The TFC Model: *T*ensor *F*actorization and Tag *C*lustering for Item Recommendation in Social Tagging Systems

Dimitrios Rafailidis and Petros Daras, *Member, IEEE*

***Abstract*—In this paper, a novel Tensor Factorization and tag Clustering (TFC) model is presented for item recommendation in social tagging systems. The TFC model consists of three distinctive steps, in each of which important innovative elements are proposed. More specifically, through its first step, the content information is exploited to propagate tags between conceptual similar items based on a relevance feedback mechanism, in order to solve sparsity and "cold start" problems. Through its second step, sparsity is further handled, by generating tag clusters and revealing topics, following an innovative $tf \cdot idf$ weighting scheme. Furthermore, we experimentally prove that a few number of expert tags can improve the performance of quality recommendations, since they contribute to more coherent tag clusters. Through its third step, the latent associations among users, topics and items are revealed by exploiting the Tensor Factorization technique of High Order Singular Value Decomposition (HOSVD). This way the proposed TFC model tackles problems of real world applications, which produce noise and decrease the quality of recommendations. In our experiments with real world social data, we show that the proposed TFC model outperforms other state-of-the-art methods, which also exploit the Tensor Factorization technique of HOSVD.**

***Index Terms*—Social tagging, recommender systems, relevance feedback, content based information retrieval, expert tagging.**

I. Introduction

Social tagging [11], is a process that allows users to annotate a set of items like photos (Flickr [17]), songs (last.Fm [18]) or web sites (del.icio.us [16]), in order to facilitate their sharing, discovery and retrieval. These annotations are in the form of free keywords also known as social tags, through which users can express their personal opinion to describe items. This is very important, since the complex multifaced information of items like images, videos, etc. can be exploited and social tagging systems are able to generate personalized recommendations, by allowing users to pose tags as queries. In particular, the collaborative-based mechanism of such systems functions as follows: users having the same tagging behavior tend to get similar recommendations. Therefore, such recommender systems are often characterized as "item collaborative filtering in social tagging systems" [35].

Despite the benefits of social tagging, several important problems produce noise [51] and reduce the recommendation accuracy. These problems are the following:

- Due the free nature of tags, polysemy and synonymy are very common problems, since tags are subject to multiple interpretations.
- "Learning tag relevance" [27], [28], [49], is the problem of how to interpret the relevance of a user-contributed tag with respect to the visual content the tag is describing.
- The "cold start" problem [14] refers to the fact that users participate rarely in the tagging process and thus there are only a few tags on which to base the recommendation.
- Sparsity in social tagging systems affects the recommendation accuracy. Specifically, since item recommendation in social tagging is collaborative-based, high accuracy is achieved only if users annotate the same items with similar tags (users having the same tagging behavior). However, this is extremely difficult in the case of real world applications. A huge amount of social tagging data is generated with users having similar taste on topics and not on exact items and therefore the quality of recommendations is decreased.
- With the introduction of tags, the usual binary relation between users and items turns into a ternary relation between users, items and tags. This ternary relation is mapped to a tripartite network [12], [24], [46], [54], which should be considered by recommender systems, in order to capture the latent associations among users, tags and items.

A. Background and Motivation

According to [35], expert tagging is a possible solution for the aforementioned problems, which usually relies on a small number of domain experts, who annotate resources based on structured vocabularies. The main advantages of using experts' opinions are: (a) the resulting well agreed tag vocabulary and (b) the accurate annotations. However, the disadvantages are (a) the time needed for the manual annotation, and (b) the limited vocabulary that must be used.

In order to capture the ternary relation among users, tags and items in social tagging systems, previous works like [41], [50], [55], focused on generating recommendations based on tensor factorization (TF) techniques [6], [22], [25], [53]. Such methods are able to (a) solve problems like polysemy and synonymity, (b) preserve the ternary relation, (c) reveal the latent associations among users, tags and items, (d) reduce the noise in social tagging systems, (e) provide more accurate recommendation compared to methods that suppress the 3-way

D. Rafailidis and P. Daras are with the Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece. e-mail {drafail,daras}@iti.gr

Manuscript received; revised

relationship to 2-way like the one in [52] and (f) generate efficient recommendations in terms of run times, since the recommendations only depend on the smaller factorization dimensions after applying the TF method [35]. The aforementioned works do not solve sparsity and the cold start problem, thus, in [38] content information is exploited by performing tag propagation in similar items. However, the main disadvantage of the method in [38], is that the proposed tag propagation cannot be uncontrolled. More precisely, by allowing extensive tag propagation, the noise that incurs may affect the quality of recommendations due to the irrelevant tag assignments to items. In order to constrain the amount of propagated tags, a threshold parameter is controlled, based on the items similarity. However, the fundamental problem in this approach is the "learning tag relevance", since tag propagation between similar items can be performed only if they belong to the same concept. Simultaneously, a lot of work has been conducted concerning relevance feedback recommender systems [15], [20], [29]. Such approaches identify items that belong to the same concept. Nevertheless, the tag dimension is always omitted.

Additionally, since item recommendation in social tagging systems is collaborative filtering-based, the quality of recommendations is directly affected by sparsity. This is due to the fact that high recommendation accuracies are achieved, under the condition that users have the same tagging behavior. However, in the case of real world applications users have similar taste upon topics. Regarding methods for revealing topics and generating recommendations in social tagging systems, several works have been presented. Such methods aim at tag clustering, in which tags are clustered to reveal a topic [9], [10], [32] [37], [40], [47], [57]. However, these methods do not handle the problems of sparsity and "cold start", which result in inaccurate recommendations.

B. Contribution and Layout

In this paper, the *T*ensor *F*actorization and tag *C*lustering (TFC) model is presented for item recommendation in social tagging systems. The TFC model consists of three steps and through each step the aforementioned problems are successfully handled. The first step of the proposed approach involves tag propagation by exploiting content, so as to face the issues of sparsity, "cold start" and "learning tag relevance". The proposed step is based on a relevance feedback mechanism, in order to perform tag propagation between similar items only if they belong to the same concept. Consequently, the TFC model is able to propagate less noisy tags, since tags of annotated items are propagated only to relevant items. The TFC model is validated in an image collection and as we experimentally prove, the method of [38] is not sufficient for image recommendation based on social tags, since the issue of "learning tag relevance" is ignored. To the best of our knowledge we are the first who exploit relevance feedback mechanisms in TF, which increase the accuracy of recommendations.

The second step of the TFC model is tag clustering in order to reveal topics and identify the taste of users in these topics. By doing so, the sparsity problem is handled, by transforming tags to tag clusters, in order to further increase the accuracy of recommendations. We evaluate our proposed method by performing two different tag clustering algorithms. The first one presented in [32] takes into account the tripartite network and the second one is an adapted version of K-Means in the social tagging systems for tag clustering. Through exhaustive experimental evaluation we conclude to the optimal number of tag clusters - topics for both tag clustering methods, without considering the information of image classification, as authors in [32] do. After producing tag clusters, an innovative $tf \cdot idf$ weighting scheme is followed to calculate users' interests and identify image relations to topics (tag clusters). Additionally, to generate even more coherent tag clusters and consequently better recommendations, we exploit tagging of few experts, by considering the fact that finding them is a time consuming process. We show experimentally that the proposed tag clustering step of the TFC model alleviates the performance of collaborative filtering based TF methods, since the discovered topics form the basis of recommendation. The personalized interest in topics is captured by the proposed $tf \cdot idf$ weighting scheme.

The third step of the proposed method is based on the TF technique called High Order Singular Value Decomposition (HOSVD) [25]. We show that the cubic complexity of HOSVD is minimized by reducing the number of tags to tag clusters, by also preserving the benefits of TF techniques. The innovation in this step is that, by exploiting HOSVD the latent associations among users, topics and images are revealed. In our experiments we evaluate the TFC model against the TF model presented in [50], and the TF model in [38], which exploits content to solve sparsity.

The rest of the paper is organized as follows, after summarizing related work to the proposed model in Section II, we describe the proposed TFC model in Section III. Our experimental results on real-world social tagging data from our image collection in Section IV provide evidence about the effectiveness of the proposed approach. Finally, we draw the basic conclusions of our study in Section V.

II. Related Work

The related works are divided into four parts according to the contributions of this paper: (a) recommender systems based on relevance feedback, (b) collaborative filtering in social tagging systems, (c) tag clustering methods for revealing topics and generating recommendations and (d) approaches exploiting expert tagging.

A. Recommender systems based on Relevance Feedback

Relevance feedback approaches have been used in several recommender systems, such as YourNews [1], Fab [2] and NewT [48]. Such systems try to recommend items similar to those a given user used in the past. In particular, the basic process of recommendation includes matching of the attributes of a user's profile in which preferences and interests are stored, in order to recommend to the user new interesting items [30]. The main difference between relevance feedback-based recommender systems and collaborative-based recommender systems in social tagging, lies in the fact that relevance feedback-based recommendation systems try to recommend items similar to those a given user used in the past, by storing the information of interest through the relevance feedback mechanism, whereas systems designed according to the collaborative recommendation paradigm identify users whose preferences are similar to those of the given user and recommend items they have liked. Additionally, recommender systems based on relevance feedback omit the tag dimension, thus, the complex multifaced information of the items cannot be exploited and as a result the quality of the recommendations is decreased.

B. Collaborative Filtering in Social Tagging Systems

In [31], the iFind model was proposed, which exploits the semantic contents of images in addition to the extracted low-level features. However, the user dimension in the tripartite social network was omitted along with the respective personalized recommendation. Tso-Sutter et al. [52] examined the generic problem of item recommendation in collaborative tagging systems. They proposed a generic method that allows tags to be incorporated to traditional two-way recommender algorithms, by reducing the three-way correlations to two-way and then, they applied a fusion method to reassociate these correlations. The widely used technique of Latent Semantic Analysis (LSA) [8] has been used as a solution for addressing the problems of synonymity, polysemy, and noise in social tags [26]. LSA reveals latent structures in the data by using matrix factorization techniques, based on the Item-based algorithm [44]. Such techniques became a popular choice for implementing collaborative filtering in social tagging systems and a survey can be found in [23]. Nevertheless, the matrix factorization techniques ignore the three-way correlations between users, items and tags, which reveal the personalized perception of users about particular items.

To address the requirements of collaborative tagging systems such as the ternary relation, new approaches have been proposed. Regarding methods that model the social tagging data with tensors, Xu et al. [55] proposed a method that recommends tags by using ternary analysis and Symeonidis et al. [50] exploited HOSVD either for item or tag recommendation. In [41], authors investigated the ternary association among users, items and tags for more effective tag recommendation using the tensor decomposition technique called Pairwise Interaction Tensor Factorization (PITF). Their method is extremely promising, since they reduced the cubic runtime of tensor factorization to linear and is applicable for midsized and large data sets. The PITF technique explicitly models the pairwise interactions between users, items and tags to provide more accurate tag recommendations. However, it cannot be readily adapted to item recommendation due to the different nature of tag recommendation and item recommendation. Tag recommendation aims at predicting the use of tags of a given user on a given item, with two entities predefined, whereas item recommendation aims at predicting the "satisfaction" on items, with only the user specified. Generally speaking, item recommendation is more challenging than tag recommendation, because less information is known about the subject to receive the recommendations.

C. Tag Clustering for Generating Recommendations

Several tag clustering methods have been proposed in the literature for recommender systems. Clustering is an offline step that is performed independently from the personalized recommendation algorithm. The discovered tag clusters form the basis of the recommendation algorithm. In [3], a tag graph was built based on the co-occurrence of tags in annotated items. A spectral bisection method was adopted to cluster the tag graph. The identified tag clusters were used for finding semantically related tags. In [9], [10], several clustering techniques were evaluated such as maximal complete link, K-Means, and hierarchical agglomerative clustering. In [47], Shepitsen et al. proposed a recommendation system based on hierarchical clustering of the tag space, using user profiles and tag clusters to personalize the recommender results. Zhou et al. [57] proposed an extended language model based on Latent Dirichlet Allocation (LDA) for information retrieval. This model incorporated the topical background of documents and social tags, as well as users' domain interests. In [40], authors incorporated social tags into two clustering methods: K-Means and a generative clustering method based on LDA. Although their work proved the value of tags as an additional information source for clustering, the user dimension was omitted.

In [37], a spectral clustering method was presented for capturing the three dimensions in social tagging data and combining multiple values of similarity to get groups of related items, in order to provide recommendations. In [32], the tripartite network was exploited to generate user, item and tag clusters simultaneously. Yet, in both works tag propagation based on content was not exploited and therefore sparsity and "cold start" problems remained unsolved.

D. Expert Tagging

In [7], authors attempted to find experts in enterprises. In other approaches [4], [21] users were automatically classified as experts according to their preferences. For all the aforementioned works the ternary relation in the social tagging system was omitted and the quality of recommendations was reduced. According to [35], expert tagging usually relies on a small number of domain experts, who annotate items based on structured vocabularies. Experts provide tags that are objective and cover multiple aspects. Pandora [19] is a notable example of a system that uses experts for tagging music items. The main advantage of using experts is the resulting well agreed tag vocabulary. However, the main disadvantages are the effort needed for finding experts, the time consumption for performing the tagging and the limited vocabulary that must be used.

III. The Proposed TFC Model

A. Overview and Notation

In this Section we provide a detailed description of the proposed TFC model. The social tagging data are in the form of triplets $\langle U_i, T_j, I_k, w\rangle$, where the weight w corresponds to the likelihood that user U_i will tag item I_k with tag T_j. We denote the set of the triplets as $\mathbf{Y}$ (in the rest of the paper sets or lists are denoted by bold letters). The input of the TFC model is the set $\mathbf{Y}$, with all weights w initially set to 1. The TFC model consists of three steps and for each step the weight w per triplet is recalculated. After the TFC model is built, we recommend N items to user U_i for a query tag T_j according to the weights in the triplets that contain U_i and T_j. The three steps of the TFC model are: (a) tag propagation based on relevance feedback, which results in a superset $\mathbf{Y}^+$, with $\mathbf{Y} \subseteq \mathbf{Y}^+$, (b) tag clustering by transforming the set $\mathbf{Y}^+$ to set $\mathbf{X}$, following the proposed $tf \cdot idf$ weighting scheme and (c) triplets in set $\mathbf{X}$ are modeled to tensor[1] $\mathcal{A}$ and then follows the tensor factorization of $\mathcal{A}$ to its low rank approximation $\hat{\mathcal{A}}$ by exploiting HOSVD, in order to reveal the latent associations among users, tag clusters and images.

B. Tag Propagation based on Relevance Feedback

The first step of the TFC model includes tag propagation based on a relevance feedback mechanism, in order to ensure that tags will be propagated between similar images only if they belong to the same concept. Firstly, a content-based descriptor vector $\overrightarrow{DV}$ is extracted from each image based on [33], called "Color and Edge Directivity Descriptor" (CEDD). In particular, CEDD incorporates the color and texture information of images in a histogram, as follows: firstly, the image is separated in a preset number of blocks. In order to extract the color information, a set of fuzzy rules undertakes the extraction of a Fuzzy-Linking histogram [5], while the latter stems from the HSV color space. Furthermore, following the methodology of CEDD, twenty rules are applied to a three-input fuzzy system in order to generate a 10-bin quantized histogram, where each bin corresponds to a preset color. The number of blocks assigned to each bin is stored in a feature vector. Then, four extra rules are applied to a two input fuzzy system, in order to change the 10-bin histogram into a 24-bin histogram, importing thus information related to the hue of each color. Next, the five digital filters that were proposed in the MPEG-7 Edge Histogram Descriptor [34] are also used for exporting the information which are related to the texture of the image, classifying each image block in one or more of the six texture regions, shaping thus the 144-bin histogram. Therefore, for each image, CEDD extracts a descriptor vector $\overrightarrow{DV}$ of 144 quantized attribute values (dimensions), which are integers ranging from 0 to 7. One of the most important attributes of CEDD is the low computational power needed for its extraction, in comparison with the needs of the most MPEG-7 descriptors. However, also other well known descriptors with similar characteristics like SIFT, SURF, etc. could be used. Finally, the similarities between images are calculated based on the $L2$ distance of their extracted descriptor vectors $\overrightarrow{DV}$.

[1]A tensor is a multi-dimensional matrix. An N-order tensor $\mathcal{A}$ is denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \ldots I_N}$, with elements $a_{i_1,\ldots,i_N}$. In this paper, only 3-order tensors are used.

Next, the relevance feedback mechanism is activated. Given a query image Q, a list $\mathbf{R}_Q$ of images is retrieved. The aim is to refine the query descriptor vector $\overrightarrow{DV_Q}$, so as to retrieve more positive (relative to the same concept) images in $\mathbf{R}_Q$. Images in $\mathbf{R}_Q$ are examined based on the class of Q. If an image in $\mathbf{R}_Q$ belongs to the same class as Q, then it is marked as positive, otherwise it is marked negative. Then, according to Rocchios formula [43] the query's descriptor vector is refined as follows:

$$\overrightarrow{DV_{Q'}} = \overrightarrow{DV_Q} + \lambda_p \sum_{i \in Pos} \overrightarrow{DV_i} - \lambda_n \sum_{j \in Neg} \overrightarrow{DV_j} \qquad (1)$$

where, λ_p and λ_n are control parameters that allow to set the relative importance to the query image of all positive and negative images in $\mathbf{R}_Q$. We set $\lambda_p = 1.8$, to retrieve the maximum number of positive marked images and $\lambda_n = 0$, since users usually ignore negative results, while focusing only on the positive ones. The refinement of Q is an iterative process and it terminates when $\mathbf{R}_Q$ remains the same.

Next, all images in the initial set $\mathbf{Y}$ of our social data are posed as queries. This way, we retrieve for each image Q a set of positive marked images, denoted by $\mathbf{I}^+$, with $\mathbf{I}^+ \subseteq \mathbf{R}_Q$. The tag propagation process for an image query Q and the resulting set $\mathbf{I}^+$ is performed according to the following equation:

$$\text{if } \langle U, T, Q, 1\rangle \in \mathbf{Y} \text{ and } \mathbf{I}^+ \subseteq \mathbf{R}_Q \text{ then}$$
$$\forall\ I \in \mathbf{I}^+ \longrightarrow \mathbf{Y}^+ \equiv \mathbf{Y} \cup \langle U, T, I, \text{sim}(I, Q)\rangle \qquad (2)$$

Thus, the initial data set $\mathbf{Y}$ is transformed to $\mathbf{Y}^+$, by propagating triplets in the form $\langle U, T, I, \text{sim}(I, Q)\rangle$, where for each user U the association between tag T and image I is calculated according to the respective weight $\text{sim}(I, Q)$. Therefore, tag propagation based on the relevance feedback mechanism is performed in a more efficient way compared to [38], where image results include both relevant and irrelevant images, ignoring the "learning tag relevance" issue. Additionally, the amount of the propagated tags is performed automatically, without using the threshold parameter of the items similarity.

At this point we must mention that except for the aforementioned relevance feedback mechanism, several methods have been proposed for image-based retrieval so far. Most of them include probabilistic networks [29], query point movement [20] and SVM techniques [15]. Nevertheless, at the current step of the TFC model we focus on the tag propagation process, which aims to capture the complex image similarity in a more efficient way, in order to (a) solve the sparsity and the "cold start" problems and (b) avoid propagating noisy tags by "learning tag relevance". Thus, experimental evaluation of relevance feedback mechanisms is out of the scope of this paper.

C. Tag Clustering and Social Data Regeneration

In the second step of the proposed TFC model, tag clustering is performed to reveal topics and consequently to further handle the problem of sparsity for collaborative filtering in social tagging, in order to increase the quality of recommendations. More precisely, the second step of the TFC model comprises: (a) tag clustering and (b) transformation of the social data set $\mathbf{Y}^+$ to data set $\mathbf{X}$, by modeling the generated tag clusters with their associated weights w for each quadruple $\in \mathbf{Y}^+$. In this transformation, an innovative $tf \cdot idf$ weighting scheme is followed, to ensure that the personal interest of users and the relation of items and tags to topics are taken into account.

For validating the impact of tag clustering on the proposed TFC model, two different clustering methods are presented, the Tripartite Clustering [32], which captures the ternary relation in the social tagging data, and an adaptation of the K-Means clustering. For both algorithms the corresponding distance functions are described, since they are built upon the same base, the K-Means algorithm. More specifically, at the beginning of each algorithm, tags (also users and images in the Tripartite Clustering) are randomly selected as centroids of the tag clusters. Next, through an iterative process tags are assigned to clusters based on the respective distance to centroids. Then, the centroids of the new clusters are recalculated. The iterations terminate until the centroids remain the same or the number of iterations exceed a maximum threshold. The main difference between the Tripartite Clustering and the adapted K-Means for tag clustering is that in the former the distances are calculated at each dimension space (dimensions corresponds to the different type of entities, i.e. users, images and tags).

1) Distance Function in Tripartite Clustering: The Tripartite Clustering [32] that produces user, image and tag clusters at the same time, is denoted by K_U, K_I and K_T, respectively. A brief description of the Tripartite Clustering follows. For each type of entity, user, image and tag, we calculate two link vectors. For each user U two link vectors are calculated, the tag link vector $\overrightarrow{TLV}^{(U)}$ and the image link vector $\overrightarrow{ILV}^{(U)}$. For each image I two link vectors are calculated, the user link vector $\overrightarrow{ULV}^{(I)}$ and the tag link vector $\overrightarrow{TLV}^{(I)}$, and for each tag T, the image link vector $\overrightarrow{ILV}^{(T)}$ and the user link vector $\overrightarrow{ULV}^{(T)}$, respectively. In particular, the value of the j-th position in the tag link vector $\overrightarrow{TLV}_i^{(U)}$ of user U_i, corresponds to the total numbers of times user U_i has assigned tag T_j to any image, which equals the term frequency value $tf(U_i, T_j)$. More formally,

$\forall$ user U_i its two link vectors are:

$$\overrightarrow{TLV}_i^{(U)} = tf(U_i, T_j) \text{ and } \overrightarrow{ILV}_i^{(U)} = tf(U_i, I_k)$$

$\forall$ image I_k its two link vectors are:

$$\overrightarrow{TLV}_k^{(I)} = tf(I_k, T_j) \text{ and } \overrightarrow{ULV}_k^{(I)} = tf(I_k, U_i)$$

$\forall$ tag T_j its two link vectors are:

$$\overrightarrow{ULV}_j^{(T)} = tf(T_j, U_i) \text{ and } \overrightarrow{ILV}_j^{(T)} = tf(T_j, I_k)$$

where $i \in 1, 2, \ldots, \mathcal{U}$, $j \in 1, 2, \ldots, \mathcal{T}$ and $k \in 1, 2, \ldots, \mathcal{I}$ and $\mathcal{U}$, $\mathcal{T}$, $\mathcal{I}$ are the total numbers of users, tags and images, respectively.

The distances between each type of entity are calculated as follows:

$$\begin{aligned} d(T_i, T_j) &= \alpha \times d(\overrightarrow{ILV}_i^{(T)}, \overrightarrow{ILV}_j^{(T)}) + \ldots \\ &\ldots + (1-\alpha) \times d(\overrightarrow{ULV}_i^{(T)}, \overrightarrow{ULV}_j^{(T)}) \end{aligned} \quad (3)$$

$$\begin{aligned} d(U_i, U_j) &= \beta \times d(\overrightarrow{ILV}_i^{(U)}, \overrightarrow{ILV}_j^{(U)}) + \ldots \\ &\ldots + (1-\beta) \times d(\overrightarrow{TLV}_i^{(U)}, \overrightarrow{TLV}_j^{(U)}) \end{aligned} \quad (4)$$

$$\begin{aligned} d(I_i, I_j) &= \gamma \times d(\overrightarrow{TLV}_i^{(I)}, \overrightarrow{TLV}_j^{(I)}) + \ldots \\ &\ldots + (1-\gamma) \times d(\overrightarrow{ULV}_i^{(I)}, \overrightarrow{ULV}_j^{(I)}) \end{aligned} \quad (5)$$

where greater value of α means that the distances between tags rely more on their item link vectors and less on their user link vectors; greater value of β means that the distances between users rely more on their item link vectors and less on their tag link vectors; and greater value of γ means that the distances between items rely more on their tag link vectors and less on their user link vectors. The value of the distance can be calculated based on various similarity measures such as those proposed in [36]. In our implementation the cosine distance is used between the link vector per type of entity at each dimension [32].

2) Distance Function in Adapted K-Means for Tag Clustering: In the adapted implementation of the K-Means tag clustering, where the ternary relation among users, images and tags are suppressed to images and tags (only the value $tf(T_i, I_k)$ is considered), the distances between tag T_i and T_j are calculated as follows:

$$d(T_i, T_j) = \frac{\sum_{\forall k \in I} tf(T_i, I_k) \times tf(T_j, I_k)}{\sqrt{\sum_{\forall k \in I} tf(T_i, I_k)^2} \times \sqrt{\sum_{\forall k \in I} tf(T_j, I_k)^2}} \quad (6)$$

3) The Proposed $tf \cdot idf$ Weighting Scheme: Both tag clustering algorithms produce K_T tag clusters, which correspond to K_T topics. Before regenerating the triplets $\mathbf{Y}^+$ to the social data set $\mathbf{X}$ in the form of tag clusters, we calculate user's U_i interest in tag cluster K_{T_j} and the relation of an image I_k to tag cluster K_{T_j}, by following the proposed $tf \cdot idf$ weighting scheme. More precisely, user's interest in tag cluster K_{T_j} is the ratio of $tf(U_i, T_j)$ over $tf(U_i)$, which is the ratio of the times that user U_i annotated an image with a tag assigned to cluster K_{T_j} over the total number of user's U_i annotations in all tag clusters. Similarly, the relation of an image I_k to tag cluster K_{T_j} is calculated as the ratio of $tf(I_k, T_j)$ over $tf(I_k)$, which is the ratio of the times that image I_k was annotated with a tag assigned to cluster K_{T_j} over the total number of times the image I_k was annotated. Each triplet $\langle U_i, T_j, I_k, w \rangle$ in the social data set $\mathbf{Y}^+$ is regenerated to $\langle U_i, K_{T_j}, I_k, w' \rangle$, where the weight w' is calculated according to the proposed $tf \cdot idf$ weighting scheme as:

$$w'(U_i, K_{T_j}, I_k) = \frac{tf(U_i, K_{T_j})}{tf(U_i)} \times \frac{tf(I_k, K_{T_j})}{tf(I_k)} \times \dots$$
$$\dots \times (1 - d(T_j, Centroid_{K_{T_j}})) \quad (7)$$

where $d(T_j, Centroid_{K_{T_j}})$ is the distance (correlation) of tag T_j from the centroid of tag cluster K_{T_j} (topic), given that tag T_j is assigned to tag cluster K_{T_j}.

4) Example of Tag Clustering and Data Set Regeneration: In the sequel, we explain in detail how the problem of sparsity affects the accuracy of collaborative filtering recommender systems in social tagging and how we handle it. We illustrate a running example of tag clustering in Figure 1 and the respective triplets are presented in their initial form of tags in Table I.

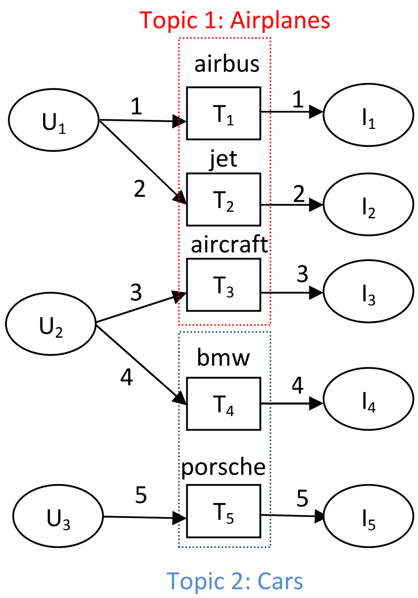


Fig. 1. Running example of tag clustering

TABLE I
TRIPLETS IN THEIR INITIAL FORM FROM THE RUNNING EXAMPLE IN FIGURE 1.

ID	User	Tag	Image	Weight
1	U_1	T_1	I_1	1
2	U_1	T_2	I_2	1
3	U_2	T_3	I_3	1
4	U_2	T_4	I_4	1
5	U_3	T_5	I_5	1

In the running example user U_2 has preferences to both topics, by posting tags to image I_3 and I_4. A typical collaborative-based social tagging system, will fail to recommend either item I_1 or I_5, because user U_2 has not posted common tags or has not annotated the same items with users U_1 and U_3. The solution is provided by clustering tags to two topics, since U_2 has common preferences to topics with U_1 and U_3 (topics Airplanes and Cars, respectively). In Table II we depict the result of the proposed regeneration process according to (7). In our experiments we show that the proposed social data regeneration based on tag clusters increase the accuracy of recommendations.

TABLE II
REGENERATE INITIAL TRIPLETS OF FIGURE 1 AND THEIR ASSOCIATED WEIGHT w IN THE FORM OF TAG CLUSTERS.

ID	User	Tag Cluster	Image	Weight
1	U_1	K_{T_1}	I_1	$w'(U_1, K_{T_1}, I_1)$
2	U_1	K_{T_1}	I_2	$w'(U_1, K_{T_1}, I_2)$
3	U_2	K_{T_1}	I_3	$w'(U_2, K_{T_1}, I_3)$
4	U_2	K_{T_2}	I_4	$w'(U_2, K_{T_2}, I_4)$
5	U_3	K_{T_2}	I_5	$w'(U_1, K_{T_2}, I_5)$

D. Tensor Factorization and Item Recommendation in TFC Model

The third step of the TFC model comprises (a) modeling the data set $\mathbf{X}$ in tensor $\mathcal{A}$ and (b) applying the tensor factorization method of HOSVD to produce the reconstructed tensor $\hat{\mathcal{A}}$, in order to reveal the latent associations among users, tag clusters and images. The final goal is to recommend images according to the detected latent associations in the reconstructed tensor $\hat{\mathcal{A}}$. The procedure of HOSVD is illustrated in Figure 2, where $I_1 = \mathcal{U}$, $I_2 = K_T$, $I_3 = \mathcal{I}$ are the user, tag cluster and image dimensions, respectively and S is the core tensor that captures the 3-way relations.

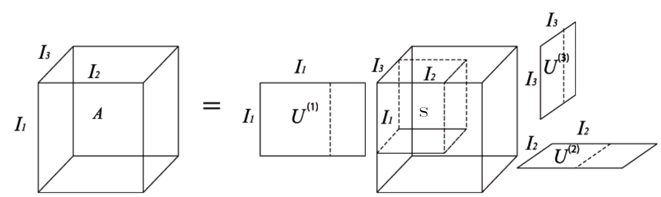


Fig. 2. Visualization of the result of HOSVD

1) Initial construction of tensor $\mathcal{A}$: Based on the social data set $\mathbf{X}$, we construct an initial 3-order tensor $\mathcal{A} \in \mathbb{R}^{U \times K_T \times I}$. The initial values assigned to each entry of $\mathcal{A}$ equals the pre-computed weights according to (7) in step 2 of the TFC model.

2) Matrix unfolding of tensor $\mathcal{A}$: Tensor $\mathcal{A}$ can be unfolded i.e., transformed to a two dimensional matrix, by arranging the corresponding fibers of $\mathcal{A}$ as columns of A_n ($1 \le n \le 3$) (further details can be found in [25]). In our approach, the initial tensor $\mathcal{A}$ is unfolded to all its three mode-dimensions. Thus, after unfolding $\mathcal{A}$, we create three new matrices A_1, A_2, A_3, as follows:

$$A_1 \in \mathbb{R}^{U \times K_T I}, \qquad A_2 \in \mathbb{R}^{K_T \times UI}, \qquad A_3 \in \mathbb{R}^{UK_T \times I}$$

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

$$A_n = U^{(n)} \cdot \Sigma^{(n)} \cdot (V^{(n)})^T, \quad 1 \le n \le 3 \quad (8)$$

To reveal latent associations and reduce noise, the dimensions of each array containing the left-singular vectors (i.e., matrices $U^{(1)}, U^{(2)}, U^{(3)}$) have to be reduced. Therefore, we maintain the dominant c_n left singular vectors in each $U^{(n)}$, $1 \le n \le 3$ matrix based on the corresponding singular values in $\Sigma^{(n)}$. The resulting matrix is denoted as $U_{c_n}^{(n)}$. The value of c_n parameters are usually chosen by preserving a percentage of information in $\Sigma^{(n)}$.

4) Construction of the core tensor $\mathcal{S}$: The core tensor S governs the interactions among the three examined modes. Its construction is implemented as:

$$\mathcal{S} = \mathcal{A} \times_1 \left(U_{c_1}^{(1)}\right)^T \times_2 \left(U_{c_2}^{(2)}\right)^T \times_3 \left(U_{c_3}^{(3)}\right)^T, \qquad (9)$$

where $\mathcal{A}$ is the initial tensor, $\left(U_{c_n}^{(n)}\right)^T$ is the transpose of $U_{c_n}^{(n)}$, $\times_n$ is the n-mode product of a third order tensor [25] and $\mathcal{S}$ is a $c_1 \times c_2 \times c_3$ tensor.

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

$$\hat{\mathcal{A}} = \mathcal{S} \times_1 U_{c_1}^{(1)} \times_2 U_{c_2}^{(2)} \times_3 U_{c_3}^{(3)} \qquad (10)$$

where $\hat{\mathcal{A}}$ is a tensor with the same size as $\mathcal{A}$. $\hat{\mathcal{A}}$ is a good approximation of $\mathcal{A}$ (also known as low rank approximation of $\mathcal{A}$), in the sense that the Frobenius norm $||\mathcal{A} - \hat{\mathcal{A}}||_F^2$ is small (element-wise squared differences [25]). Moreover, $\hat{\mathcal{A}}$ contains less noise and thus, the latent associations are revealed by keeping only a subset of the dominant left singular vectors.

6) The Generation of Item Recommendation: The elements of the reconstructed tensor $\hat{\mathcal{A}}$ represent the final triplets $\langle U_i, K_{T_j}, I_k, w\rangle$, where w corresponds to the likelihood that user U_i will tag item I_k with tag T_j, where tag T_j has been assigned to tag cluster K_{T_j} at step 2 of the TFC model. If N items have to be recommended to U_i queried with tag T_j, then the N items are selected that have the highest weights from triplets that contain both U_i and K_{T_j}.

IV. Experimental evaluation

A. Data Set

For evaluation purposes we prepared a real data set, created by researchers in 10 different European research institutes and universities. There are $\mathbf{Y}$ =48,564 triplets in the form user-image-tag with $\mathcal{U}$ =400 users, $\mathcal{I}$=1,023 images and $\mathcal{T}$ =11,779 tags. The images have been gathered from the same 10 research organizations and they were classified to 11 categories (Table III).

TABLE III
Number of images included in the 11 categories.

category	**Animals**	**Apparels**	**Buildings**	**Electronics**
# images	163	79	57	48
category	**Furniture**	**Toys**	**Tableware**	**Plants/Trees**
# images	102	37	50	174
category	**Tools**	**Trans. Vehicles**	**Weapons**	–
# images	72	158	83	–

We choose to evaluate the proposed TFC model in an image collection, since images are complex items to describe with social tags, representing many different aspects and therefore such a data set is challenging for recommender systems in social tagging. In order to decrease sparsity, we filtered out images that are annotated only once (or not annotated at all) as well as tags that are used only once in our data set. Additionally, tags are treated as regular text. The three preprocessing steps are: (a) tokenization based on a standard stop list (e.g. in, the, of, at, etc.), (b) tags are turned into lower case, (c) all non-letter or non-digit characters in the tags are removed (e.g. dots, commas, question marks, etc.). Tags preprocessing steps resulted in $\mathcal{U}$ =400 users, $\mathcal{I}$ =849 images, $\mathcal{T}$ =2,334 tags and $\mathbf{Y}$ =45,784 triplets. Concerning the experts, we asked from different experts to post tags at the remaining 849 images. Their expertise domains are in animals, airplanes, trains, flowers & trees and weapons, respectively. The tagging process of experts resulted in 237 expert tags and 376 additional triplets. Examples of the evaluation data are presented in Figure 3, where expert tags are denoted by bold letters.

Fig. 3. Examples of the evaluation data.

B. Experiments Organization

The rest of this section is organized as follows: firstly, we describe the evaluation protocol for item recommendation in social tagging systems. Then, we demonstrate how the parameters selection of both tag clustering methods is performed to conclude to the optimal number of tag clusters (topics) in our data set. In case of using expert tagging we denote it by (exp.). Afterwards, we evaluate the impact of step 1 (tag propagation based on relevance feedback) and step 2 (tag clustering) in the TF method (HOSVD) in terms of recommendations' quality. For comparison reasons we separately evaluate each of the first two steps of the TFC model against the method presented in [50], where HOSVD is used for item collaborative filtering and the work in [38], where in addition to HOSVD, content information was exploited to decrease sparsity and to solve the "cold start" problem. The reason for selecting both works is that they share the same TF technique (HOSVD) and therefore, the ternary relation among users tags and images is preserved. The wok in [50] is denoted by TF and the method presented in [38] is denoted by TF(a=0.5), where a is the control parameter of the propagated tags.

The TF technique (HOSVD) was implemented in Matlab Tensor Toolbox[2], following the MET technique presented in [22], which is suitable for handling scalable tensors. The

[2]http://csmr.ca.sandia.gov/ tgkolda/TensorToolbox/

Tripartite Clustering and the adapted K-Means tag clustering algorithms were also implemented in Matlab. Additionally, the implementation of the CEDD descriptor was provided by the authors of [33].

C. Evaluation Protocol

For the task of image recommendation, the following evaluation protocol is used: for each user, one of its triplets is randomly selected. The set of all selected triplets forms the test data, whereas the remaining triplets form the training data. The task of image recommendation is to predict the images in the hidden triplets. Following the evaluation protocol of the works in [52] and [38], the quality of recommendations are measured in terms of recall. Thus, for a test user U_i that receives a list of N recommended images (top-N list) by posing a tag query T_j, recall is defined as the the ratio of the number of relevant images in top-N list over the total number of relevant images (all images in the hidden triplets contain test user U_i and tag T_j). Other commonly used measures are precision and $F1$. However, the following two factors should be clearly mentioned: 1) For each user/tag combination in the test data, a constant number of images has to be predicted (images annotated with tag T_j by user U_i); and 2) only a pre-specified number N of recommendations is taken into account. Therefore, for this kind of evaluation protocol, it is redundant to evaluate precision (thus $F1$ too) because it is just the same as recall up to multiplicative constants. Regarding the problem that hidden triplets contain tags and not tag clusters, each tag in the triplets $\langle U_i, T_j, I_k \rangle$ is mapped to a tag cluster K_{T_j} according to (7) in order to transform the query from tag T_j to tag cluster-topic K_{T_j}. Moreover, we must mention that in all our experiments mean values of recall are reported, where each experiment was repeated 10 times. However, in many cases the differences between the recall values are minimal. To verify this, for all experiments we applied statistical pairwise t-tests, where the calculated differences of means were insignificant at level 0.05.

D. Parameters Selection for Tag Clustering Methods

Our parameter selection for the tag clustering methods differs to the approach presented in [32], since we aim to generate tag clusters and not item clusters. Therefore, each cluster is evaluated according to the Silhouette value $s(T_j)$ of each clustered tag T_j (also user and image in case of Tripartite Clustering). Silhouette [42] refers to a method of interpretation and validation of clusters of data and it is independent of the classification information. Let $x(T_j)$ be the average dissimilarity of T_j with all other tags within the same cluster. Then, the average dissimilarity of T_j with tags of another single cluster is found. We repeat this for every cluster of which T_j is not a member. We denote the lowest average dissimilarity to T_j of any such cluster by $y(T_j)$. Silhouette $s(T_j)$ of a clustered tag T_j (also user and image in case of Tripartite Clustering) is calculated as follows:

$$s(T_j) = \frac{x(T_j) - y(T_j)}{max(x(T_j), y(T_j))} \tag{11}$$

with $-1 \leq s(T_j) \leq 1$. Values below 0 means that the tag should be assigned to another cluster. Values near 0 means that the tag is on the borders of its cluster and values close to 1 means that the cluster is extremely coherent. This way we vary the number of clusters for the Tripartite Clustering and for the adapted K-Means algorithm to conclude to the optimal number of tag clusters, by finding the "peak", since for the extreme case where the number of tag clusters equals the number of tags, the Silhouette value is 1. Nevertheless, as we previously mentioned (Section III-C), we aim to cluster tags in order to reveal topics and decrease sparsity, therefore the extreme case is not our case. In the experiments for the Tripartite clustering we have to conclude to the optimal parameter, by keeping the rest parameters constant. Therefore, image and tag clusters are initially set to 11 (number of classes), weights α, β and γ are equal to 0.5. The experiments for the Tripartite Clustering are illustrated in Figure 4 from (a) to (g) and the experiment for the adapted K-Means is depicted in Figure 4(h). In all figures, mean values of the clustered tags' (or users' or images') Silhouette are reported. To compare our results with the one in [32] (in which the classification information was used), we evaluate only the image clusters in terms of Purity [56], since only the image clusters are those which are strongly correlated to classes. To compute Purity, each image cluster K_{I_b}, with $b \in 1, 2, \ldots, K_I$, is assigned to the category-class L_p, with $p \in 1, 2, \ldots, 11$ which is most frequent in the image cluster, and then, the accuracy of the overall cluster assignments is measured by dividing the total number of correctly assigned images by R (the number of clustered images in K_{I_b}). Purity for each image cluster K_{I_b} is calculated as $\frac{1}{R}\sum_r max_j |K_{I_b} \cap L_j|$. The respective experiment is presented in Figure 4(c), which confirms the result of Figure 4(b) that the optimal number of image cluster K_I equals 15.

To summarize our findings, for the Tripartite Clustering we conclude to the following optimal parameters: number of user clusters K_U=10, number of image clusters K_I=15, number of tag clusters K_T=35, where weights α, β and γ are equal to 0.5, 0.5 and 0.3, respectively. Also, for the case of the adapted K-Means tag clustering method we conclude to same number of tag clusters K_T=35. Note that despite the fact that a few expert tags are used, for each clustering method expert tagging produces more coherent clusters (larger Silhouette values), since these tags are very accurate to a topic [35]. More precisely, Tripartite clustering generates tag clusters with mean Silhouette values 0.374 and 0.480 with the expert tagging (adapted K-Means for tag clustering generates mean Silhouette values 0.342 and 0.409, respectively). In Table V, we illustrate examples of tag clusters K_T generated by the Tripartite Clustering. In particular, the first column corresponds to the cluster's topic, the second one to top-5 tags, ranked by their similarities to cluster's K_T centroid, and the third column contains examples of expert tags in K_T. Note that expert tags are located far from K_T's centroid, denoted by small similarities, since expert tags are only posted by a few experts, less often than the non-expert ones. However, expert tags contribute to more coherent clusters, compared to the initial data set in which they lack. As we experimentally

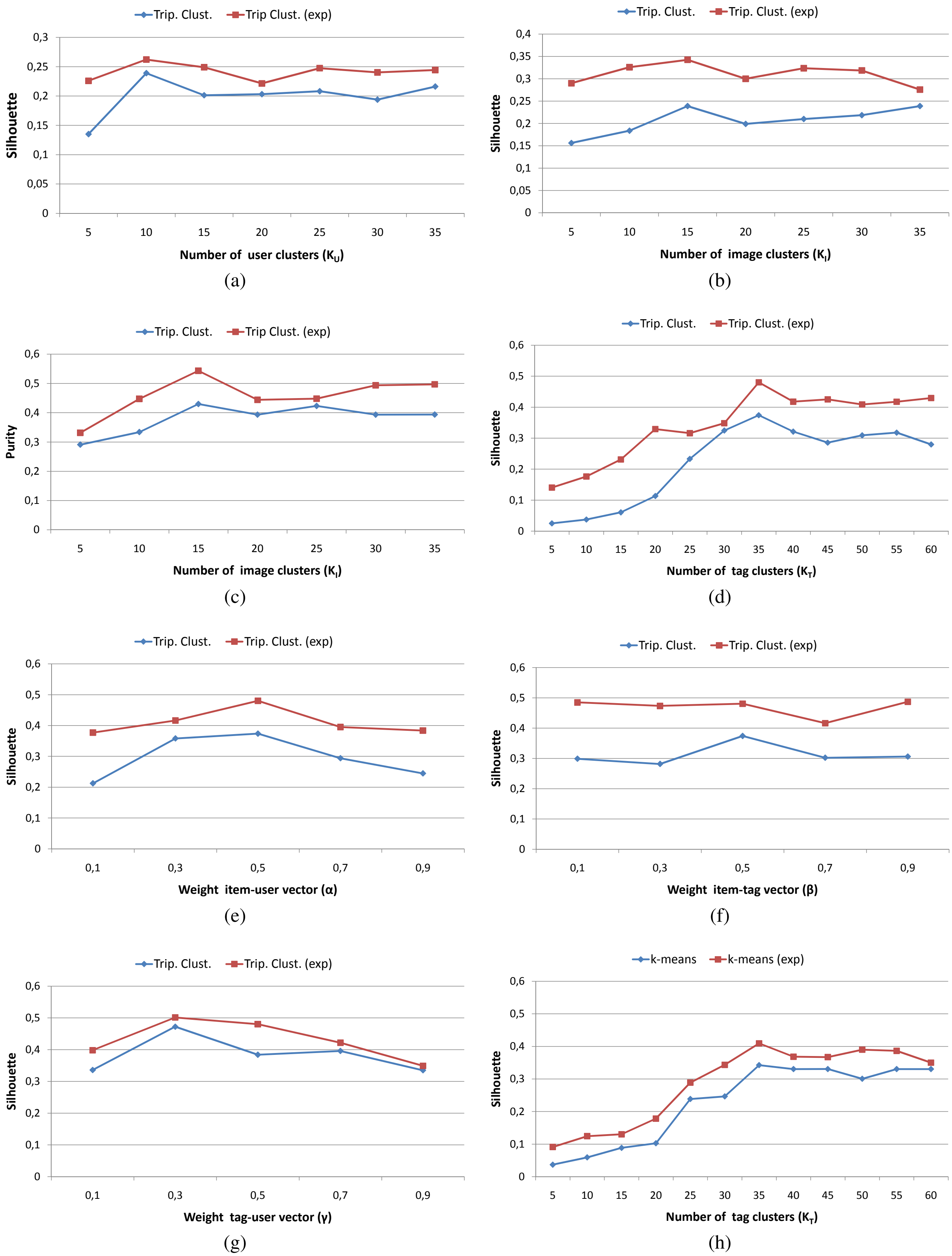


Fig. 4. For Tripartite Clustering mean Silhouette values of (a) users versus K_U with K_I=K_T=11 and (b) images versus K_I with K_U=10 and K_T=11. In (c) is illustrated the purity of image clusters on different settings of K_I and in (d) the mean Silhouette values of tags versus K_T with K_U=10 and K_I=15. Additionally, for K_U=10, K_I=15 and K_T=35 mean Silhouette values of tags by varying (e) the α parameter when $\beta = \gamma = 0.5$, (f) the β parameter when $\alpha = \gamma = 0.5$ and (g) the γ parameter when $\alpha = \beta = 0.5$. Finally, in (h) are presented the mean values of tags' Silhouette versus K_T for adapted K-Means tag clustering.

TABLE IV
EXAMPLES OF TOPICS OF TAG CLUSTERS K_T.

Topic of K_T	*Top-5 Tags (Sim. to K_T Centroid)*	*Examples of Experts' Tags in K_T*
airplanes	airplane (0.98) aircraft (0.97) plane (0.97) flight (0.93) jet (0.91)	c130 (0.58) boeing737 (0.39) airbus320 (0.22) f16 (0.21) p51mustang (0.19)
flowers & trees	flower (0.95) nature (0.89) green (0.81) tree (0.76) plant (0.63)	clematis (0.2) gillyflower (0.17) japanese maple (0.15) oleander (0.12) cedar (0.11)
trains	train (0.97) locomotive (0.92) rail (0.91) railway (0.88) station (0.83)	bullet train (0.43) electric train (0.37) TGV (0.29) capital metrotrain (0.23) intercity (0.14)
swords & knives	knife (0.95) sword (0.94) weapon (0.84) sharp (0.74) metal (0.71)	rapier sword (0.18) ballad (0.16) baseland (0.11) quillon (0.1) randel (0.05)
birds	animal (0.91) bird (0.87) fly (0.81) wings (0.76) sky (0.71)	bald eagle (0.27) sparrow (0.24) collared dove (0.18) roadrunner (0.18) quail (0.13)
fish	fish (0.99) sea (0.9) marine (0.85) ocean (0.83) aquarium (0.73)	blue merlin (0.24) spinner dolphin (0.22) bream(0.18) black bass (0.14) white shark (0.13)

demonstrate, more coherent tag clusters increase the quality of recommendations.

E. Impact of Tag Clustering on the TFC Model

Firstly, in Figure 5(a) we demonstrate how we conclude to the optimal number of retained singular vectors (parameters c_1, c_2 and c_3) of the initial TF method presented in [50]. A percentage of $c_1 = c_2 = c_3 = 60\%$ suffices in terms of recall, because we found that higher values increase the HOSVD's computational time, without paying-off in terms of the accuracy of prediction, based on the complexity of HOSVD [25]. Next, in Figures 5(b) and 5(c) we present the experimental results for the TFC model with the Tripartite clustering and adapted K-Means, denoted by TFC(Trip. Clust.) and TFC(k-means) respectively, by varying the percentage c_2 of the tag clusters' dimension I_2. This way, c_2 is set to 90%, since a further decrease of percentage c_2 affects the accuracy of the prediction. Note that the first step of tag propagation based on relevance feedback is omitted, since in the current Section we aim to validate only the impact of tag clustering on the TFC model.

Next, in Figure 6 from (a) to (c) we present the experimental results of how recall varies with respect to parameters α, β and γ in (3), (4) and (5), respectively. Furthermore, in Figure 6(d), we present recall of TFC(Trip. Clust.) and TFC(k-means), by varying the number of tag clusters K_T. We observed that we conclude to the same optimal values of α=0.5, β=0.5, γ=0.3 and K_T=35, with the experimental results in Section IV-D. According to this observation, we verify that generating coherent tag clusters (larger Silhouette values) is crucial for the TFC model to achieve high recommendation accuracies.

Furthermore, we compare TF to the TFC model by performing the Tripartite Clustering for the data set with expert tagging and for the initial data set in which they lack. From the results of Figure 7(a) we make the following three observations: (a) expert tagging in TF, denoted by TF(exp.), reduces the accuracy of TF, since sparsity increases along with the increase of user and tag dimensions, (b) the TFC model with the Tripartite clustering, outperforms the TF method, since sparsity is reduced by transforming tags to tag clusters, which directly affects the collaborative filtering mechanism as described in Section III-C, (c) a few expert tags can slightly improve the TFC model, denoted by TFC(Trip. Clust. exp.), since the tag clusters-topics are more coherent (larger Silhouette values) according to the experiments in Section IV-D and Figure 6.

Additionally, we evaluate the impact of the two different clustering methods in the TFC model, denoted by TFC(Trip. Clust.) and TFC(k-means). As shown in Figure 7(b), Tripartite Clustering slightly improves the accuracy of the TFC model, compared to the adapted K-Means in tag clustering, since the former method preserves the ternary relation among users, tags and images. Nevertheless, the ternary relation is enhanced according to the proposed $tf \cdot idf$ weighting scheme (7) in which users' interest, images and tags relation to a tag cluster are taken into account. Therefore, the improvement of Tripartite Clustering is small compared to the adapted K-Means in tag clustering.

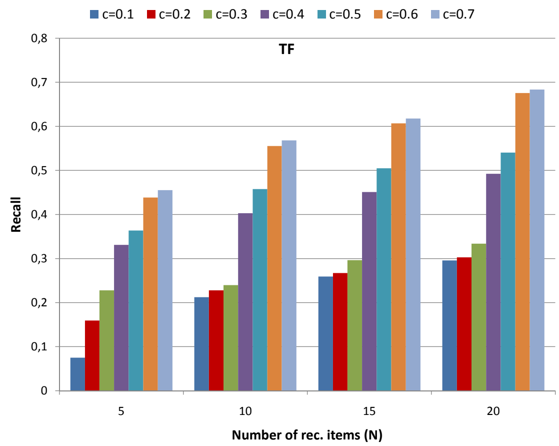
F. Impact of Tag Propagation based on Relevance Feedback in TFC Model

To evaluate the recommendations quality of the method in [38], firstly we control the amount of tag propagation by using the a parameter. TF(a=0.5) denotes that the tag propagation is performed only if the similarity between two images is greater than 0.5. Note that TF(a=1) is the initial TF method, without tag propagation. For lower values of a, tags are propagated aggressively. The evaluation of the method is presented in Figure 8(a). For all three values of a, the method can address sparsity and "cold start" problems, by propagating tags and generating additional triplets, as illustrated in Table V (note that the number of propagated triplets is equal to 0, for the initial case of a=1). However, the quality of recommendations is seriously affected by the noisy tags, regardless of performing aggressive or conservative tag propagation (low or high values of a) compared to the initial case of TF(a=1). This result indicates that the method in [38] is not sufficient for exploiting content to address sparsity and "cold start" problems, without being significantly affected by noise. This happens because the "learning tag relevance" issue is omitted.

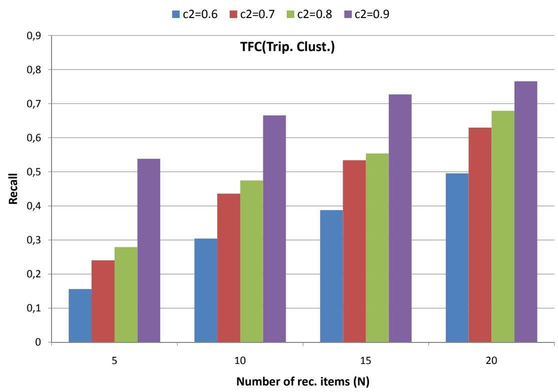
TABLE V
NUMBER OF PROPAGATED TRIPLETS WITH RESPECT TO THE CONTROL PARAMETER a, BASED ON WORK OF [38].

a	0.25	0.5	0.75
Num. Prop. Trip.	12,412,973	3,952,186	25,284

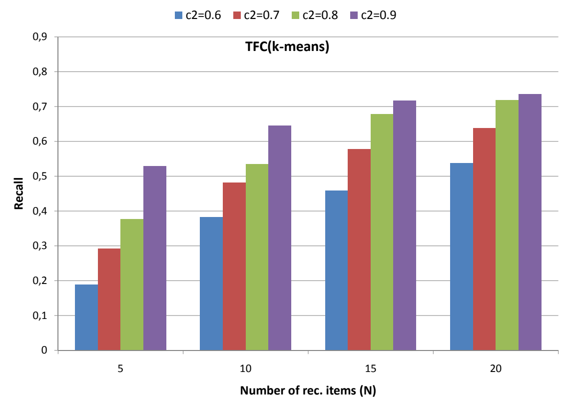
Next, we evaluate the step 1 of our TFC model, denoted



(a)



(b)



(c)

Fig. 5. Recall versus percentage of retained singular vectors for (a) TF with $c_1, c_2, c_3 = c$, (b) TFC(Trip. Clust) and (c) TFC(k-means) with $c_1, c_3 = 0.6$.

by TF(Rel. Feed.), where we perform tag propagation based on the relevance feedback mechanism according to (2), by omitting the tag clustering step. In Figure 8(b), we present how we conclude to the optimal value of the Rocchio parameter of positive feedback λ_p in (1). Note that the extreme case of λ_p=0, TF(Rel. Feed.) is the initial TF method, where the tag propagation process in (2) is omitted. Based on the results of Figure 8(b), we can make the following observations: (a) for all values of λ_p TF(Rel. Feed.) outperforms the accuracy of the TF method (λ_p=0), since according to (2) the tag propagation technique is performed between conceptual similar items, by considering the "learning tag relevance" issue, and (b) the accuracy is increased, since sparsity and "cold start" problems are addressed according to the number of propagated triplets, as depicted in Table VI. Therefore, we set λ_p equal to 1.8, to achieve the maximum number of propagated triplets and consequently the maximum recommendation accuracy of the TF(Rel. Feed.) method.

TABLE VI
NUMBER OF PROPAGATED TRIPLETS WITH RESPECT TO ROCCHIO PARAMETER OF POSITIVE FEEDBACK λ_p BASED ON (2).

λ_p	1	1.2	1.4	1.6
Num. Prop. Trip.	16,976	18,171	25,164	27,926
λ_p	1.8	2	2.2	–
Num. Prop. Trip.	30,759	26,262	12,786	–

Additionally, we conducted experiments to evaluate the whole TFC model, which includes all three steps, denoted by TFC(Rel. Feed.-Trip. Clust.), against the TF, the TF(Rel. Feed,) and the TFC(Trip. Clust.) method. In Figure 8(c) the respective results are presented, based on which two observations can be made: (a) TF(Rel. Feed.) outperforms TF, since the sparsity and the "cold start" problems are addressed and less noisy tags are propagated based on the relevance feedback mechanism by considering the "learning tag relevance" and therefore, tag propagation between conceptual similar images is performed and (b) an analogous improvement is demonstrated for the case of TFC(Trip. Clust.) compared to the entire TFC model TFC(Rel. Feed.-Trip. Clust.), since sparsity is further addressed.

G. Performance Issues

The TFC model consists of an offline part to build the model and an online part to generate the recommendations. Firstly, the computational time of the offline part is presented, consisting of (a) tag propagation based on relevance feedback, (b) tag clustering and (c) tensor factorization time.

The complexity of the tag propagation process, is $O(Iter \cdot \mathcal{I} \cdot \mathbf{R}_Q + \mathbf{Y} \cdot \mathbf{R}_Q)$, where $Iter$ is the iterations number of each query refinement, until the process terminates when the result list $\mathbf{R}_Q$ remains the same, and $\mathbf{Y}$ is the initial social data set of triplets. In our experiments we set $Iter = 4$, since for larger

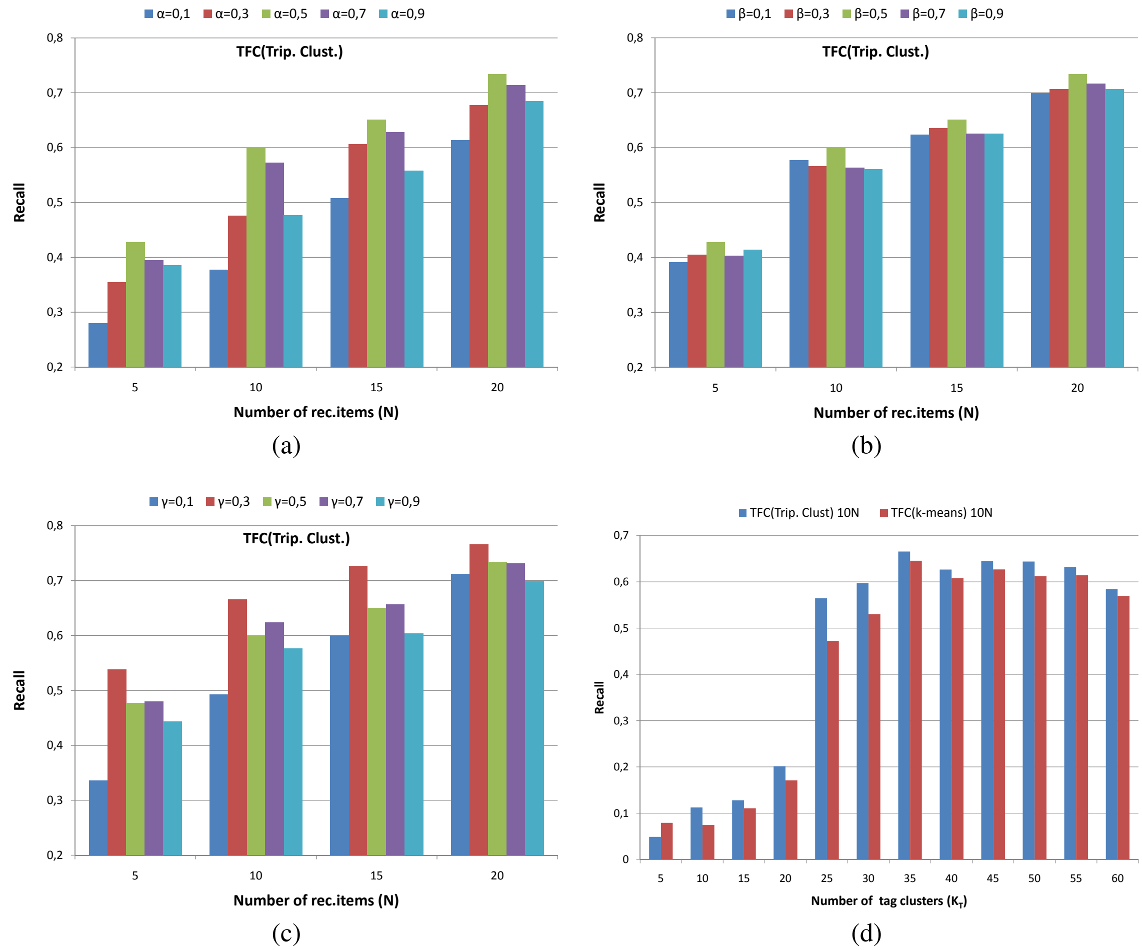


Fig. 6. Recall of TFC(Trip. Clust.) with respect to the number of recommended items N, by varying (a) the α parameter in (3), with $\beta = \gamma = 0.5$, (b) the β parameter in (4), with $\alpha = \gamma = 0.5$ and (c) the γ parameter in (5), with $\alpha = \beta = 0.5$. In (d) we present recall of TFC(Trip. Clust.) and TFC(k-means) versus the number of tag clusters K_T for $10N$ recommended items.

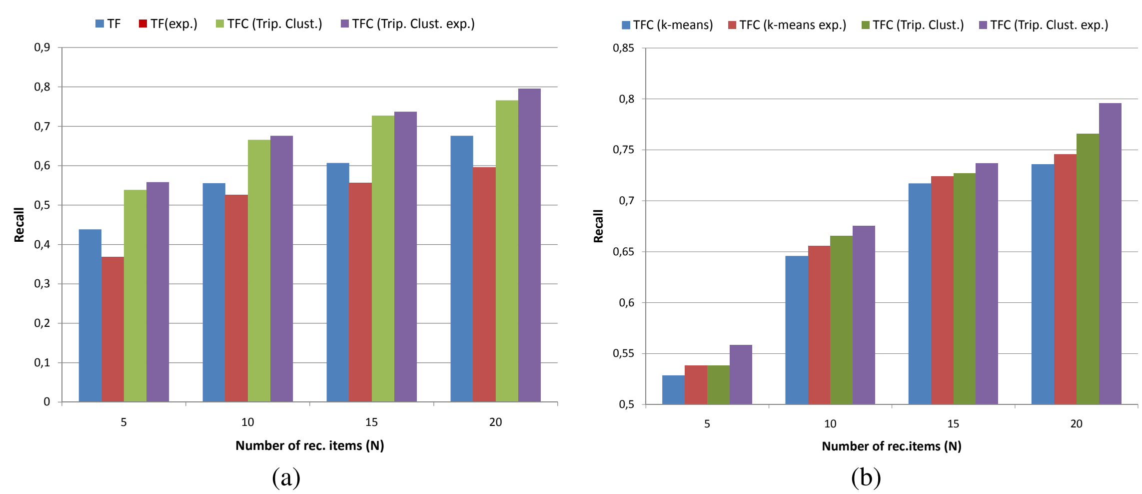
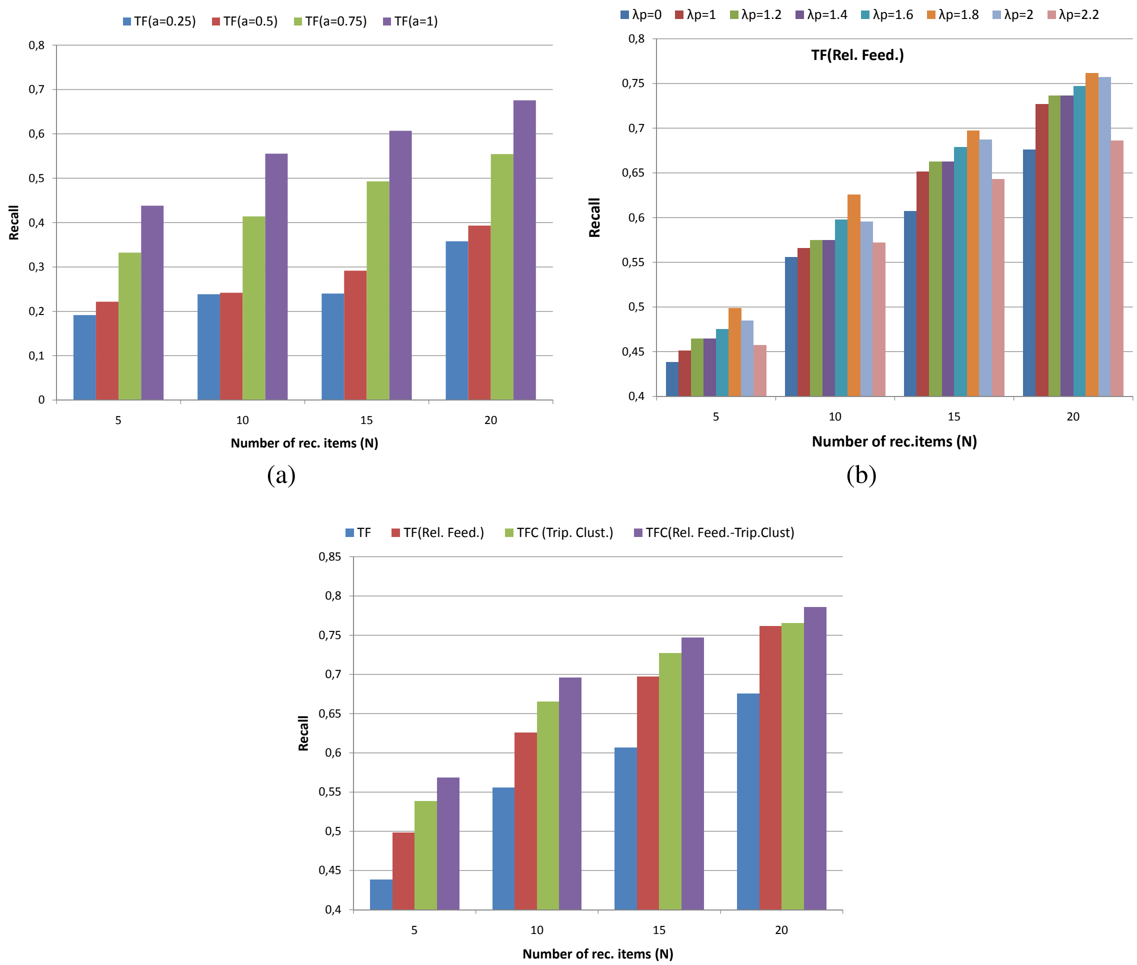


Fig. 7. Recall of TFC(Trip. Clust.) versus (a) TF and (b) TFC(k-means), with respect to the number of recommended items N.



(a) (b)

(c)

Fig. 8. Recall of TF with respect to the number of recommended items N, using tag propagation based on content, by varying (a) the value of the control parameter a based on [38] and (b) the parameter λ_p in (1) based on the proposed tag propagation via relevance feedback in (2). In (c) we present the evaluation comparison of TFC(Rel. Feed.-Trip. Clust.) against the TF, the TF(Rel. Feed,) and the TFC(Trip. Clust.) method.

number of $Iter$ $\mathbf{R}_Q$ for all queries remained the same. Also, we set $\mathbf{R}_Q$=10, since it is the minimum length of the result list $\mathbf{R}_Q$, to retrieve the maximum number of positive marked images for all queries.

The complexity of the scalable adapted K-Means for tag clustering is $O(Iter \cdot K_T \cdot \mathcal{T} \cdot \mathcal{I})$, where $Iter$ is the iterations number until the centroids remain the same or the number of iterations exceeds a maximum threshold. In our experiments, we set the maximum threshold of iterations to 30, to ensure that after a large number of iterations the centroids remain the same. The complexity of the Tripartite Clustering [32] is $O(Iter \cdot K_T \cdot \mathcal{T} \cdot (\mathcal{U} + \mathcal{I}) + Iter \cdot {K_T}^2 \cdot \mathcal{T})$. Nevertheless, the Tripartite Clustering generates users and image clusters simultaneously, which should be added to the total cost of the algorithm.

The complexity of the Tensor Factorization technique (HOSVD), according to [25], for the initial case of TF is:

$$O(max(c_1 \cdot \mathcal{U}^2 \cdot \mathcal{T} \cdot \mathcal{I}, \mathcal{U} \cdot c_2 \cdot \mathcal{T}^2 \cdot \mathcal{I}, \mathcal{U} \cdot \mathcal{T} \cdot c_3 \cdot \mathcal{I}^2)) \quad (12)$$

and for the TFC model is:

$$O(max(c_1 \cdot \mathcal{U}^2 \cdot K_T \cdot \mathcal{I}, \mathcal{U} \cdot c_2 \cdot {K_T}^2 \cdot \mathcal{I}, \mathcal{U} \cdot K_T \cdot c_3 \cdot \mathcal{I}^2)) \quad (13)$$

The respective results are presented in Figure 9, based on which we can observe that by reducing tags $\mathcal{T}$ to tag clusters K_T, the computational cost of HOSVD is decreased. However, for the case of the TFC model, the computational cost for tag propagation and tag clustering have to be added. Yet, the whole computational cost of the TFC model is low, especially for the case, where the adapted K-Means tag clustering is performed, by an affordable loss in accuracy, as depicted in Figure 7(b).

The online part of generating recommendations based on the TFC model is very efficient, because it just needs to sort a vector of predictions-images for a pair (user, tag cluster), according to the w values in the final reconstructed tensor. Therefore, the complexity of the online part is $O(\mathcal{I} \cdot \log \mathcal{I})$ for a pair of a user U_i and a tag cluster K_{T_j}. Thus, the online part of the proposed TFC model lasts $\approx$ 0.17 msec and it is the same for all users' queries.

H. Evaluation of TFC model against state-of-the-art methods

For comparison reasons, we conducted experiments to evaluate TFC(k-means) against the TF and the TF(a=0.5) methods in two additional data sets. The first evaluation data set (denoted by DS1) is described in [39], consisting of 12,773

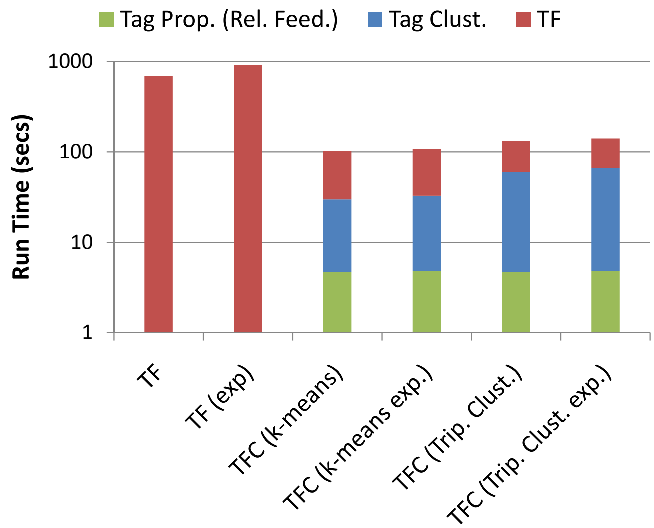


Fig. 9. Run times of TF and TFC model. TFC time is calculated by the sum of tag propagation based on relevance feedback, tag clustering and TF time.

triplets in the form user-song-tag, with 4,442 users, 1,620 songs and 2,939 tags. The second evaluation data set (denoted by DS2 and described in [38]), consists of 64,025 triplets with 732 users, 991 songs and 2,527 tags. Note that in case of TFC(k-means) the first step of tag propagation based on relevance feedback is omitted, since the information of the conceptual similar songs is unavailable in DS1 and DS2 (the triplets have been randomly crawled from [18], by ignoring the class information of songs). Regarding the parameter selection for the tag clustering method of adapted K-Means in TFC(k-means), we followed the experimental evaluation of Figure 5, where the percentage of singular vectors c_1, c_2, c_3 are set to 60%, 90% and 60% for both data sets. Additionally, following the experimental configuration of Sections IV-D and IV-E, the numbers of tag clusters are set to 50 and 40 for data sets DS1 and DS2, respectively, based on the results in Figures 10(a) and 10(b), where mean Silhouette values of tags and recall for $10N$ recommended items are reported.

Next, in Figures 10(c) and 10(d), we report recall of TF (denoted by TF(a=1)), TF(a=0.5) and TFC(k-means), by varying the number of recommended items (N) on the x-axis, as presented in [39] and [38], respectively. We can make the following observations: (a) TFC(k-means) outperforms the TF method, confirming the experimental results in Section IV-F, since the former address the sparsity and "cold start"problems and (b) TFC(k-means) slightly increases recall compared to TF(a=0.5), since the latter also handles sparsity and "cold start" problems, by exploiting the songs content for data sets DS1 and DS2, as also presented in the experimental evaluation of [39] and [38]. Nevertheless, as we experimentally proved in Figure 8(a) for the TF(a=0.5) or TF(a=0.75) methods, controlling the number of propagated triplets with the a parameter can also decrease the quality of recommendation in terms of recall, because the "learning tag relevance" issue is not handled, Additionally, based on the complexity analysis in (12) and (13), TFC(k-means) has lower building requirements than TF(a=0.5), since the latter equals the complexity of the TF method according to [38]. Considering the aforementioned factors, TFC(k-means) clearly outperforms the TF and TF(a=0.5) methods in both evaluation data sets, despite the fact that the first step of the TFC model is omitted.

V. Conclusion

The proposed TFC model for item recommendation in social tagging systems is a promising method, since each of its steps handles several issues that exist in social tagging systems, which produce noise and decrease the quality of recommendations. Through its first step, the content information is exploited to propagate tags between the most visually similar images that belong to the same concept, in order to handle the issue of "learning tag relevance" and to solve the sparsity and "cold start" problems. Through its second step, the problem of sparsity is further handled, by revealing tag clusters, which correspond to topics. Furthermore, users interest in topics is taken into account, by following an innovative $tf \cdot idf$ weighting scheme. In our experiments we show that a few expert tags improve the performance of quality recommendations, since they contribute to more coherent tag clusters. Through the third step, the TFC model preserves the ternary relation among users, tag clusters and images, solves the problem of polysemy and synonymity and reveals the latent associations among users, topics and images. Additionally, the proposed TFC model minimizes the computational cost of the Tensor Factorization technique (HOSVD), by reducing the dimension of tags to tag clusters. In our experiments we show that by tackling several real world problems, the proposed TFC model achieves higher recommendation accuracies than the state-of-the-art methods of [50] and [38], where the same Tensor Factorization technique (HOSVD) is applied.

Nevertheless, two points should be further investigated in the TFC model. The first point is that the HOSVD method is time consuming for building the recommendation model at the offline part, since tuning is required for finding the optimal rank of the reconstructed tensor (values c_1, c_2, c_3). This problem is already identified in [41], but their proposed method is for tag recommendation purposes, which differs to item recommendation. The second point is that the the optimal number of tag clusters-topics is concluded in an empirical way (by finding the "peak"). Therefore, an automatic way for finding the optimal number of tag clusters is required, since as we experimentally proved the number of tag clusters is crucial for the TFC model to achieve high recommendation accuracies. Recently, there has been a growing interest in multi-way probabilistic clustering and some efficient algorithms have been developed for this problem. For example He et al.'s [13] algorithm is proposed for automatically detecting number of clusters in a tensor-based framework. However, all these are useful directions for future research.

Acknowledgment

This work was supported by the EC-funded project I-SEARCH. We would like to thank the members of the I-SEARCH consortium for participating in the data set pro-

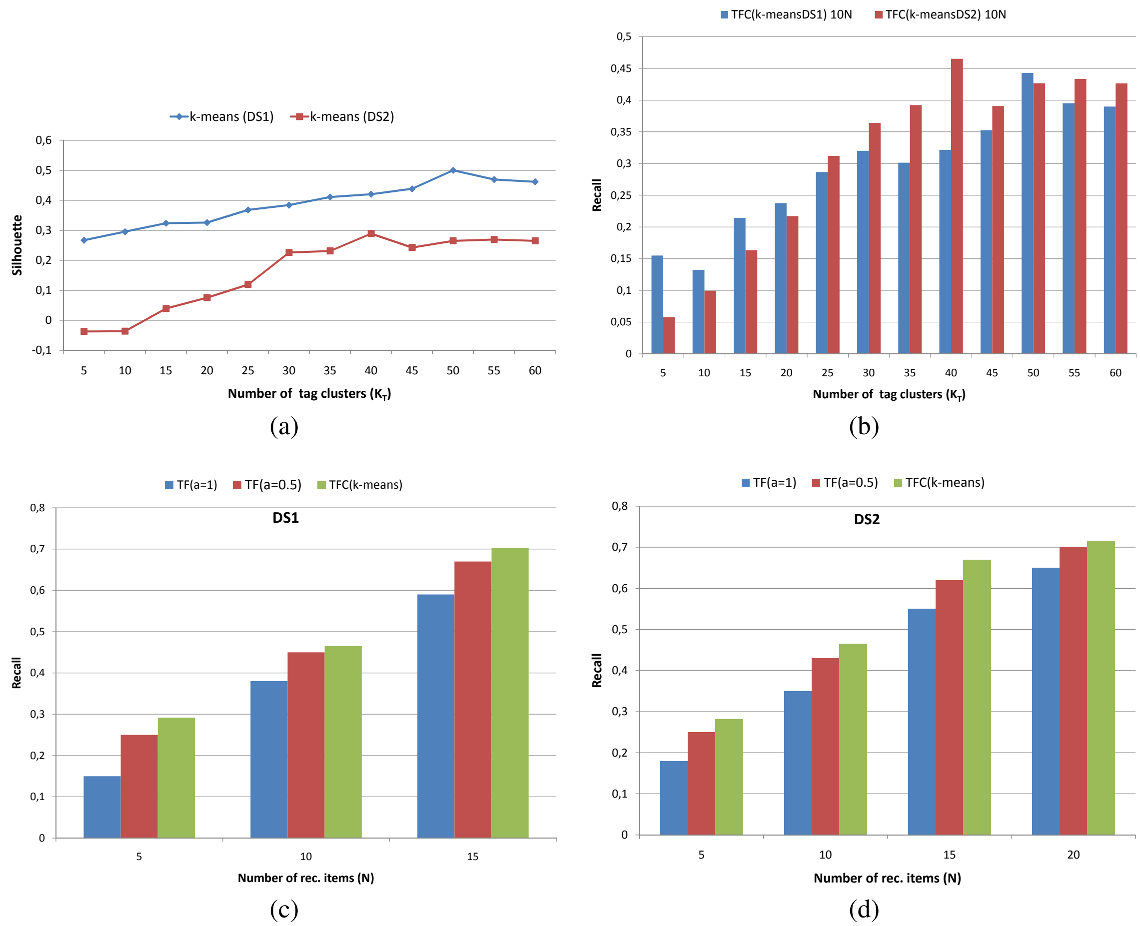


Fig. 10. TF versus TFC model, in the evaluation data set of [39] and [38], denoted by DS1 and DS2, respectively, by varying the number of tag clusters K_T in terms of (a) tags' Silhouette and (b) recall. In (c) and (d) the respective results are reported for DS1 and DS2, by varying the number of N recommended items, as presented in [39] and [38].

duction. We would also like to thank the author of [39] for providing us the respective evaluation data set.

References

[1] J. Ahn, P. Brusilovsky, J. Grady, D. He, and S.Y Syn, "Open User Profiles for Adaptive News Systems: Help or Harm?" in *Proc. Int. Conf. on WWW*, pp. 11-20, 2007.

[2] M. Balabanovic, Y. Shoham, "Fab: Content-based, Collaborative Recommendation", *Communications of the ACM*, vol. 40, no. 3, pp. 66-72, 1997.

[3] G. Begelman, P. Keller, and F. Smadja, "Automated tag clustering: Improving search and exploration in the tag space", in *Proc. WWW Collaborative Web Tagging Workshop*, 2006.

[4] T. Bogers, W. Thoonen, A. Van Den Bosch, "Expertise classification: Collaborative classification vs. automatic extraction", in *Proc. 17th SIG Classification Research Workshop*, 2006.

[5] S. Chatzichristofis, and Y. Boutalis, "A Hybrid Scheme for Fast and Accurate Image Retrieval Based on Color Descriptors", in *Proc. IASTED International Conference on Artificial Intelligence and Soft Computing*, 2007.

[6] P. Drineas and M. W. Mahoney, "A randomized algorithm for a tensor-based generalization of the svd", *Linear Algebra and Its Applications*, vol. 420, no. 2-3, pp. 553-571, 2007.

[7] S. Farrell and T. Lau, "Fringe contacts: People-tagging for the enterprise", in *Proc. WWW Collaborative Web Tagging Workshop*, 2006.

[8] G. Furnas, S. Deerwester, and S. Dumais, "Information retrieval using a singular value decomposition model of latent semantic structure", in *Proc. ACM SIGIR Conf.*, 1988, pp. 465-480.

[9] J. Gemmell, A. Shepitsen, B. Mobasher, and R. Burke, "Personalization in Folksonomies Based on Tag Clustering", *Intelligent Techniques for Web Personalization & Recommender Systems Workshop*, 2008.

[10] J. Gemmell, A. Shepitsen, B. Mobasher, and R. Burke, "Personalizing navigation in folksonomies using hierarchical tag clustering", in *Proc. Int. Conf. on Data Warehousing and Knowledge Discovery*, 2008, pp. 259-266.

[11] S. Golder and B. Huberman, "The structure of collaborative tagging systems", in *Proc. CoRR*, 2005, pp. 1-8.

[12] H. Halpin, V. Robu, and H. Shepherd, "The complex dynamics of collaborative tagging", in *Proc. Int. Conf. on WWW*, 2007, pp. 211-220.

[13] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the Number of Clusters in n-Way Probabilistic Clustering", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, November 2010.

[14] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl, "Evaluating collaborative fltering recommender systems", *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5-53, Jan. 2004.

[15] S. C. Hoi, M. R. Lyu, and R. Jin, "A unified log-based relevance feedback scheme for image retrieval", *IEEE Trans. On Knowledge and Data Engineering*, vol. 18, no. 4, pp. 509-524, 2006.

[16] http://www.flickr.com/

[17] http://www.delicious.com/

[18] http://www.lastfm.com/

[19] http://www.pandora.com/

[20] F. Jing, B. Zhang, F. Z. Lin, W. Y. Ma, and H. J. Zhang, "A novel region based image retrieval method using relevance feedback", in *Proc. ACM Int. Workshop on Multimedia Information Retrieval (MIR)*, 2001.

[21] A. John and D. Seligmann, "Collaborative tagging and expertise in the enterprise", *in Proc. WWW Collaborative Web Tagging Workshop*, 2006.

[22] T.G. Kolda and J. Sun, "Scalable Tensor Decompositions for Multi-aspect Data Mining", in *Proc. IEEE Int. Conf. on Data Mining*, 2008, pp. 363-372.

[23] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems", *IEEE Computer*, vol. 42, no. 8, pp. 30-37, 2009.

[24] R. Lambiotte and M. Ausloos, “Collaborative tagging as a tripartite network”, *Technical report*, SUPRATECS, Universit de Lige,B5 Sart-Tilman, B-4000 Liege, Belgium, 2005.
[25] L. De Lathauwer, B. De Moor, and J. Vandewalle, “A multilinear singular value decomposition”, *SIAM Journal of Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253-1278, 2000.
[26] M. Levy and M. Sandler, “Learning latent semantic models for music from social tags”, *Journal of New Music Research*, vol. 37, no. 2, pp. 137-150, 2008.
[27] X. Li, C.G.M. Snoek, and M. Worring, “Unsupervised multi-feature tag relevance learning for social image retrieval”, in *Proc. ACM Int. Conf. on Image and Video Retrieval CIVR*, 2010, pp. 10-17.
[28] X. Li, C.G.M. Snoek, and M. Worring, “Learning Social Tag Relevance by Neighbor Voting”, *IEEE Trans. on Multimedia*, vol. 11, no. 7, pp. 1310-1322, 2009.
[29] D. Liu and T. Chen, “Probabilistic relevance feedback wtih binary semantic feature vectors, in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2005, pp 513-516.
[30] P. Lops, M. De Gemmis, and G. Semeraro, “Content-based Recommender Systems: State of the Art and Trends”, in *Recommender Systems Handbook*, Springer 1st Edition 2011, ch 3, pp. 73-105.
[31] Y. Lu, C. Hu, X. Zhu, H. J. Zhang, and Q. Yang , “A unified framework for semantics and feature based relevance feedback in image retrieval systems”, in *Proc. ACM Int. Conf. on Multimedia*, 2000, pp. 31-37.
[32] C. Lu, X. Hu, and J.R. Park, “Exploiting the Social Tagging Network for Web Clustering”, *IEEE Trans. Syst.,Man, Cybern. A, Syst., Humans*, vol. 41, no. 5, pp. 840-852, 2011.
[33] M. Lux and S. A. Chatzichristofis, “Lire: lucene image retrieval: an extensible java CBIR library”, in *Proc. ACM Int. Conf. on Multimedia*, 2008, pp. 1085-1087.
[34] B.S. Manjunath, P. Salembier, and T. Sikora, Eds., “Introduction to MPEG-7: Multimedia Content Description Interface”, Chichester, West Sussex, U.K.: Wiley, 2002.
[35] L. B. Marinho, A. Nanopoulos, L. Schmidt-Thieme, R. Jaschke, A. Hotho, G. Stumme and P. Symeonidis, “Social Tagging Recommender Systems” in *Recommender Systems Handbook*, Springer 1st Edition 2011, ch. 19, pp. 615-644.
[36] B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme, “Evaluating similarity measures for emergent semantics of social tagging” in *Proc. Int. Conf. on WWW*, 2009, pp. 641-650.
[37] A. Nanopoulos, H. Gabriel, and M. Spiliopoulou, “Spectral clustering in social-tagging systems”, in *Proc. Int. Conf on Web Information Systems Engineering*, 2009, pp. 87-100.
[38] A. Nanopoulos, D. Rafailidis, P. Symeonidis, and Y. Manolopoulos, “Musicbox: Personalized music recommendation based on cubic analysis of social tags”, *IEEE Trans. on Audio, Speech and Language Processing*, vol. 18, no. 2, pp. 407-412, 2010.
[39] A. Nanopoulos, “Item Recommendation in Collaborative Tagging Systems”, *IEEE Trans. Syst.,Man, Cybern. A, Syst., Humans*, vol. 41, no. 4, pp. 760-771, 2011.
[40] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina, “Clustering the tagged Web”, in *Proc. ACM Int. Conf. Web Search Data Mining*, 2009, pp. 54-63.
[41] S. Rendle, L.B. Marinho, A. Nanopoulos, and L. SchmidtThieme, “Learning optimal ranking with tensor factorization for tag recommendation”, in *Proc. ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, 2009, pp. 727-736.
[42] P. J. Rousseeuw, “Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis”, *Computational and Applied Mathematics*, vol. 20, pp. 53-65, 1987.
[43] G. Salton, “The Smart document retrieval project”, in *Proc. ACM SIGIR*, 1991, pp. 356-358.
[44] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms”, in *Proc. Int. Conf. on WWW*, 2001, pp. 285-295.
[45] R. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, and G. Weikum, “Efficient top-k querying over social tagging”, in *Proc. ACM Int. Conf. SIGIR* , 2008, pp. 523-530.
[46] C. Schmitz, A. Baldassarri, V. D. P. Servedio, V. Loreto, A. Hotho, M. Grahl, and G. Stumme, “Network properties of folksonomies”, *AI Commun.*, vol. 20, no. 4, pp. 245-262, Dec. 2007.
[47] A. Shepitsen, J. Gemmell, B. Mobasher, and R.D. Burke, “Personalized recommendation in social tagging systems using hierarchical clustering”, in *Proc. ACM Conf. on Recommender Systems*, 2008, pp. 259-266.
[48] B. Sheth and P. Maes, “Evolving Agents for Personalized Information Filtering”, in *Proc. IEEE Artificial Intelligence for Applications*, 1993, pp. 345-352.
[49] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349-1380, 2000.
[50] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, “A Unified Framework for Providing Recommendations in Social Tagging Systems Based on Ternary Semantic Analysis”, *IEEE Trans. on Knowl. and Data Eng.*, vol 22, no. 2, pp. 179-192, 2010.
[51] J. Tang, S. Yan , R. Hong , G. J. Qi and T. S. Chua, “Inferring Semantic Concepts from Community-Contributed Images and Noisy Tags”, in *Proc ACM Int. Conf. on Multimedia*, 2009, pp. 223-232.
[52] K. Tso-Sutter, B. Marinho, and L. Schmidt-Thieme, “Tag-aware recommender systems by fusion of collaborative filtering algorithms”, in *Proc. ACM SAC*, 2008, pp. 1995-1999.
[53] P. Turney, “Empirical evaluation of four tensor decomposition algorithms”, *Technical Report* (NRC/ERB-1152), 2007.
[54] X. Wu, L. Zhang, and Y. Yu, “Exploring social annotations for the Semantic Web”, in *Proc. Int. Conf. on WWW*, 2006, pp. 417-426.
[55] Y. Xu, L. Zhang, and W. Liu, “Cubic analysis of social bookmarking for personalized recommendation”, in *Proc. APWeb*, 2006, pp. 733-738.
[56] Y. Zhao and G. Karypis, “Criterion functions for document clustering: Experiments and analysis”, Dep. CS, Univ. Minnesota, Minneapolis, MN, 2001, Tech. Rep.
[57] D. Zhou, J. Bian, S. Zheng, H. Zha, and C. L. Giles, “Exploring social annotations for information retrieval”, in *Proc. Int. Conf. on WWW*, 2008, pp. 715-724.

Dimitrios Rafailidis was born in Larisa, Greece, in 1982. He received the Diploma in Informatics from the Computer Science Department, the M.Sc degree in Information Systems and the Ph.D. degree in Music Information Retrieval from the Aristotle University of Thessaloniki, Greece in 2005, 2007 and 2011, respectively. His main research interests include data mining, information retrieval and recommender systems. Currently, he is a Postdoctoral Research Fellow at Information Technologies Institute of the Center for Research and Technology Hellas (CERTH).

Petros Daras (M’07) was born in Athens, Greece, in 1974. He received the Diploma degree in electrical and computer engineering, the M.Sc. degree in medical informatics, and the Ph.D. degree in electrical and computer engineering, all from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1999, 2002, and 2005, respectively. He is a Researcher Grade C, at the Information Technologies Institute of the Center for Research and Technology Hellas (CERTH). His main research interests include search, retrieval and recognition of 3-D objects, 3-D object processing, medical informatics applications, medical image processing, 3-D object watermarking, and bioinformatics. He serves as a reviewer/evaluator of European projects. Dr. Daras is a key member of the IEEE MMTC 3DRPC IG.