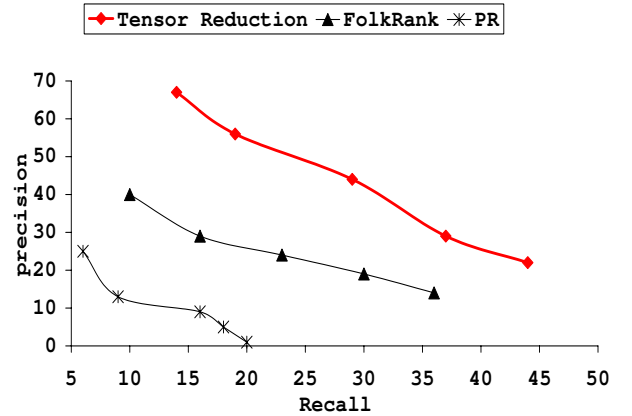


Figure 8: Comparison of Tensor Reduction, FolkRank and PR algorithms for the BibSonomy dataset.

For the Last.fm data set (N is between [1..5]), in Figure 9, we plot also a precision versus recall curve for all three algorithms. Tensor Reduction algorithm again attains the best performance. Despite the different nature of the two data sets (the one is for bibliographic data and the other for musical data), we observe similar behavior of algorithms for both data sets. It is important that Tensor Reduction provides more accurate recommendations in both cases.

6. CONCLUSIONS

Collaborative tagging systems recommend tags to users based on what tags other users have used for the same items, aiming to develop a common consensus about which tags best describe an item.

In this paper, we developed a unified framework to model the three types of entities that exist in a social tagging system: users, items and tags. These data is represented by a 3-order tensor.

We applied dimensionality reduction in a 3-order tensor, to reveal the latent semantic associations between users, items and tags. The latent semantic analysis and dimensionality reduction is performed using the Higher Order Singular Value Decomposition (HOSVD) technique.

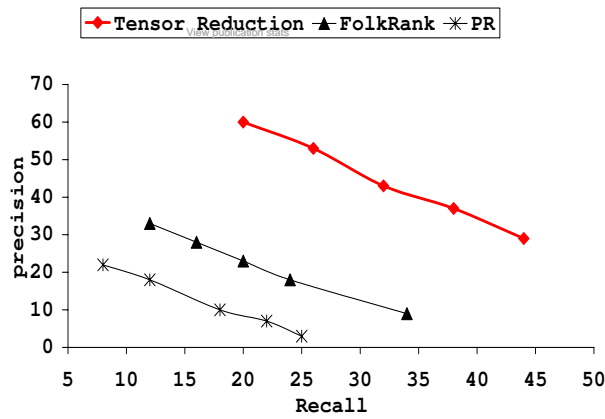


Figure 9: Comparison of Tensor Reduction, FolkRank and PR algorithms for the Last.fm dataset.

We also performed experimental comparison of the proposed method against two state-of-the-art tag recommendations algorithms, with two real data sets (Last.fm and BibSonomy). Our results show substantial improvements in terms of effectiveness measured through recall/precision.

As future work, we intend to examine the following topics:

- To examine different methods for extending SVD to high-order tensors. Another approach for multi dimensional decompositions is the Parallel factor analysis (Parafac).
- To apply different weighting methods for the initial construction of a tensor. A different weighting policy for the tensor's initial values could improve the overall performance of our approach.
- To adjust our Tensor Reduction framework in order to be able of handling on-line, the newly emerged objects (new users, new items and new tags), at the time they are inserted in a social tagging system. This may result to 4-dimensional tensors, where time represents the additional dimension.
- The extension of the method for the recommendation of items. This concerns the cases where a user by providing a tag in a social tagging system requests for item recommendations.

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] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. Conf. on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
- [3] G. Furnas, S. Deerwester, and S. e. a. Dumais. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proc. ACM SIGIR Conf.*, pages 465–480, 1988.
- [4] S. Golder and B. Huberman. The structure of collaborative tagging systems. In *Technical Report*, 2005.
- [5] H. Halpin, V. Robu, and H. Shepherd. The dynamics and semantics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 211–220, 2007.
- [6] J. Herlocker, J. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4):287–310, 2002.
- [7] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. on Information Systems*, 22(1):5–53, 2004.
- [8] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, pages 411–426, 2006.
- [9] Z. Huang, H. Chen, and D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems*, 22(1):116–142, 2004.
- [10] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007*, pages 506–514.
- [11] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proc. ACM CIKM Conf.*, pages 247–254, 2001.
- [12] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [13] L. d. Lathauwer, B. d. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal of Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [14] L. Page, S. Brin, R. Motwani, and W. T. The pagerank citation ranking: bringing order to the web. In *Technical Report*, 1998.
- [15] J. Sun, D. Shen, H. Zeng, Q. Yang, Y. Lu, and Z. Chen. Cubesvd: a novel approach to personalized web search. In *World Wide Web Conference*, pages 382–390, 2005.
- [16] H. Wang and N. Ahuja. A tensor approximation approach to dimensionality reduction. *International Journal of Computer Vision*, pages 217–229, 2007.
- [17] Y. Xu, L. Zhang, and W. Liu. Cubic analysis of social bookmarking for personalized recommendation. In *Frontiers of WWW Research and Development - APWeb 2006*, pages 733–738, 2006.
- [18] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. *Collaborative Web Tagging Workshop*, 2006.