Recommending Posts in Political Blogs based on Tensor Dimensionality Reduction

Panagiotis Symeonidis[1] Anastasia Deligiaouri[2]

[1]Department of Informatics

Aristotle University, Greece

symeon@csd.auth.gr

[2] Department of Public Relations and Communication

ATEI Kastorias, Greece

a.deligiaouri@kastoria.teikoz.gr

Abstract

Social Tagging is the process by which many users add metadata in the form of keywords, to annotate and categorize items (posts, songs, pictures, web links, products etc.). Political blogs can recommend posts to users, based on tags they have in common with other similar users. However, a post in politics may be interpreted in a number of ways by different users. This is because terms, especially in politics, carry an ideological burden and therefore it is very likely for posts to present a semantic ambiguity. The significance of this study is that in contrast to current recommendation algorithms, we apply Higher Order Singular Value Decomposition (HOSVD) on a 3-dimensional tensor to find latent semantic relationships between the three types of entities that exist in a social blogging system: users, posts, and tags. We perform experimental comparison of the proposed method against state-of-the-art recommendation algorithms with two real data sets (Wordpress and Technorati). 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 songs, pictures, products, etc. Social tagging is associated to the "Web 2.0" technologies and has already become an important source of information for recommender systems. For example, political blogs found in Wordpress [1] and Technorati [2] allow users[3] to tag posts. Social tags carry useful information not only about the items they label, but also about the users who tagged. Thus, social tags are a powerful mechanism that reveal 3-dimensional correlations between users, tags, and items.

Political blogs constitute a powerful part of blogosphere. The use of political blogs have offered new perspectives in public deliberation. Today thousands of political blogs are being traced by technorati.com and wordpress.com. However, posts in political blogs carry a significant "semantic ambiguity" and are highly opinionated. That is, posts in political blogs can never escape from the ideological pre-suppositions of the user and they may signify a different meaning, according to the ideology prism used to interpret them. Thus, it is very likely a social tag in a political blog to have different meanings, if it is put in a different national or political context.

Blogging, as a form of "self publication" reflects both the ideological as long as the sociopolitical framework within the blogger is placed. Henceforth the layers of meaning in posts and in tags are dependent on several variables such as the historical knowledge [17], the political context of the message or even the mainstream ideology of a specific society. Because of this feature, it is very possible for blogs to constitute like-minded communities with bloggers sharing the same referential experiences, excluding users with oppositional views. Due to this effect, the credibility of political information is strongly criticized [14]. The example we present below with the use of the terms liberal and conservatives in European and and US context proves this statement. However, the aspect of personalization and contextualization in blogs can be reduced to a certain degree by the 3 order tensor proposed in this paper in order to make political information acquired from political blogs more objective and less biased.

Several social tagging systems (STSs), e.g., Wordpress, Technorati, Amazon, YouTube, etc., recommend items to users, based on tags they have in common with other similar users. Traditional recommender systems use techniques such as Collaborative Filtering (CF) [3, 9, 10, 16], which apply to 2-dimensional data, i.e., users and items. Thus, such systems do not capture the multimodal use of tags.

[1] http://en.wordpress.com/tag/politics/

[2] http://technorati.com/politics/

[3] In the paper, the word "user" means the author of a post in a blog.

To alleviate this problem, [27] proposes a generic method that allows tags to be incorporated to standard CF algorithms, by reducing the three-dimensional correlations to three two dimensional correlations and then applying a fusion method to re-associate these correlations.

1.1 Motivation

Existing algorithms do not consider the 3 dimensions of the problem. In contrast, they split the 3-dimensional space into pair relations {user, item}, {user, tag}, and {tag, item}, that are 2-dimensional, in order to apply already existing techniques like CF, link mining, etc. Therefore, they miss a part of the total interaction between the three dimensions. What is required is a method that is able to capture the three dimensions all together without reducing them into lower dimensions.

Moreover, the existing approaches fail to reveal the latent associations between tags, users, and items. Latent associations exist due to three reasons: (i) users have different interests for an item, (ii) items have multiple facets, and (iii) tags have different meanings for different users. As an example, assume two users in an STSs for web bookmarks (e.g., Del.icio.us). The first user is a car fan and tags a site about cars, whereas the other tags a site about wild cats. Both use the tag "jaguar". When they provide the tag "jaguar" to retrieve relevant sites, they will receive both sites (cars and wild cats). Therefore, what is required is a method that can discover the semantics that are carried by such latent associations, which, in the previous example can help to understand the different meanings of the tag "jaguar".

A relevant example of semantic ambiguity in politics can be found e.g. in the word "liberal". This word has a different political meaning in USA and in Europe. Liberals represent the left wing politicians in USA while in Europe they are considered as conservative right wing politicians. In USA, it is associated with the welfare-state policies of the New Deal program of Democratic President Franklin D. Roosevelt, whereas in Europe liberals are more commonly conservative in their political and economic outlook (see wikipedia, http://en.wikipedia.org/wiki/Liberalism). Moreover, the word "liberal", when it is used to tag a post may refer to other meanings such as liberal arts. Therefore, the word "liberal" presents a significant "semantic ambiguity" that should be reduced in order to make more accurate posts recommendations.

1.2 Contribution

In this paper, we develop a framework that models the three dimensions, i.e., items, tags, users. The 3-dimensional data are represented by 3-dimensional matrices, which are called *3-order tensors*. We avoid

splitting the 3-dimensional correlations and we handle all dimensions equally. To reveal latent semantics, we perform 3-mode analysis, using the Higher Order Singular Value Decomposition (HOSVD) [19]. Our method reveals latent relations among objects of the same type, as well among objects of different types.

The contributions of our approach are summarized as follows:

- We use a 3-order tensor to model the three types of entities (user, post, and tag) that exist in political blogs.
- We apply dimensionality reduction (HOSVD) in 3-order tensors, to reveal the latent semantic associations between users, posts, and tags.
- We perform extensive experimental comparison of the proposed method against state-of-the-art recommendation algorithms, using Wordpress and Technorati data sets.
- Our method substantially improves accuracy of post recommendations.

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

In this section we briefly present some of the research literature related to Social Tagging. We also present related work in tag, item, and users recommendation algorithms. Finally, we present works that applied HOSVD in various research domains.

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 STSs. In particular, Golder and Huberman [7] analyzed the structure of collaborative tagging systems as well as their dynamical aspects. Moreover, Halpin et al. [8] 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.

Blogging has offered new perspectives in public deliberation. It constructs a lively community that potentially enables citizens' civic engagement. Today more than 118 million blogs being traced by

technorati.com [15]. Four distinct types of blogs have been emerged: Classic, Community, Institutional, and Bridge [15]. Each of them, offers a different kind of user's participation. In this paper, we focus on classic blogs, which serve as platforms of low-cost self-publishing daily news and experiences. Durant and Smith [5] have reported that sentiment classification in classic political blogs appears to be a more difficult task than in traditional texts because of the interplay between language, word meanings and images. For example, the word "ideology" has accepted a multifaceted approach from different political theories (e.g. Marxism and Liberalism). Thus, it is very likely for a social tag (i.e ideology) in a political blog to have different meanings, if it is put in a different national or political context.

In the area of item recommendations, many recommender systems already use Collaborative Filtering (CF) to recommend items based on preferences of similar users, by exploiting a two-way relation of users and items [3]. In 2001, Item-based algorithm was proposed, which is based on the items' similarities for a neighborhood generation [20]. However, because of the ternary relational nature of Social Tagging, 2-way CF cannot be applied directly, unless the ternary relation is reduced to a lower dimensional space. Jaschke et al. [13], in order to apply CF in Social Tagging, considered for the ternary relation of users, items, and tags two alternative 2-dimensional projections. 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. Another recently proposed state-of-the-art item recommendation algorithm is tag-aware Fusion [27]. They propose a generic method that allows tags to be incorporated to standard CF algorithms, by reducing the three-dimensional correlations to three two dimensional correlations and then applying a fusion method to re-associate these correlations.

Differently from existing approaches, our method develops a unified framework to concurrently model all three dimensions. Usage data are modelled by a 3-order tensor, on which latent semantic analysis is performed using the Higher Order Singular Value Decomposition (HOSVD) [19]. Moreover, HOSVD method can be combined with a Kernel-SVD smoothing technique to overcome data sparsity problems [25]. Our method has been used for recommending tags [24] and users [23] in online tagging systems. However, in this paper we recommend posts i.e items. Moreover, HOSVD has been used for music data [26], while this paper focuses on blogging data.

HOSVD is a generalization of singular value decomposition and has been successfully applied in several areas. In particular, Wang and Ahuja [28] present a novel multi-linear algebra based approach to reduced dimensionality representation of multidimensional data, such as image ensembles, video sequences and

volume data. In the area of Data Clustering, Chen et al. [4] used also a high-order tensor. However, they transform the initial tensor (through Clique Expansion algorithm) into lower dimensional spaces, so that clustering algorithms (such as k-means) can be applied. Finally, in the area of Personalized Web Search, Sun et al. proposed CubeSVD [21] 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, Web Search activities can be carried out more efficiently.

3 Preliminaries - Tensors and HOSVD

In this section, we summarize the HOSVD procedure. 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) [2] of a matrix $F_{I_1 \times I_2}$ can be written as a product of three matrices, as shown in Equation 1:

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

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) [6], to deal with the latent semantic associations of terms in texts and to reveal the major trends in F.

Tensors

A *tensor* is a multi-dimensional matrix. A N-order tensor $\mathcal{A}$ is denoted as $\mathcal{A} \in R^{I_1 \cdots I_N}$, 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 [19] 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 [19]:

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

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.

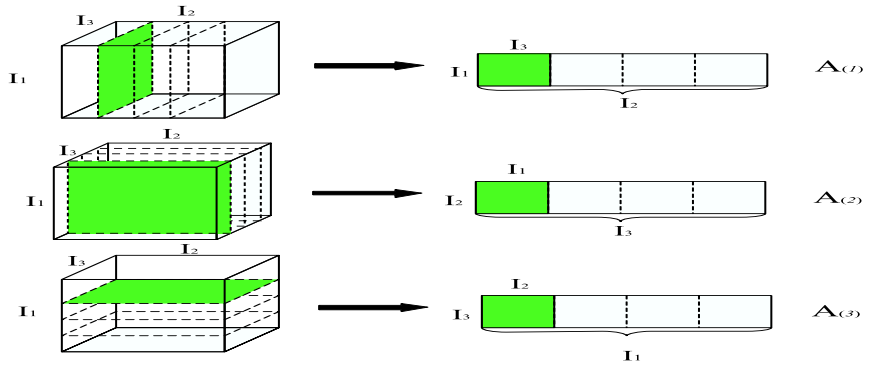


Figure 1: Visualization of the three unfoldings of a 3-order tensor.

Example: Define a tensor $\mathcal{A} \in R^{3\times2\times3}$ 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.

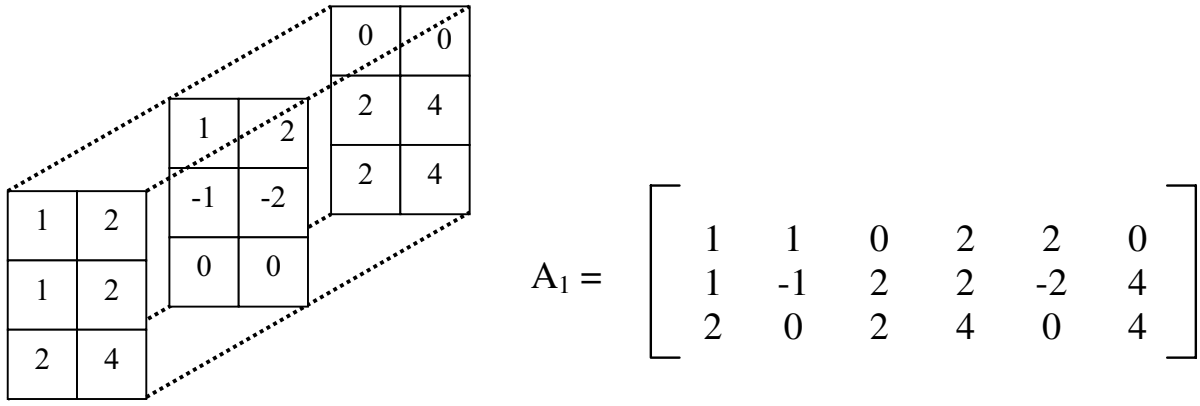


Figure 2: Visualization of tensor $\mathcal{A} \in R^{3\times2\times3}$ and its 1-mode matrix unfolding.

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:

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

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

In terms of n-mode products, SVD on a regular two-dimensional matrix (i.e., 2-order tensor), can be rewritten as follows [19]:

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

where $U^{(1)} = (u_1^{(1)} u_2^{(1)} \dots u_{I_1}^{(1)})$ is a *unitary* $(I_1 \times I_1)$-matrix, $U^{(2)} = (u_1^{(2)} u_2^{(2)} \dots u_{I_1}^{(2)})$ 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: $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{\min\{I_1, I_2\}})$

(ii) ordering: $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min\{I_1, I_2\}} \geq 0$.

By extending this form of SVD, HOSVD of 3-order tensor $\mathcal{A}$ can be written as follows [19]:

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

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

In this section, we elaborate on how HOSVD is applied on tensors and on how the recommendation of items is performed according to the detected latent associations. Note that a similar approach is followed for the tag and user recommendations.

When using a social blogging system, to be able to retrieve information items easily, a user u tags a post i with a tag t. After some time of usage, the blogging 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, posts, and tags. Each element of $\hat{\mathcal{A}}$ can be represented by a quadruplet $\{u, i, t, p\}$, where

p measures the likeliness that user u will tag post i with tag t. Therefore, posts can be recommended to u according to their weights associated with $\{u, t\}$ 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 3, 3 users tagged 3 different items (weblinks). In Figure 3, 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 "CONSERVATIVE", denoted as T_1. The remaining tags are "LIBERAL", denoted as T_2, "DEMOCRATE", denoted as T_3.

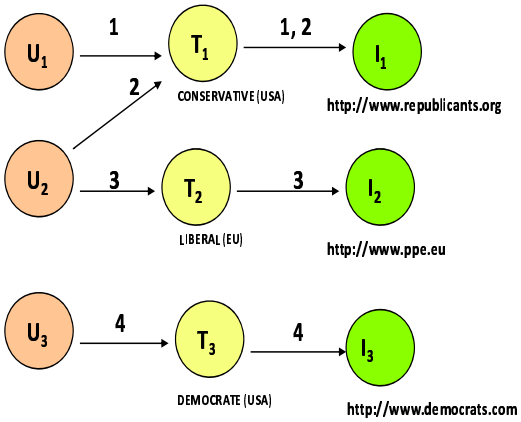


Figure 3: Usage data of the running example

From Figure 3, we can see that users U_1 and U_2 have common interests on conservative ideology, while user U_3 is interested in democratic ideology. A 3-order tensor $\mathcal{A} \in R^{3\times3\times3}$, 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. Note that all associated weights are initialized to 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 4 depicts the contents of $\hat{\mathcal{A}}$ graphically (the weights are omitted). As shown in Table 2 and Figure 4, the output of the Tensor Reduction algorithm for the running example is interesting, because a new association among

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.

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 4).

If we have to recommend to U_1 an item for tag T_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 $\{U_1, T_2, I_2\}$ is 0.44, whereas for U_1 there is no other element associating other tags with I_2. Thus, we recommend item I_2 to user U_1, who used tag T_2.

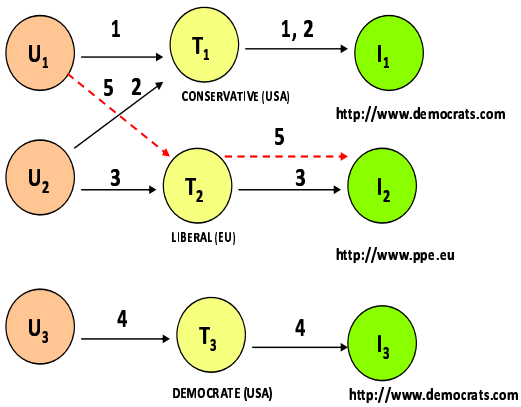


Figure 4: Illustration of the Tensor Reduction Algorithm output for the running example

The resulting recommendation is reasonable, because U_1 is interested in conservative policies rather than democratic policies. 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 item recommendation procedure, as will be verified by our experimental results.

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	$\mathbf{U_1}$	$\mathbf{I_2}$	$\mathbf{T_2}$	**0.44**

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

4.2 The Tensor Reduction Algorithm

The 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.

4.2.1 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. Figure 5 presents the tensor construction of our running example.

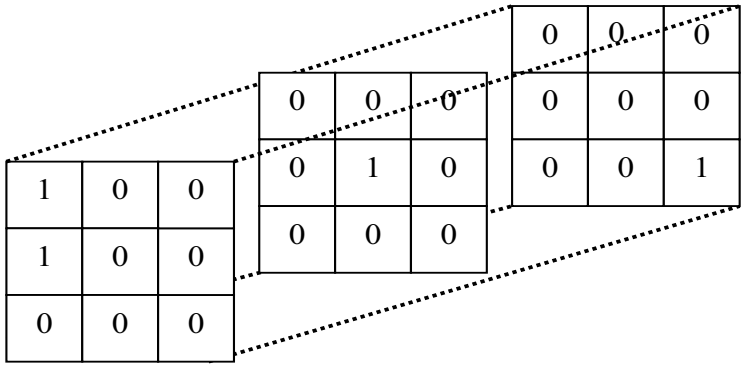


Figure 5: The tensor construction of our running example.

4.2.2 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. In Figure 6, we present the matrix unfoldings of our running example.

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

$$A_1 = \begin{bmatrix} 1&0&0&0&0&0&0&0&0 \\ 1&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \end{bmatrix}, A_2 = \begin{bmatrix} 1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \end{bmatrix}, A_3 = \begin{bmatrix} 1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

Figure 6: The tensor 1-mode, 2-mode and 3-mode matrix unfoldings of our running example.

4.2.3 Application of SVD on each mode

We apply SVD on the three matrix unfoldings A_1, A_2, A_3. For the running example, Figures 7, 8, 9 present these matrixes with the left-singular vectors and the matrixes with the singular values for the decomposition in each mode (to ease presentation, we omit the corresponding matrixes with the right-singular vectors).

$$A_1 = U^{(1)} \cdot S^{(1)} \cdot (V^{(1)})^T \tag{5}$$

-0.53	0	-0.85
-0.85	0	0.53
0	1	0

$U^{(1)}$

1.62	0	0
0	1	0
0	0	0.62

S_1

Figure 7: Example of: $U^{(1)}$ (left singular vectors of A_1), $S^{(1)}$ (singular values of A_1).

$$A_2 = U^{(2)} \cdot S^{(2)} \cdot (V^{(2)})^T \tag{6}$$

1	0	0
0	1	0
0	0	1

$U^{(2)}$

1.41	0	0
0	1	0
0	0	1

S_2

Figure 8: Example of: $U^{(2)}$ (left singular vectors of A_2), $S^{(2)}$ (singular values of A_2).

$$A_3 = U^{(3)} \cdot S^{(3)} \cdot (V^{(3)})^T \tag{7}$$

1	0	0
0	1	0
0	0	1

$U^{(3)}$

1.41	0	0
0	1	0
0	0	1

S_3

Figure 9: Example of: $U^{(3)}$ (left singular vectors of A_3), $S^{(3)}$ (singular values of A_3).

4.2.4 Computing the low-rank approximations

In matrix dimensionality reduction, low-rank approximation is used to filter out the small singular values that introduce "noise". Thus, SVD is truncated to the first c higher singular values and the corresponding singular vectors. This operation is called *thin-SVD* and is used in Latent Semantic Indexing (LSI) [6]. The resulting matrix is denoted as rank-c approximation and SVD is optimal in the sense that it computes the rank-c approximation with the minimum *Frobenious norm*.

In the case of tensor dimensionality reduction, we have to compute a rank-c_1, c_2, c_3 approximation tensor, where c_i is the number of dimensions maintained for i-mode. To compute the rank-c_1, c_2, c_3 approximation, we retain c_i singular values and the corresponding left singular vectors from $U^{(i)}$, when applying SVD on the unfolded matrix A_i of i-mode. The selection of c_1, c_2, c_3 determines the final dimensionality of the core tensor $\mathcal{S}$. 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 a different c_i value for each matrix $U^{(i)}$ $(1 \leq i \leq 3)$. This results to $(U_{c_i}^{(i)})$ matrixes, which denote the c_i-dimensionally reduced $U^{(i)}$ matrix $(1 \leq i \leq 3)$.

Determining the c_1, c_2, and c_3 parameters in Tucker models (like HOSVD) is a tedious task [1]. A

practical option is to use ranks indicated by SVD on unfolded data in each mode. This way, c_1, c_2, and c_3 are chosen by preserving a percentage of information of the original $S^{(1)}$, $S^{(2)}$, $S^{(3)}$ matrices after appropriate tuning. Our experimental results indicate that a 70% of the original diagonal of $S^{(1)}$, $S^{(2)}$, $S^{(3)}$ matrices can give good approximations of A_1, A_2 and A_3 matrices. Notice that a percentage of the original diagonal can be obtain by summing the singular values of S matrix and then by keeping those singular values of S matrix that give us the wanted percentage. In our running example, the diagonal of $S^{(1)}$ matrix (see Figure 7) sums to 3.24 (1.62+1+0.62). Thus, by setting c_1 parameter equal to 2, we keep 80% (2.62/3.24) of the original diagonal of matrix $S^{(1)}$. Due to its simplicity and its efficiency, this approach has been followed in several related works that use tensor decompositions [18, 22]. For this reason we follow this approach too. In our running example c_1 is equal to 2, c_2 is equal to 3, and c_3 is equal to 3. Figure 10 presents the transposes of the dimensionally reduced $U^{(i)}$ matrixes.

-0.53	-0.85	0
0	0	1

$(U_{c_1}^{(1)})^T$

1	0	0
0	1	0
0	0	1

$(U_{c_2}^{(2)})^T$

1	0	0
0	1	0
0	0	1

$(U_{c_3}^{(3)})^T$

Figure 10: The transposes of the dimensionally reduced $U^{(i)}$ matrices.

4.2.5 The core tensor $\mathcal{S}$ construction

The core tensor S governs the interactions among user, item and tag entities. From the the initial tensor $\mathcal{A}$ we proceed to the construction of the core tensor S, as follows:

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

Figure 11 presents the core tensor $\mathcal{S}$ for the running example.

4.2.6 The tensor $\hat{\mathcal{A}}$ construction

Finally, tensor $\hat{\mathcal{A}}$ is built by the product of the core tensor $\mathcal{S}$ and the mode products of the three matrices $U_{c_1}^{(1)}$, $U_{c_2}^{(2)}$ and $U_{c_3}^{(3)}$ as follows:

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

For the current example, the resulting $\hat{\mathcal{A}}$ tensor is presented in Figure 12.

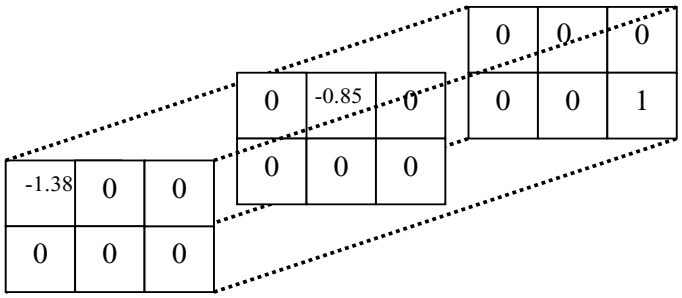


Figure 11: The resulting core tensor $\mathcal{S}$ for the running example.

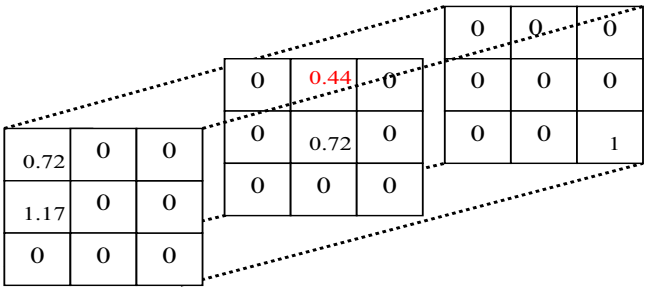


Figure 12: The resulting $\hat{\mathcal{A}}$ tensor for the running example.

4.2.7 The generation of the recommendations

The reconstructed tensor $\hat{\mathcal{A}}$ measures the associations among the users, tags, and items, so that each element of $\hat{\mathcal{A}}$ represents a quadruplet $\{u, t, i, p\}$, where p is the likeliness that user u will tag item i with tag t.

On this basis, items can be recommended to u according to their weights associated with $\{u, t\}$ pair. However, we see that in Figure 12, 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, in our running example, we recommend to user U_1 item I_2 for tag T_2.

5 Experimental Evaluation

In this section, in the area of post recommendations, we compare experimentally our approach with state-of-the-art item recommendation algorithms. Henceforth, our proposed approach is denoted as Tensor Reduction. We use in the comparison the tag-aware Fusion algorithm [27] and the Item-based CF algorithm [20], denoted as Fusion and Item-based, respectively. Our experiments were performed on a 3 GHz Pentium IV, with 1 GB of memory, running Windows XP. The tensor construction and

processing is implemented in Matlab. All algorithms were implemented in C++ and their parameters were tuned according to the original papers. To evaluate the examined algorithms, we have chosen real data sets from two different Political blogs: Technorati and Wordpress.

Technorati: The data was gathered crawling the Technorati site (http://technorati.com/politics/). We used a snapshot of all users, items and tags publicly available at May 1, 2009. The number of users, items and tags is 1,237, 2,648, and 6,563, respectively.

Wordpress: The data for Wordpress was gathered during February 2009, partly crawling the Wordpress site (http://en.wordpress.com/tag/politics/). The number of users, items and tags is 1,142, 920, and 1,927, respectively.

Following the approach of [11] 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 Technorati data set there are 115 users, 276 items and 544 tags, whereas for the Wordpress data set there are 106 users, 216 items and 570 tags.

5.1 Experimental Protocol and Evaluation Metrics

For the post recommendations, all algorithms had the task to predict the posts of the users' postings in the test set. 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.

Based on the approach of [12, 10], a more realistic evaluation of recommendation should consider the division of posts of each test user into two sets: (i) the *past* posts of the test user and, (ii) the *future* posts of the test user. Therefore, for a test user we generate the recommendations based only on the posts in his past set. 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 for post recommendations, 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).

5.2 Influence of the Core Tensor Dimensions

We first conduct experiments to study the influence of the core tensor dimensions on the performance of our Tensor Reduction algorithm. If one dimension of the core tensor is fixed, we can find the recommendation accuracy varies as the other two dimensions change, as shown in Figure 13. The vertical axes denotes the precision and the other two axes denote the corresponding dimensions. For the leftmost figure, the tag dimension is fixed at 202 and the other two dimensions change. For the middle figure, the post dimension is fixed at 103. For the rightmost figure, the user dimension is fixed at 61.

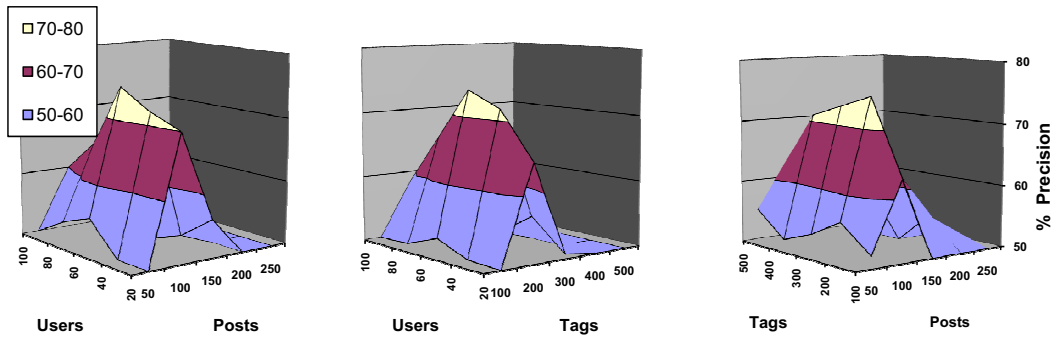


Figure 13: Precision of Tensor Reduction as dimensions of core tensor vary for Technorati data set.

Our experimental results indicate that a 70% of the original diagonal of $S^{(1)}$, $S^{(2)}$, $S^{(3)}$ matrices can give good approximations of A_1, A_2 and A_3 matrices. Thus, 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 61, 103 and 202 for the Technorati data set, whereas are set to 41, 83 and 198 for the Wordpress data set.

5.3 Algorithms' Settings

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

Fusion algorithm: We have varied the λ parameter for 0 to 1 by an interval of 0.1 and the neighborhood k parameter from 10-150 by an interval of 10. We have found the best λ to be 0.3 and k to be 20.

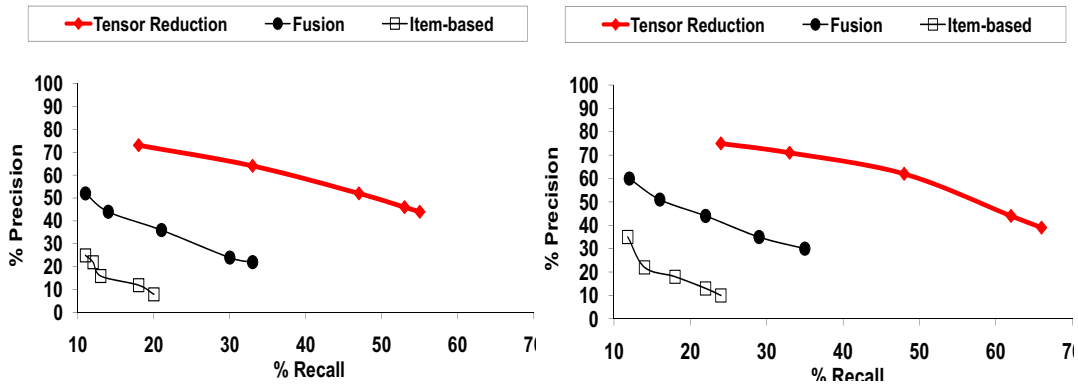
Item-based algorithm: We have varied the neighborhood k parameter from 10-300 by an interval of 10. We found the best k to be 40.

Tensor Reduction algorithm: Our tensor reduction algorithm is modified appropriately to recommend posts to a target user. In particular, our tensor represents a quadruplet $\{u, t, i, p\}$ where p is the likeliness that user u will tag item i with tag t.

5.4 Results

In this section, we proceed with the comparison of Tensor Reduction with Fusion, and Item-based, 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 post. In this situation, the recall and precision vary as we proceed with the examination of the top-N list.

For the Technorati data set (N is between [1..5]), in Figure 14a, we plot a precision versus recall curve for all three algorithms. As shown, all algorithms' precision falls as N increases. In contrast, as N increases, recall for all three algorithms increases too. Tensor Reduction algorithm attains 75% precision, when we recommend a top-1 list of tags. In contrast, Fusion gets a precision of almost 56%. 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, posts, tags), and through HOSVD, it addressed sparsity and finds latent associations. Item-based algorithm present the worst results, because they do not exploit all the existing information (they are applied in 2 dimensional data).



(a) (b)

Figure 14: Comparison of Tensor Reduction, Fusion, and Item-based algorithms for the (a) Technorati data set (b) Wordpress data set.

For the Wordpress data set (N is between [1..5]), in Figure 14b, we plot also a precision versus recall curve for all three algorithms. Tensor Reduction algorithm again attains the best performance. Thus, 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

Political blogs provide recommendations to users based on what tags other users have used on posts. However, the meaning of words in political blogsphere is characterized by different interpretations. In this paper, we provide a model to improve recommendations of posts to users. More specifically, we developed a framework to model the three types of entities that exist in a social tagging system: users, posts, and tags. We examined multi-way analysis on data modelled as 3-order tensor, to reveal the latent semantic associations between users, posts, and tags. The multi-way latent semantic analysis and dimensionality reduction is performed by combining the Higher Order Singular Value Decomposition (HOSVD). Our approach improves recommendations by capturing users multimodal perception of post/tag/user. We also performed experimental comparison of the proposed method against state-of-the-art recommendations algorithms, with two real data sets (Wordpress and Technorati). Our results show significant improvements in terms of effectiveness measured through recall/precision.

As future work, we intend to examine different methods for extending SVD to high-order tensors such as the Parallel Factor Analysis. We also indent 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.

References

[1] E. Acar and B. Yener. Unsupervised multiway data analysis: A literature survey. *IEEE Transactions on Knowledge and Data Engineering*, (to appear), 2008.

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

[3] 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.

[4] S. Chen, F. Wang, and C. Zhang. Simultaneous heterogeneous data clustering based on higher order relationships. In *Workshop on Mining Graphs and Complex Structures (MGCS07), in conjunction with ICDM2007*, pages 387–392, 2007.

[5] K. Durant and M. Smith. Mining sentiment classification from political web logs. In *Workshop on knowledge Discovery from the Web (WebKDD 06)*, pages 150–159, 2006.

[6] 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.

[7] S. Golder and B. Huberman. The structure of collaborative tagging systems. In *Technical Report*, 2005.

[8] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 211–220, 2007.

[9] 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.

[10] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. on Information Systems*, 22(1):5–53, 2004.

[11] 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.

[12] 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.

[13] R. Jaschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007*, pages 506–514.

[14] J. Johnson, K. Kaye, L. Bichard, and J. Wong. Every blog has its day: Politically-interested internet users' perceptions of blog credibility. *Journal of Computer-Mediated Communication*, 13(1), 2007.

[15] D. Karpf. Understanding blogspace. *Journal of Information Technology and Politics*, 5(4):369–384, 2008.

[16] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proc. ACM CIKM Conf.*, pages 247–254, 2001.

[17] L. Kenix. Blogs as alternative. *Journal of Computer-Mediated Communication*, 14(4):790–822, 2009.

[18] T. Kolda and S. J. Scalable tensor decompositions for multi-aspect data mining. In *IEEE International Conference on Data Mining (ICDM2008).*

[19] 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.

[20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the WWW Conference*, pages 285–295, 2001.

[21] 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.

[22] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proc. KDD Conf.*, pages 374–383, 2006.

[23] P. Symeonidis. User recommendations based on tensor dimensionality reduction. In *Proc. of the 5th IFIP Conference on Artificial Intelligence Applications and Innovations*, pages 331–340, 2009.

[24] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *2nd ACM Conference in Recommender Systems*, pages 43–50, 2008.

[25] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Transactions on Knowledge and Data Engineering*, 22(2), 2010.

[26] P. Symeonidis, M. Ruxanda, A. Nanopoulos, and Y. Manolopoulos. Ternary semantic analysis of social tags for personalized music recommendation. In *Proc. of the 9th International Symposium on Music Information Retrieval*, pages 219–224, 2008.

[27] K. Tso-Sutter, B. Marinho, and L. Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. *Proc. SAC Conf.*, 2008.

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