

User Recommendations based on Tensor Dimensionality Reduction

Panagiotis Symeonidis

Abstract Social Tagging is the process by which many users add metadata in the form of keywords, to annotate and categorize items (songs, pictures, web links, products etc.). Social tagging systems (STSs) can recommend users with common social interest based on common tags on similar items. However, users may have different interests for an item, and items may have multiple facets. In contrast to the current recommendation algorithms, our approach develops a model to capture the three types of entities that exist in a social tagging system: users, items, and tags. These data are represented by a 3-order tensor, on which latent semantic analysis and dimensionality reduction is performed using the Higher Order Singular Value Decomposition (HOSVD) method. We perform experimental comparison of the proposed method against a baseline user recommendation algorithm with a real data set (BibSonomy), attaining significant improvements.

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, music recommender systems such as Last.fm and MyStrands allow users to tag artist, songs, or albums. In e-commerce sites such as Amazon, users tag products to easily discover common interests with other users. Moreover, social media sites, such as Flickr and YouTube use tags for annotating their content. All these systems can further exploit these social tags to improve the search mechanisms and personalized recommendations. Social tags carry

Aristotle University, Department of Informatics, Thessaloniki 54124, Greece
e-mail: symeon@csd.auth.gr

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.

Several social tagging systems (STSs), e.g., Last.fm, Amazon, etc., recommend *interesting* users to a target user, opting in connecting people with common interests and encouraging people to contribute and share more contents. With the term *interesting* users, we mean those users who have similar profile with the target user. If a set of tags are frequently used by many users, then these users spontaneously form a community of interest, even though they may not have any physical or online connections. The tags represent the commonly interested web contents to this community of common interest. For example, Amazon recommends to a user who used a specific tag, other new users considering them as *interesting* ones. Amazon ranks them based on how frequently they used the specific tag.

In this paper, we develop a model based on 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) [4]. 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, item, and tag) that exist in social sites.
- We apply dimensionality reduction (HOSVD) in 3-order tensors, to reveal the latent semantic associations between users, items, and tags.
- For the first time to our knowledge we recommend interesting users to other users.

The rest of this paper is organized as follows. Section 2 summarizes the related work. The proposed approach is described in Section 3. Experimental results are given in Section 4. Finally, Section 5 concludes this paper.

2 Related work

In the area of discovering shared interests in social networks there are two kinds of existing approaches [5]. One is user-centric, which focuses on detecting social interests based on the on-line connections among users; the other is object-centric, which detects common interests based on the common objects fetched by users in a social community. In the user-centric approach, recently Ali-Hasan and Adamic [1] analyzed user’s online connections to discover users with particular interests for a given user. Different from this kind

of approaches, we aim to find the people who share the same interest no matter whether they are connected by a social graph or not. In the object-centric approach, recently Guo et al. [6] explored the common interests among users based on the common items they fetched in peer-to-peer networks. However, they cannot differentiate the various social interests on the same items, due to the fact that users may have different interests for an information item and an item may have multiple facets. In contrast, our approach focuses on directly detecting social interests and recommending users by taking advantage of social tagging, by utilizing users' tags.

Differently from existing approaches, our method develops a unified framework to concurrently model the three dimensions. Usage data are represented by a 3-order tensor, on which latent semantic analysis is performed using the Higher Order Singular Value Decomposition (HOSVD), which has been introduced in [4].

HOSVD is a generalization of singular value decomposition and has been successfully applied in several areas. In particular, Wang and Ahuja [8] 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. [2] 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 [7] 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. In the next section, we provide more information on HOSVD.

3 The Tensor Reduction Algorithm

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

The high-order singular value decomposition [4] 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 [4]:

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

where A_1, A_2, A_3 are called the 1-mode, 2-mode, 3-mode matrix unfoldings of \mathcal{A} , respectively.

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:

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

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

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 5 steps, as follows.

3.1 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 .

3.2 Matrix unfolding of tensor \mathcal{A}

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 .

Then, we apply SVD on the three matrix unfoldings A_1, A_2, A_3 . We result to 3 new matrices which contain $U^{(1)}, U^{(2)}, U^{(3)}$ left-singular vectors of A_1, A_2, A_3 matrices, respectively. After decomposition, we also result to $S^{(1)}, S^{(2)}, S^{(3)}$ matrices which contain the singular values of A_1, A_2, A_3 , respectively (to ease presentation, we omit the corresponding matrices with the right-singular vectors).

3.3 The core tensor \mathcal{S} construction

The core tensor \mathcal{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 \mathcal{S} , as follows:

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

where c_1 , c_2 , and c_3 are parameters that are chosen by preserving a percentage of information of the original $S^{(1)}$, $S^{(2)}$, $S^{(3)}$ matrices after appropriate tuning.

3.4 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:

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

3.5 The generation of the 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 item i with tag t . On this basis, users can be recommended to u according to their weights associated with $\{i, t\}$ pair.

4 Experimental Performance

In this section, in the area of user recommendations, we present experimental results for the performance of our approach against a baseline algorithm (BL). To evaluate the examined algorithms, we have chosen a real data set from BibSonomy which has been used as benchmarks in past works [3]. We used a snapshot of all users, items (both publication references and bookmarks) and tags publicly available at April 30, 2007. The number of users, items and tags is 1,037, 28,648, and 86,563, respectively.

To evaluate the effectiveness of Tensor Reduction and BL algorithms in recommending *interesting* users, we compute the item similarity within the recommended users [5]. This evaluation is based on the fact that users with shared interests are very likely to tag similar items.

A metric to evaluate this characteristic of each Neighborhood N of recommended users is to compute the average cosine similarity of all item pairs inside the Neighborhood of users with common social interest [5]:

$$ACS_N = \frac{\sum_{u,v \in N} \left[\sum_{i \in I(u), j \in I(v)} sim(i, j) \right]}{\sum_{u,v \in N} |I(u)||I(v)|}, \quad (4)$$

where for a user u , $I(u)$ denotes the items tagged by u . ACS_N evaluates the tightness or looseness of each Neighborhood or recommended users.

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

- **Baseline algorithm (BL):** BL algorithm is quite similar to Amazon.com’s method to recommend *interesting* users to a target user. BL logic is as follows: if a user uses a specific tag for item search, then he is recommended (except of recommended items) also *interesting* users, whose profiles are considered similar to him. These recommended users must have used the specific tag and are ranked based on how many times they used it. The basic idea behind this simple algorithm is that a tag corresponds in a topic of common interest. Thus, users that use the same tag could be interested in a common topic, forming a community of common interest.
- **Tensor Reduction algorithm:** Our tensor reduction algorithm is modified appropriately to recommend Neighborhoods of users to a target user. In particular, our tensor represents a quadruplet $\{t, i, u, p\}$ where p is the likeliness that tag t will be used to label item i by the user u . Therefore, new users can be recommended for a tag t , according to their total weight, which results by aggregating all items, which are labelled with the same tag by the target user.

4.1 Results

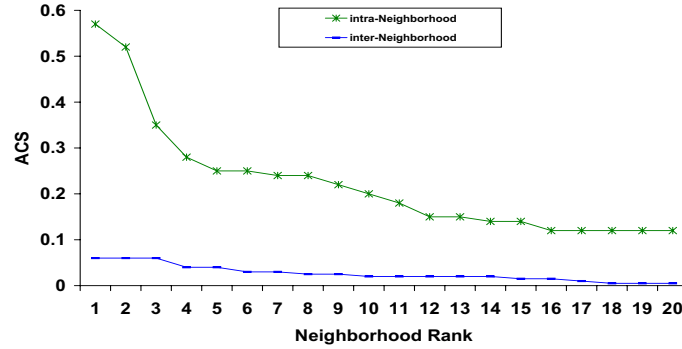
In this section, we evaluate the effectiveness of Tensor Reduction and BL algorithms in recommending *interesting* users. We compute the item similarity within the recommended neighborhoods of users. Note that as already described, some of the recommended neighborhoods can be consisted of users that are quite related, while others are consisted of users that are less related.

In the BibSonomy data set, users tag web sites. Thus, we crawled for each web site the first page and preprocess it to create a vector of terms. Preprocessing involved the removal of stop words, stemming and TF/IDF. Then, we find correlation between two web sites based on the keyword terms

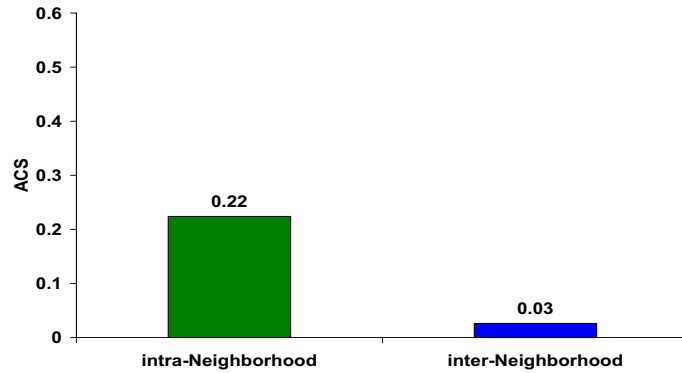
they include. We compute the similarity between two web sites with the inner product, i.e., the cosine similarity of their TF/IDF keyword term vectors [5].

For each user's neighborhood, we compute the Average Cosine Similarity (ACS) of all web site pairs inside the neighborhood, called intra-Neighborhood similarity. We also randomly select 20 neighborhood pairs among the 105 user neighborhoods and compute the average pairwise web site similarity between every two neighborhoods, called inter-Neighborhood similarity. In this figure, x axis is the rank of neighborhoods similarity, sorted by the descending order of their intra-Neighborhood similarities. y -axis shows the intra-Neighborhood similarity of each neighborhood and the corresponding average inter-Neighborhood similarity of this neighborhood with other 20 randomly selected neighborhoods.

Figure 1a, shows the comparison between the intra-Neighborhood and the inter-Neighborhood similarity of our Tensor Reduction Algorithm.



(a)

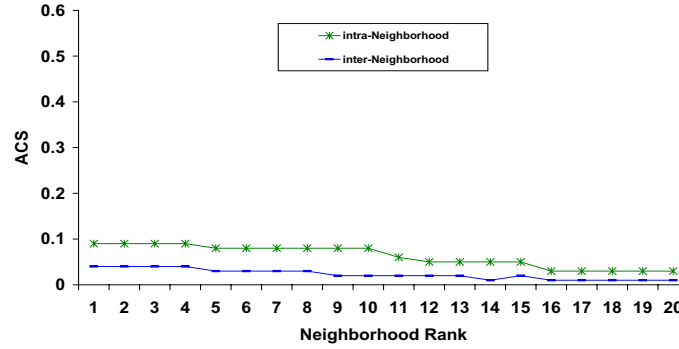


(b)

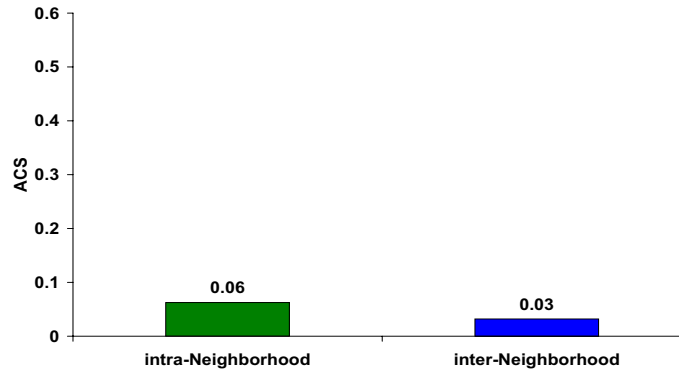
Fig. 1 Comparison of intra-Neighborhood and inter-Neighborhood similarity of Tensor Reduction Algorithm for the BibSonomy data set.

In this figure, x axis is the rank of neighborhoods similarity, sorted by the descending order of their intra-Neighborhood similarities. y -axis shows the intra-Neighborhood similarity of each neighborhood and the corresponding average inter-Neighborhood similarity of this neighborhood with other 20 randomly selected neighborhoods. As we can see, for all users' neighborhoods, the intra-Neighborhood similarity is consistently and significantly higher than the average inter-Neighborhood with others neighborhoods. As also shown in Figure 1b, the average intra-Neighborhood similarity across all neighborhoods is 0.22, while the average of inter-Neighborhood similarities across all neighborhood pairs is only 0.03.

Corresponding to Figures 1a and b, we show for the BL algorithm the comparison of intra- and inter-Neighborhood similarity for each neighborhood, and the average intra- and inter-Neighborhood similarity for all neighborhoods in Figures 2a and b, respectively.



(a)



(b)

Fig. 2 Comparison of intra-Neighborhood and inter-Neighborhood similarity of BL Algorithm for the BibSonomy data set.

As we can see, BL’s intra- and inter-Neighborhood similarity values are very close. This means, that BL fails to recommend coherent and related neighborhoods of users. In addition, our Tensor Reduction algorithm attains at least 3 times higher Average Cosine Similarity (ACS) than BL. That is, our approach recommends neighborhoods of users that are more related, while BL recommends users which are less relevant.

5 Conclusions

STSs provide recommendations to users based on what tags other users have used on items. 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. 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) method. Our approach improves user recommendations by capturing users multimodal perception of item/tag. Moreover, for the first time to our knowledge, we provide user recommendations. We also performed experimental comparison of the proposed method against a baseline user recommendation algorithm. Our results show significant improvements in terms of effectiveness measured through, intra- and inter-neighborhood similarity.

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.

References

1. N. Ali-Hasan and A. Adamic. Expressing social relationships on the blog through links and comments. In *Proc. ICWSM Conf.*, 2007.
2. 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.
3. 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.
4. 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.
5. X. Li, L. Guo, and Y. Zhao. Tag-based social interest discovery. In *Proc. ACM WWW Conf.*, 2008.
6. K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proc. INFOCOM Conf.*, 2003.
7. 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.
8. H. Wang and N. Ahuja. A tensor approximation approach to dimensionality reduction. *International Journal of Computer Vision*, 2007.