Image tagging using tensor decomposition

Michail Panagopoulos
Department of Audio and Visual Arts
Ionian University
Tsirigoti Sq. 7, 49100 Corfu, Greece
Email: mpanagop@ionio.gr

Constantine Kotropoulos
Department of Informatics
Aristotle University of Thessaloniki
University Campus 54124 Thessaloniki, Greece
Email: costas@aiia.csd.auth.gr

***Abstract*—Social media growing has resulted into a huge amount of information. Meta-data, accompanying the raw data, can assist data manipulation and processing, e.g.the tags assigned to social images. Many systems for automatic image tagging are based on pair-wise relations. Recent approaches focus on relations among multiple vertices (i.e. items), which are respresented as hyperedges in a hypergraph. In this work, an image tagging methodology is proposed that exploits a factorization of a tensor, capturing high-order relations among multiple Flickr images. The proposed approach uses an extended graph, where the vertices represent images as well as users. By analysing the data communities and their similarities, image annotation is proposed.**

I. Introduction

Automatic image tagging has received a lot of attention over the last years because of the rapid development of social media networks like Flickr[1], Instagram[2], Panoramio[3]. Image annotation is of major importance in information retrieval tasks and particularly for social media recommendation systems. The most unambiguous way to tag an image is the manual labeling. However, this is time consuming and costs too much. So automatic tagging methods take place in order to achieve fast and cheap data organization. Most of the social media platforms offer to users the ability to add meta-data when each image is uploaded, such as tags, geo-tags, groups, etc. Many users do not tag their images or, when they do tag them, they only use a small number of tags. If we take into account the fact that there are many different kinds of tags, e.g., content-based tags, ownership tags, context-based tags etc., it is easy to understand the difficulties of annotating social images. Noisy tags, namely tags that consist of the same words, but with completely different meaning or tags that are irrelevant with the context or the content of an image, can lead to false tag recommendations. Therefore, multiple information sources should be established.

A. Image tagging

Previous work on image tagging focuses mostly on the image content, i.e., identifying image features to create clusters, that are next exploited to annotate the images. Researchers are using *Support Vector Machines* (SVM) [1], or classify the images by means of artificial neural networks (ANN) [2]. Automatic image annotation using a multiple Bernoulli relevance model using the joint probability distribution of the possible annotations and the image feature vectors is proposed in [3]. Other approaches resort to nearest neighbors [8], [9], structural group sparsity for feature selection or boost annotation performance by exploiting the correlations among multiple tags [10] and semantic distance functions [11], while mining the image search results is used in [12] to accomplish the annotation. More recently, scholars examine the optimal combination of, namely a sparse kernel learning framework for the continuous relevance model [14], while others propose integration of Hessian regularization with discriminative sparse coding for multiview problems [15]. Non-negative matrix factorization for image annotation is used in [16]. Detailed reviews about automatic annotation techniques can be found in [4] and [5].

This work is focused on multiple meta-data connections, such as the combined relations among users, friendships, groups, tags, geo-tags, and image similarities. Existing methods that use information from multi-type interrelated objects are described in [6], [7]. Such methods employ graph-based methods. Multi-label image tagging is also addressed in [13] within a sparse coding framework.

B. Tensor recommendation

Since the problem at hand has a multivariable nature, a formulation of tag recommendation using tensor looks appealing. Indeed, tensors can be applied to tag recommendation systems, because they can represent easily the connection of more than two nodes of a graph. The present work targets on recommending tags for images that are connected with users, who have uploaded them to a social network. The connections between the items, which form the graph, are multiple. For example, users are connected to the uploaded image and also to a corresponding tag, creating this way a triple edge, which can be be represented by an element of 3rd-order tensor.

Recommendation systems using tensors have received a lot of attention over the last years, because of the powerful properties of tensor decompositions. More specifically, a Higher Order Singular Value Decomposition (HOSVD) is performed in [17] and ranking scores for tag recommendation are obtained through tensor unfoldings and SVD factorization. Personalized tag recommendation is exploited in [18], handling missing values and learning from pairwise ranking constraints.

II. Tensor factorization

A. Notation and preliminaries

A matrix can be defined as a tensor with two dimensions. A vector is a tensor of dimension one. Multidimensional matrices

[1]http://www.flickr.com/
[2]http://instagram.com/
[3]http://www.panoramio.com/

are called higher order tensors, e.g., a three dimensional matrix is a third-order tensor. In this paper, the notation in [19] will be used. Namely, scalars will be denoted by lowercase letters and vectors by boldface lowercase letters. A vector element will be denoted as α_i. Matrices appear in boldface capital letters, and their elements will be denoted as α_{ij}. Tensors will be denoted by boldface Euler script letters, e.g. for tensor $\mathcal{X}$, its elements will be x_{ijk}. Next, some important term definitions related to tensors have to be made. Tensor mode or way is the tensor dimension. A tensor fiber is defined, when two of the tensor indices are fixed. For example, for the first dimension, one can obtain mode-1 fiber, which corresponds to a matrix column. A slice is a matrix instance of the tensor. In other words, when only one tensor index is fixed, one can obtain a two dimensional representation, e.g., for fixed 3^{rd} dimension the result is the frontal tensor slice $\boldsymbol{X}_{::k}$. Finally, the tensor norm is defined similarly to the Frobenius matrix norm, i.e.,

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x^2_{i_1 i_2 \cdots i_N}}. \quad (1)$$

Matricization: The procedure of transforming a tensor to matrix is called matricization or unfolding. There exist three types of unfolding, one for each tensor mode. More specifically, the mode-1 matricization of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ yields the matrix $X_{(1)} \in \mathbb{R}^{I_1 \times I_2 I_3}$. In an analogous manner, mode-2 and mode-3 matricizations can be performed by concatenating the transposed frontal slices and the tube (mode-3) fibers respectively, resulting in the unfoldings $X_{(2)} \in \mathbb{R}^{I_2 \times I_1 I_3}$ and $X_{(3)} \in \mathbb{R}^{I_3 \times I_1 I_2}$.

B. Tensor Decompositions

As described in the Introduction, the most important benefit of using tensors to model recommendation systems is the ability of tensor factorizations to create communities, clusters, and ranked similarities. In an analogous way with the SVD decomposition for matrices, tensor factorization can be described as the procedure that provides basic matrices, which have specific properties. Some of the decomposed matrices can group items. Some others show the connections between terms and items. Two major tensor decompositions are the Canonical Decomposition/Paralel Factor Analysis (CP) CANDECOMP/PARAFAC (CP) decomposition and the TUCKER decomposition. Each of these decompositions, along with their different versions, has advantages and disadvantages. A detailed description of their applications can be found in [19]. Both methods of higher-order factorization apply Alternating Least Squares (ALS) to obtain an approximation of the input tensor. The fitting error can vary, depending on the rank selection, the applied ALS method, and other parameters (e.g., the selected regularization method for the loss function evaluation).

1) CANDECOMP/PARAFAC: Tensor factorization decomposes a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ into a sum of rank-one tensors. Each one of the rank-one tensors, is an outer product of three vectors. It corresponds to a factor. If these vectors are normalized to length one, the decomposition is [20]:

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda(r) \boldsymbol{A}(:,r) \circ \boldsymbol{B}(:,r) \circ \boldsymbol{C}(:,r) \equiv [\![\boldsymbol{\lambda}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!] \quad (2)$$

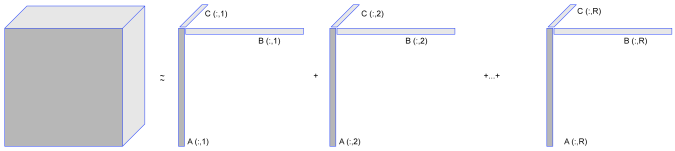

Fig. 1. CANDECOMP/PARAFAC decomposition

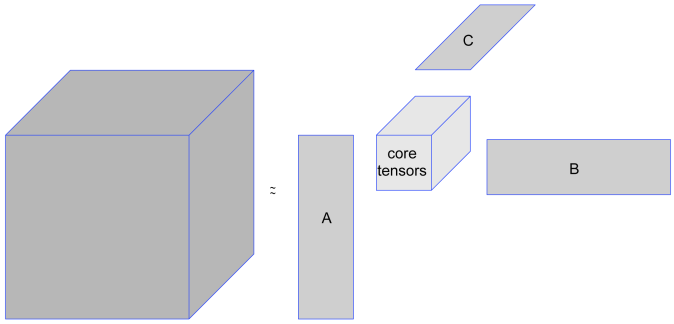

Fig. 2. TUCKER decomposition

where $\lambda \in \mathbb{R}^R$ and r denotes the factor. Figure 1 demonstrates graphically the tensor decomposition into rank-one factors.

2) TUCKER: Tucker decomposition differs from CP decomposition, since the factorization provides a core tensor multiplied by a matrix along each dimension and the factor matrices have different dimensions. In this case, the approximation of the input tensor will be:

$$\mathcal{X} \approx \mathcal{G} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \boldsymbol{a}_p \circ \boldsymbol{b}_q \circ \boldsymbol{c}_r \equiv [\![\mathcal{G}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!] \quad (3)$$

where $\boldsymbol{A} \in \mathbb{R}^{I_1 \times P}$, $\boldsymbol{B} \in \mathbb{R}^{I_2 \times Q}$, $\boldsymbol{C} \in \mathbb{R}^{I_3 \times R}$ and $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$. $\mathcal{G}$ is the core tensor and determines the connections between the decomposed matrices. An illustration of the Tucker factorization is shown in Figure 2.

3) Nonnegative decomposition (NTF): It is an extension of nonnegative matrix factorization (NMF) to tensors. Nonnegativity is a constraint that can be useful to the factorization procedure, because the input data are positive numbers. When the resulting factorized matrices have no negative elements, it is easier to inspect them and analyse the scores they provide. A major difficulty NTF has is that it suffers from slow convergence speed due to the nonnegativity constrains, especially for large-scale problems. In [21], a fast low rank approximation is introduced and, additionally, bipolar noise is suppressed.

III. Multivariable Data Representation

The problem at hand consists of images downloaded from Flickr, which come with metadata. Motivated by [20], the present work uses the term similarities provided by the various decompositions and creates ranked relations between the tag terms in order to provide similar tags for images. The goal of this paper is to create a centroid (i.e., a codevector)

TABLE I. USER-IMAGE RELATIONS SLICE

$\boldsymbol{W}_U$	0
0	$\boldsymbol{W}_I$

that is representative of each tag for a specific image. The technique is based on [20]. However, here the basic nodes of the graph describing the system are the images together with the users who have uploaded them. Since the feature vectors, provided by the factorized matrices, correspond to the factors, it is logical to assume that these factors represent the data communities.

A. Dataset

The provided dataset was retrieved form Flickr. It consists of $N_{im} = 1292$ images, $N_{us} = 440$ users, $N_{tg} = 2366$ tags, $N_{geo} = 125$ geotags, and $N_{gr} = 1644$ user groups. Different kinds of pairwise relations of the dataset vertices are defined, namely, connections between users (user friendships), between images (image similarity), between users and user groups and between images and geotags. In this dataset, connections between three different items are described. A user who uploaded an image and gave it a corresponding tag or geotag, creating hyperedges. Accordingly, the use of 3-order tensor for this hyperedge representation becomes necessary.

B. Tensor representation of the data

As mentioned in previous sections, the nodes of the hyoergraph that is represented by the tensor will be users and images. Since the retrieved dataset consists of 440 users and 1392 images the representative tensor will have a $N \times K$ size where $N = N_{us} + N_{im} = 1732$ and K is the number of the tensor slices, (i.e., 6).

For the first slice $\boldsymbol{X}_{::1}$, the only relations captured are the users' friendships. The nonzero elements of this slice come from the user-user relation adjacency matrix, which has binary elements; one if a connection between users is established, zero otherwise.

The second slice $\boldsymbol{X}_{::2}$ represents the relations that users have, concerning the groups which they belong to. Since all slices must have same dimension, it is not possible to use the user-group adjacency matrix. Therefore, if $\boldsymbol{U} \in \mathbb{R}^{N_{us} \times N_{gr}}$ is the user-group matrix, then the square matrix providing the corresponding relations, will be:

$$\boldsymbol{W}_{gr} = \boldsymbol{U}\boldsymbol{U}^T \tag{4}$$

where $\boldsymbol{W}_{gr} \in \mathbb{R}^{N_{us} \times N_{us}}$. The second slice consists of matrix $\boldsymbol{W}_{gr}$ and the remaining elements are set to zero.

The third slice $\boldsymbol{X}_{::3}$ represents the user-image connection. These relations are derived from the user-image matrix $\boldsymbol{T} \in \mathbb{R}^{N_{us} \times N_{im}}$. One can obtain the matrices:

$$\boldsymbol{W}_U = \boldsymbol{T}\boldsymbol{T}^T \text{ and } \boldsymbol{W}_I = \boldsymbol{T}^T\boldsymbol{T}, \tag{5}$$

where $\boldsymbol{W}_U \in \mathbb{R}^{N_{us} \times N_{us}}$ and $\boldsymbol{W}_I \in \mathbb{R}^{N_{im} \times N_{im}}$. Then, matrices $\boldsymbol{W}_U$ and $\boldsymbol{W}_I$ constitute the third slice, as shown in Table I.

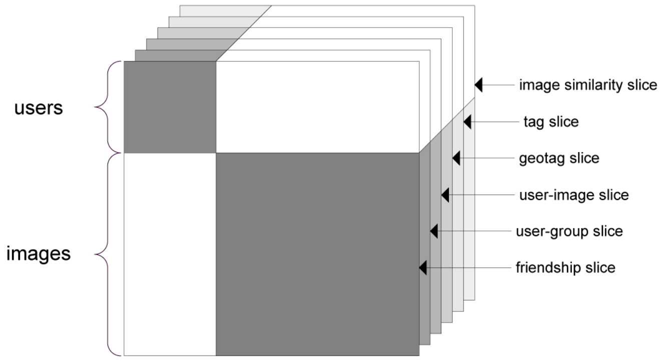

Fig. 3. Tensor representation

In a similar manner, the fourth slice is created. The particular slice provides the relations between users-images-geotags. From the initial matrices $\boldsymbol{U}_g \in \mathbb{R}^{N_{us} \times N_{geo}}$ and $\boldsymbol{I}_g \in \mathbb{R}^{N_{im} \times N_{geo}}$, which describe the user-geotag connections and image-geotag connections respectively, matrices $\boldsymbol{W}_{UG} \in \mathbb{R}^{N_{us} \times N_{us}}$ and $\boldsymbol{W}_{IG} \in \mathbb{R}^{N_{im} \times N_{im}}$ are obtained via:

$$\boldsymbol{W}_{UG} = \boldsymbol{U}_g{\boldsymbol{U}_g}^T \text{ and } \boldsymbol{W}_{IG} = \boldsymbol{I}_g{\boldsymbol{I}_g}^T, \tag{6}$$

Matrices $\boldsymbol{W}_{UG}$ and $\boldsymbol{W}_{IG}$ build the fourth slice $\boldsymbol{X}_{::4}$.

For the tag relation slice, namely the fifth slice $\boldsymbol{X}_{::5}$, a similar method is applied. Hence, considering matrices containing the relations of user-tag $\boldsymbol{U}_t \in \mathbb{R}^{N_{us} \times N_{tg}}$ and of image-tag $\boldsymbol{I}_t \in \mathbb{R}^{N_{im} \times N_{tg}}$, the user-tag-image slice is created, after the calculation of the corresponding square matrices has taken place i.e.,

$$\boldsymbol{W}_{UT} = \boldsymbol{U}_t{\boldsymbol{U}_t}^T \text{ and } \boldsymbol{W}_{IT} = \boldsymbol{I}_t{\boldsymbol{I}_t}^T, \tag{7}$$

where $\boldsymbol{W}_{UT} \in \mathbb{R}^{N_{us} \times N_{us}}$ and $\boldsymbol{W}_{IT} \in \mathbb{R}^{N_{im} \times N_{im}}$.

Finally, the sixth slice, that represents the image similarity is created from the image-image similarity matrix. The image similarity matrix $\boldsymbol{S}_I \in \mathbb{R}^{N_{im} \times N_{im}}$ contains the top five most similar images for each image. The 100 nearest neighbors to each image were identified using the GIST descriptors [22] and they were reduced to the 5 most similar images to the reference image, by using scaleinvariant feature transform (SIFT) [23]. More specifically, $\boldsymbol{S}_I$ has elements:

$$\boldsymbol{S}_I(i,j) = \begin{cases} 1 & \text{if image } j \text{ belongs to top 5 similar to image } i \\ 0 & \text{otherwise} \end{cases}$$

In order to obtain a symmetric image similarity matrix, the sum $(\boldsymbol{S}_I + \boldsymbol{S}_I^T)/2$ is calculated. In Figure 3, the complete tensor of the data is depicted. For more information about the dataset, refer to [27].

IV. EXPERIMENTS AND EVALUATION

Tensor $\mathfrak{X}$ is of dimension $N \times N \times 6$ and it is very sparse. The tensor decomposition techniques described in Section II are performed and the resulting factor matrices have been derived. Several toolboxes were used, namely the Tensor Toolbox [25], the TensorLab [26], and TADLAB [24]. Firstly, the dataset is divided into training and test sets. Considering the data representing tensor, and taking into account the fact that the problem tackled is tag recommendation, the training set is formed by retaining 75% of the tags of each image to zero. Latent representations for each user and each image that are spanned by matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ from Equations 2 and 3, provide user to user similarity as well as image to image similarity [20]. In addition, these feature vectors are used to analyze a body of work via centroids. Following this methodology, one can use the centroids to describe the communities of items, calculate similarity scores and, as a consequence, recommend tags, geotags, images etc., as is detailed next.

A. CANDECOMP/PARAFAC

More specifically centroids can be retrieved from matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ by computing the mean values of their columns. In this application, $R = 30$ is used, providing factorized matrices of dimension $N \times 30$. Let us suppose that one is interested to create a centroid vector $\underline{g} \in \mathbb{R}^{1\times 30}$ representative of one particular tag. All training images are identified, which bear the tag under study. For the rows of the factor matrix $\boldsymbol{A}$ which are associated to these images, we average across the columns of the factor matrix. If this procedure is repeated for matrix $\boldsymbol{B}$ as well, the similarity scores can be computed using tags as parameter as

$$s(tag) = \frac{1}{2}\underline{A}\underline{g}_A(tag) + \frac{1}{2}\underline{B}\underline{g}_B(tag). \qquad (8)$$

For all tags, corresponding centroids are provided and, by using cosine distances, a cosine similarity matrix between tags can be computed. Next, for queering images, the centroid scores for the training tags related to them are obtained and an updated tag centroid value, for the query image, is computed. Finally, a ranked vector of the most similar tags for the image at hand is provided and is compared to the ground truth. Hence, precision and recall measures can evaluate the performance of the method.

B. Tucker

Although Tucker factorization properties vary from those of CP decomposition, the same technique can be applied in order to get tagging recommendations. In this case, the selected value for the core tensor dimension was $R \times R \times R$ with $R = 6$. Now, in an analogous manner as with CP, one can use the Tucker decomposed matrices to create the communities and to receive the semantic feature spaces. Centroids can be obtained from the mean values of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ provided by Equation 3, exactly as with CP decomposition, however the vector dimension will be of size 6, because of the core tensor selection.

C. NTF

As claimed in previous sections, nonnegativity can be very helpful when analysing the component matrices. The input tensor that describes the hypergraph has nonnegative values. In fact all elements have either zero value or value one. Hence, NTF is also applicable and the results are compared to other techniques.

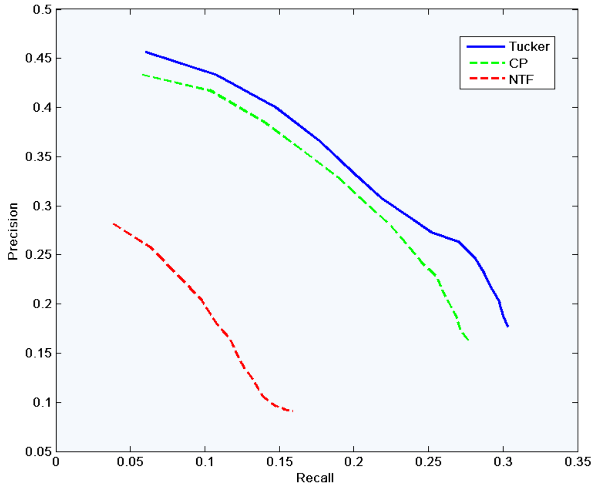

Fig. 4. Precision-recall curves for Tucker, CP, and NTF

D. Comparisons

From the comparisons between the different approaches of tensor factorization, it is easy to notice the advantages and the disadvantages of each method. Tucker approach outperforms the CP decomposition, because, the factorized matrices in CP are not orthogonal and this property appears to be essential in the specific problem. Concerning the nonnegative decomposition, it is seen that the expectation of getting more accurate recommendations by enforcing nonnegativity in the factorization is not fulfilled. The resulting precision - recall curves for all tensor decompositions are depicted in Figure 4.

V. CONCLUSION

The proposed methodology to image tagging applies ranked similarity cosine distances from representative centroid tags. It has been demonstrated that it is possible to represent by hypergraph vertices of different kind, i.e., users together with images, in order to obtain a low rank feature space associated to latent semantic communities. In addition, the similarities provided by centroid analysis of the factorized matrices are used to create tag scores and consequently are exploited for tag recommendation.

ACKNOWLEDGMENT

This research has been co-financed by the European Union (European Regional Development Fund - ERDF) and Greek national funds through the Operation Program “Competitiveness-Cooperation 2011” - Research Funding Program: SYN-10-1730-ATLAS.

REFERENCES

[1] X. Qi and Y. Han, *Incorporating multiple SVMs for automatic image annotation*, Pattern Recognition vol. 40, no. 2, pp. 728-741, 2007.

[2] S. Kim, S. Park, and M. Kim, *Image classication into object/non-object classes*, Proceedings of the International Conference on Image and Video Retrieval, pp. 393-400, 2004.

[3] S. Feng, R. Manmatha, and V. Lavrenko, *Multiple Bernoulli relevance models for image and video annotation*, IEEE Computer Society Conf. Computer Vision and Pattern Recognition, vol. 2, pp. II1002-II1009, 2004.

[4] D. Zhang, M. Islam, and G. Lu, *A review on automatic image annotation techniques*, Pattern Recognition, 2012.

[5] A. Milicevic, A. Nanopoulos, and M. Ivanovic, *Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions*, Artificial Intelligence Rev., vol. 33, no. 3, pp. 187-209., 2010.

[6] X. Zhang, X. Zhao, Z. Li, J. Xia, R. Jain, and W. Chao, *Social image tagging using graph-based reinforcement on multi-type interrelated objects*, Signal Processing, vol. 93, no. 8, pp. 2178-2189, 2013.

[7] Z. Guan, J. Bu, Q. Mei, C. Chen, and C. Wang, *Personalized tag recommendation using graph-based ranking on multi-type interrelated objects*, in Proc. 32nd ACM SIGIR Int. Conf. Research and Development in Information Retrieval, 2009.

[8] R. Liu, Y. Wang, H. Yu, and S. Naoi , *A Renewed Image Annotation Baseline by Image Embedding and Tag Correlation*, ICPR, 2012.

[9] X. Li, C. Snoek, and M. Worring, *Learning social tag relevance by neighbor voting*, IEEE Trans. Multimedia, vol. 11, no. 7, 1310-1322 , 2009.

[10] Y. Han, *Multi-label boosting for image annotation by structural grouping sparsity*, in Proc. ACM Multimedia Conf, pp. 15-24, 2010.

[11] T. Mei, Y. Wang, X. Hua, S. Gong,and S. Li, *Coherent image annotation by learning semantic distance*, CVPR, pp. 1-8, 2008.

[12] X. Wang, L. Zhang, X. Li, and W. Ma, *Annotating images by mining image search results*, IEEE Trans. Pattern Anal. Mach. Intell. 30(11), 1919-1932, 2008.

[13] C. Wang, S. Yan, L. Zhang and H. Zhang, *Multi-label sparse coding for automatic image annotation*, in Proc. of IEEE Int. Conf. Computer Vision and Pattern Recognition, pp. 1643-1650. Florida, USA , 2009.

[14] S. Moran and V. Lavrenko, *Sparse Kernel Learning for Image Annotation*, in Proceedings of International Conference on Multimedia Retrieval, New York, NY, USA, 2014, pp. 113:113113:120.

[15] W. Liu, D. Tao, J. Cheng, and Y. Tang, *Multiview Hessian discriminative sparse coding for image annotation*, Computer Vision and Image Understanding, vol. 118, pp. 5060, Jan. 2014.

[16] M. Kalayeh, H. Idrees, and M. Shah, *NMF-KNN: Image Annotation using Weighted Multi-view Non-negative Matrix Factorization*, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 184191.

[17] P. Symeonidis, A. Nanopoulos and Y. Manolopoulos, *Tag Recommendations based on tensor dimensionality reduction*, in Proc. ACM Conf. Recommender Systems. New York, pp. 43-50, 2008.

[18] S. Rendle , B. Marinho, A. Nanopoulos and L. Thieme, *Learning optimal ranking with tensor factorization for tag recommendation*, in Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 727-736, 2008.

[19] T. Kolda and B. Bader, *Tensor decompositions and applications*, SIAM Rev., 51(3):455–500, 2009.

[20] D. Dunlavy, T. Kolda and W. Kegelmeyer *Multilinear algebra for analyzing data with multiple linkages*, in *Graph Algorithms in the Language of Linear Algebra*, Fundamentals of Algorithms, SIAM, Philadelphia, pp. 85–114, 2011.

[21] G. Zhou A. Cichocki and S. Xie, *Senior Member IEEE*, *Fast Nonnegative Matrix/Tensor Factorization Based on Low-Rank Approximation*, IEEE Transactions on Signal Processing, Vol.60, no.6, pp.2928–2940, 2012.

[22] A. Oliva and A. Torralba, *Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope*, Int. J. of Computer Vision, Vol. 42, Issue 3, pp 145–175, 2001.

[23] D. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, Int. J. Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[24] G. Zhou and A. Cichocki , *Matlab Toolbox for Tensor Decomposition and Analysis Ver1.1*, Available: http://bsp.brain.riken.jp/TDALAB/, 2013.

[25] B. Bader and T. Kolda, *MATLAB Tensor Toolbox Version 2.5*, Available: http://www.sandia.gov/ tgkolda/TensorToolbox/, 2013.

[26] L. Sorber, M. Van Barel and L. De Lathauwer, *Tensorlab v1.0*, Available: http://esat.kuleuven.be/sista/tensorlab/, 2013.

[27] K. Pliakos and C. Kotropoulos, *Simultaneous image tagging and geo-location prediction within hypergraph ranking framework*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.6894-6898, 4-9 May 2014.