WEIGHT ESTIMATION IN HYPERGRAPH LEARNING

Konstantinos Pliakos and Constantine Kotropoulos

Department of Informatics, Aristotle University of Thessaloniki
Thessaloniki, 54124, Greece
Email: {kpliakos, costas}@aiia.csd.auth.gr

ABSTRACT

The unremitting rising popularity of social media has led to an exponential increase in web activity as manifested by the vast volume of uploaded images. This boundless volume of image data has triggered the interest in image tagging. Here, an efficient hypergraph weight estimation scheme is proposed that improves the accuracy of image tagging, using hypergraph learning. The proposed method models high-order relations between hypergraph vertices (i.e., users, user social groups, tags, geo-tags, and images) by hyperedges. The information captured by the hyperedges is efficiently distilled by estimating the hyperedge weights. Experiments conducted on a dataset crawled from $Flickr$ demonstrate the effectiveness of the proposed approach. Specifically, an average precision of 91% at 26% recall has been achieved for image tagging.

***Index Terms*—** Image tagging, Hypergraph learning, Hyperedge weight learning.

1. INTRODUCTION

Nowadays, the overwhelming number of data uploaded to the web in conjunction with the ever rising popularity of social media sharing platforms has led to an indisputable need for efficient tagging methods. Image tagging is a crucial procedure, affecting considerably both the retrieval accuracy and the organization of the images uploaded to the web. Popular social media sharing platforms, such as *Flickr*[1], *Picasa Web Album*[2], or *Instagram*[3], enable users to describe the content of images by tagging them. However, quite often, the user provided tags are far from being accurate, or may be redundant. Despite the research effort made so far, there are persisting problems, such as achieving satisfactory efficiency and accuracy. Consequently, an accurate and efficient image tagging model is of crucial importance.

Many works were focused on image tagging, using graphs or hypergraphs. In [1], users, tags, and images were modelled in a graph, harnessing the information distilled from the tags, the image visual attributes, and the social friendship relations among the users. Image tagging was treated in a "query and ranking" manner and a graph-based reinforcement algorithm for interrelated multi-type objects was proposed. A visual image similarity graph and an image-tag bipartite graph were fused in a unified graph in [2]. A random walk model was proposed, employing a fusion parameter to regularize the influence between the visual and textual information provided by the image visual content and the tags, respectively. In [3], image tagging was addressed within a hypergraph ranking canvas by enforcing group sparsity constraints. Multi-label image annotation was formulated as a regression model with a regularized penalty, exploiting the structural group sparsity in [4].

Hypergraphs have also been employed on various machine learning and retrieval tasks, harnessing their higher-order relation modelling. In [5], images were taken as vertices in a probabilistic hypergraph with hyperedges, linking images according to their visual content. In [6], a hypergraph was used for classification, modelling the images by their visual attributes. Hypergraph learning was also applied to social image search [7, 8, 9]. Furthermore, a music recommendation method was developed based on hypergraph learning, exploiting social media information and audio signal similarities [10]. A hypergraph-based news recommendation model was proposed in [11], encapsulating both user behaviour and news content information.

Here, the problem of image tagging is addressed within a hypergraph learning framework. The hypergraph has vertices made by concatenating different kinds of objects (users, user groups, geo-tags, tags, images) and hyperedges linking these vertices [12, 13, 14, 15]. In contrast to the edges of a simple graph, the hyperedges link more than two vertices, capturing higher order relations, such as the triple relation between a user, a tag, and an image.

Motivated by [7], an effective hyperedge weight learning scheme is proposed, treating each hyperedge in a different manner. The novelty of this paper is in the analytical solution of the optimization problem, which leads to the weight estimation. Here, the method in [7] is revisited by taking into account the vertex degree matrix that was omitted in [7], including the topology of the hypergraph as is captured by the incidence matrix of the hypergraph. This way, the hyperedges capturing more informative relations are better exploited and the impact of the less informative hyperedges is reduced. The superiority of the proposed method is demonstrated by its application to image tagging, extending the work presented in [3]. Experiments conducted on a dataset from $Flickr$ indicate the effectiveness of the proposed method, yielding an average precision of 91% at 26% recall for image tagging.

The outline of the paper is as follows. In Section 2, the general hypergraph model is introduced and the ranking on a hypergraph is detailed. The hyperedge weight learning method is addressed in Section 3. In Section 4, the dataset is described. The hypergraph construction is explained in Section 5. Experimental results are presented in Section 6, demonstrating the merits of the proposed method. Conclusions are drawn in Section 7.

2. HYPERGRAPH MODEL

In the following, $|\cdot|$ denotes set cardinality, $\|.\|$ is the ℓ_2 norm of a vector, and $\mathbf{I}$ is the identity matrix of compatible dimensions. Let

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

$G(V,E,w)$ denote a hypergraph with set of vertices V and set of hyperedges E to which a real weight function w is assigned. The vertex set V is made by concatenating sets of objects of different type (users, social groups, geo-tags, tags, images). These vertices and hyperedges form a $|V| \times |E|$ incidence matrix $\mathbf{H}$ with elements $H(v,e) = 1$ if $v \in e$ and 0 otherwise. The vertex and hyperedge degrees are obtained by:

$$\left.\begin{array}{l} \delta(v) = \sum_{e \in E} w(e) H(v,e) \\ \delta(e) = \sum_{v \in V} H(v,e) \end{array}\right\}. \quad (1)$$

The following diagonal matrices are defined: the vertex degree matrix $\mathbf{D}_u$ of size $|V| \times |V|$, the hyperedge degree matrix $\mathbf{D}_e$ of size $|E| \times |E|$, and the $|E| \times |E|$ matrix $\mathbf{W}$ containing the hyperedge weights.

Let $\mathbf{A} = \mathbf{D}_u^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_u^{-1/2}$. $\mathbf{A}$ is a symmetric matrix, as the diagonal matrices $\mathbf{W}$ and $\mathbf{D}_e^{-1}$ commute in multiplication. Then, $\mathbf{L} = \mathbf{I} - \mathbf{A}$ is known as Zhou's normalized Laplacian of the hypergraph [13]. The elements of $\mathbf{A}$, $A(j,i)$, indicate the relatedness between the vertices j and i. To perform clustering on a hypergraph, one is seeking for a real-valued ranking vector $\mathbf{f} \in \mathbb{R}^{|V|}$, minimizing the cost function:

$$\Omega(\mathbf{f}) = \mathbf{f}^T\mathbf{L}\mathbf{f}. \quad (2)$$

That is, one requires all vertices with the same value in the ranking vector $\mathbf{f}$ to be strongly connected [16]. $\Omega(\mathbf{f})$ is small, if the vertices with high affinities are assigned the same label [16]. For instance, two images are probably similar, if they are linked with many common tags. The aforementioned optimization problem was extended to a recommendation problem by including the ℓ_2 regularization norm between the ranking vector $\mathbf{f}$ and a query vector $\mathbf{y} \in \mathbb{R}^{|V|}$ [10]. This guarantees that the ranking vector does not differ too much from the initial query. The function to be minimized is then expressed as

$$Q(\mathbf{f}) = \Omega(\mathbf{f}) + \vartheta\, ||\mathbf{f} - \mathbf{y}||^2 \quad (3)$$

where ϑ is a positive regularizing parameter. The best ranking vector, $\mathbf{f}^* = \arg\min_{\mathbf{f}} Q(\mathbf{f})$, is found to be [10]:

$$\mathbf{f}^* = \frac{\vartheta}{1+\vartheta}\left(\mathbf{I} - \frac{1}{1+\vartheta}\mathbf{A}\right)^{-1}\mathbf{y}. \quad (4)$$

3. HYPEREDGE WEIGHT UPDATING

Clearly, all the hyperedges do not have the same effect on the learning procedure. Some relations captured by hyperedges are not as informative as others. For example, two users might be friends without having common interests or a user might have assigned an irrelevant tag to an image. Thus, the hypergraph learning is enhanced by optimizing the hyperedge weights. Let $n = |E|$ and $\mathbf{w} = (w_1, w_2, \cdots, w_n)^T$ be formed by the elements lying in the main diagonal of $\mathbf{W}$. Moreover, we enforce $\mathbf{1}_n^T\mathbf{w} = 1$. By adding an ℓ_2 norm regularizer on $\mathbf{w}$ and optimizing for both $\mathbf{w}$ and $\mathbf{f}$, the following minimization function is defined:

$$\underset{\mathbf{f},\mathbf{w}}{\text{argmin}} \left\{Q(\mathbf{f}) + \kappa||\mathbf{w}||^2\right\} \qquad \text{s.t. } \mathbf{1}_n^T\mathbf{w} = 1. \quad (5)$$

The method is illustrated in Fig.1. An alternating minimization of (5) starts with a fixed $\mathbf{w}$ and optimizes $\mathbf{f}$ as in (4). Next, $\mathbf{f}$ is fixed

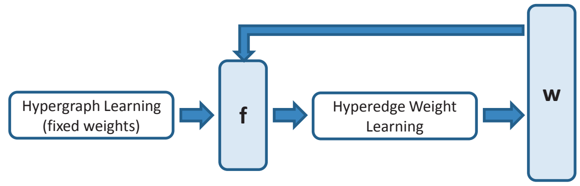

Fig. 1. Description of the hyperedge weight learning method.

and $\mathbf{w}$ is optimized. Having fixed $\mathbf{f}$, the optimization w.r.t. $\mathbf{w}$ is read as:

$$\underset{\mathbf{w}}{\text{argmin}} \left\{\mathbf{f}^T\mathbf{L}\mathbf{f} + \kappa||\mathbf{w}||^2\right\} \qquad \text{s.t. } \mathbf{1}_n^T\mathbf{w} = 1. \quad (6)$$

The Lagrangian function of the optimization problem is:

$$\begin{aligned} \Psi(\mathbf{w}, c) &= \mathbf{f}^T\mathbf{L}\mathbf{f} + \kappa\mathbf{w}^T\mathbf{w} + c(\mathbf{1}_n^T\mathbf{w} - 1) \\ &= \mathbf{f}^T\left(\mathbf{I} - \mathbf{D}_u^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_u^{-1/2}\right)\mathbf{f} \\ &\quad + \kappa\mathbf{w}^T\mathbf{w} + c(\mathbf{1}_n^T\mathbf{w} - 1). \end{aligned} \quad (7)$$

Let $\boldsymbol{\Lambda} = \mathbf{D}_u^{-1/2}\mathbf{H}$. The partial derivatives of Ψ w.r.t. w_i, $i = 1, 2, \cdots, n$ are given by:

$$\frac{\partial \Psi}{\partial w_i} = \frac{\partial}{\partial w_i}\left(-\mathbf{f}^T\boldsymbol{\Lambda}\mathbf{W}\mathbf{D}_e^{-1}\boldsymbol{\Lambda}^T\mathbf{f}\right) + 2\kappa w_i + c = 0. \quad (8)$$

Solving (8) w.r.t. w_i, we obtain:

$$w_i = \frac{1}{2\kappa}\left[\frac{\partial}{\partial w_i}\left(\mathbf{f}^T\boldsymbol{\Lambda}\mathbf{W}\mathbf{D}_e^{-1}\boldsymbol{\Lambda}^T\mathbf{f}\right) - c\right] \quad (9)$$

and by substituting (9) into the constraint $\mathbf{1}_n^T\mathbf{w} = 1$, the Lagrange multiplier is determined:

$$c = \frac{1}{n}\left[\mathbf{1}_n^T\frac{\partial}{\partial w_i}\left(\mathbf{f}^T\boldsymbol{\Lambda}\mathbf{W}\mathbf{D}_e^{-1}\boldsymbol{\Lambda}^T\mathbf{f}\right) - 2\kappa\right]. \quad (10)$$

The partial derivative in (9) and (10) is analysed as:

$$\begin{aligned} \frac{\partial}{\partial w_i}\left(\mathbf{f}^T\boldsymbol{\Lambda}\mathbf{W}\mathbf{D}_e^{-1}\boldsymbol{\Lambda}^T\mathbf{f}\right) &= \mathbf{f}^T\frac{\partial(\boldsymbol{\Lambda})}{\partial w_i}\mathbf{W}\mathbf{D}_e^{-1}\boldsymbol{\Lambda}^T\mathbf{f} \\ + \mathbf{f}^T\boldsymbol{\Lambda}\frac{\partial(\mathbf{W})}{\partial w_i}\mathbf{D}_e^{-1}\boldsymbol{\Lambda}^T\mathbf{f} &+ \mathbf{f}^T\boldsymbol{\Lambda}\mathbf{W}\mathbf{D}_e^{-1}\frac{\partial(\boldsymbol{\Lambda}^T)}{\partial w_i}\mathbf{f} \\ &= \mathbf{f}^T D_e^{-1}(i,i)\boldsymbol{\Lambda}_i{\boldsymbol{\Lambda}_i}^T\mathbf{f} - \mathbf{f}^T\boldsymbol{\Xi}_i\mathbf{f}, \end{aligned} \quad (11)$$

where $\boldsymbol{\Lambda}_i \in \mathbf{R}^{|V|}$ is the i-th column of $\boldsymbol{\Lambda}$. $\boldsymbol{\Xi}_i = \text{diag}(\boldsymbol{\Lambda}_i)\mathbf{D}_u^{-1/2}\mathbf{A}$. Observe that $\boldsymbol{\Xi}_i$ is a symmetric matrix and $\text{diag}(\boldsymbol{\Lambda}_i)$ is a $|V| \times |V|$ diagonal matrix having $\boldsymbol{\Lambda}_i$ in its main diagonal. By substituting (11) into (10) and (9), we obtain the following closed expressions:

$$\begin{aligned} c &= \frac{1}{n}\left\{\mathbf{f}^T\sum_{i=1}^{n}\left[D_e^{-1}(i,i)\boldsymbol{\Lambda}_i{\boldsymbol{\Lambda}_i}^T - \boldsymbol{\Xi}_i\right]\mathbf{f} - 2\kappa\right\} \\ w_i &= \frac{1}{2\kappa}\left\{\mathbf{f}^T\left[D_e^{-1}(i,i)\boldsymbol{\Lambda}_i{\boldsymbol{\Lambda}_i}^T - \boldsymbol{\Xi}_i\right]\mathbf{f} - c\right\}. \end{aligned} \quad (12)$$

Having optimized the hyperedge weights, $\mathbf{f}$ is re-optimized. The proposed method is summarized in Algorithm 1.

Algorithm 1 Image tagging via hyperedge weight learning.

Input: The objects (users, images, social groups, geo-tags, and tags) and their relations.

Output: The ranking vector $\mathbf{f}$.

1 Form matrices $\mathbf{H}$, $\mathbf{D}_e$, $\mathbf{D}_u$, and $\mathbf{W}$, having initialized the hyperedge weights w_i.

2 Compute the affinity matrix $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$. Set the regularization parameter ϑ and the query vector $\mathbf{y} \in \mathbb{R}^{|V|}$.

3 Find result ranking vector $\mathbf{f} \in \mathbb{R}^{|V|}$ as in (4).

4 Optimize Eq.(6) to find the hyperedge weights as in (12).

5 Having found the new hyperedge weights, update $\mathbf{D}_u$ and $\mathbf{W}$.

6 Repeat the steps $2-5$ until convergence to find the final ranking vector $\mathbf{f}$.

Table 1. Dataset objects, notations, and counts.

Object	Notation	Count
Images	Im	1292
Users	U	440
User Groups	Gr	1644
Geo-tags	Geo	125
Tags	Ta	2366

4. DATASET DESCRIPTION

For evaluation purposes, an image dataset was collected from $Flickr$. It contains both indoor and outdoor medium sized photos of popular Greek landmarks, including city scenes and landscapes. Using $FlickrApi$[4], a large set of "geotagged" images was downloaded along with valuable information related to them (id, title, owner, latitude, longitude, tags, image views). Then, the dataset was filtered based on image views (i.e., the times that the specific image has been seen in $Flickr$) and owner's uploading statistics. At this point, it was assumed that images with many views normally depict worth seeing landmarks and owners (users) with many uploaded images were active ones, possessing many social relations (friends, social groups). The image owners were the users in the dataset. Then, corresponding social information (friends, social groups) was crawled and only the groups that had at least 5 owners from the dataset as members were kept. The specific cardinalities are summarized in Table 1.

In order to form a proper set of tags, all characters were converted to lower case, unreadable symbols and redundant information were removed. Next, a vocabulary of unique words was generated along with their frequencies. Terms with frequency less than 2 occurrences were removed from the set of tags and the vocabulary. Finally, spelling mistakes were corrected and any morphological variations merged using the Edit Distance [17].

Having computed pairwise distances according to the "Haversine formula"[5], geo-tags were clustered into 125 distinct clusters using hierarchical clustering.

[4]http://www.flickr.com/services/api

[5]http://www.movable-type.co.uk/scripts/latlong.html

Table 2. The structure of the hypergraph incidence matrix $\mathbf{H}$ and its sub-matrices.

$E^{(1)}$	$E^{(2)}$	$E^{(3)}$	$E^{(4)}$	$E^{(5)}$	$E^{(6)}$
0	0	$ImE^{(3)}$	$ImE^{(4)}$	$ImE^{(5)}$	$ImE^{(6)}$
$UE^{(1)}$	$UE^{(2)}$	$UE^{(3)}$	$UE^{(4)}$	$UE^{(5)}$	0
0	$GrE^{(2)}$	0	0	0	0
0	0	0	$GeoE^{(4)}$	0	0
0	0	0	0	$TaE^{(5)}$	0

5. HYPERGRAPH CONSTRUCTION

The hypergraph structure is displayed in Table 2. The vertex set is defined as $V = Im \cup U \cup Gr \cup Geo \cup Ta$. The incidence matrix of the hypergraph $\mathbf{H}$ has size 5867×30924 elements. In the following, the initial weights of the hyperedges are set equal to $\frac{1}{n}$, where n is the volume of the hyperedges. The dataset has captured 2276 friendship relations and 19127 tagging ones.

$E^{(1)}$ represents a pairwise friendship relation between users. The incidence matrix of the hypergraph $UE^{(1)}$ has size 440×2276 elements.

$E^{(2)}$ represents a user group. It contains all the vertices of the corresponding users as well as the ones corresponding to the user group. The incidence matrix of the hypergraph $UE^{(2)} - GrE^{(2)}$ has size $(440 + 1644) \times 1644$ elements.

$E^{(3)}$ contains a user and an uploaded image, representing a user-image possession relation. Each image has only one owner. The incidence matrix of the hypergraph $UE^{(3)} - ImE^{(3)}$ has size $(440 + 1292) \times 1292$ elements.

$E^{(4)}$ captures a geo-location relation. This hyperedge set contains triplets of Im, U, and Geo. The incidence matrix of the hypergraph $ImE^{(4)} - UE^{(4)} - GeoE^{(4)}$ has size $(1292+440+125) \times 125$ elements.

$E^{(5)}$ also contains triplets, Im, U, and Ta. Each hyperedge represents a tagging relation. The incidence matrix of the hypergraph $ImE^{(5)} - UE^{(5)} - TaE^{(5)}$ has size $(1292 + 440 + 2366) \times 19127$ elements.

$E^{(6)}$ contains pairs of vertices, which represent two images. Both global and local features were used to determine visual relations between images. Firstly, the 100 nearest neighbors to each image were identified using the GIST descriptors [18] and they were reduced to the 5 most similar images to the reference image, by using scale-invariant feature transform (SIFT) [19]. The incidence matrix of the hypergraph $ImE^{(6)}$ has size 1292×6460.

The query vector $\mathbf{y}$ is initialized by setting the entry corresponding to the test image im and its owner o to 1. The tags ta connected to this image are set equal to $A(im, ta)$. The objects corresponding to gr and geo associated to the image owner o are set equal to $A(o, gr)$ and $A(o, geo)$, respectively. The query vector $\mathbf{y}$ has a length of 5867 elements. During testing, the tags contained in the test set were not included in the training procedure.

The ranking vector $\mathbf{f}^*$ has the same size and structure as $\mathbf{y}$. The values corresponding to tags are used for image tagging with the top ranked tags being recommended for the test image.

6. EXPERIMENTS

The averaged Recall-Precision, MAP, and F_1 measure are used as figures of merit. Precision is defined as the number of correctly recommended tags divided by the number of all recommended tags. Recall is defined as the number of correctly recommended tags divided by the number of all tags the user has actually set. The F_1 measure is the weighted harmonic mean of precision and recall, which measures the effectiveness of tagging when treating precision and recall as equally important.

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \quad (13)$$

The MAP is the mean value of the Average Precision (AP). The AP is the average of precisions computed at the point of each correctly retrieved item:

$$AP = \frac{\sum_i^{Num} Precision@i \cdot true_i}{cNum} \quad (14)$$

where $Precision@i$ is the precision at ranking position i, Num is the number of retrieved items, $cNum$ is the number of correctly retrieved items, and $true_i = 1$, if the item at position i is correctly retrieved. Let us refer to the ranking obtained by the proposed approach as Image Tagging on Hypergraph with Hyperedge Weight Estimation (ITH-HWE) and that obtained by (4) as Image Tagging on Hypergraph (ITH). The ranking obtained by the approach in [7] is denoted as HG-WE.

For evaluation purposes, a test set containing the 25% of the tags and a training set containing the remaining 75% are defined. The results of the image tagging are demonstrated in Fig. 2, in which the averaged Recall-Precision curves are plotted. These curves were obtained by averaging the Recall-Precision curves over 1186 images with at least 4 tags. To calculate the recall and precision, the 15 top ranked tags are being recommended to any test image. In Fig.2, image tagging on hypergraph was performed by initializing the hyperedge weights to $\frac{1}{n}$. It is demonstrated that by applying the hyperedge weight learning scheme, tagging precision is improved considerably. The ITH-HWE outperforms the ITH significantly, reaffirming the effectiveness of the proposed method. In this experiment, the HG-WE fails to yield competitive results.

In Fig.3, image tagging was performed by initializing the hyperedge weights unequally, as in [3]. The results obtained by the ITH-HWE outperform the other methods, reaffirming the superiority of the ITH-HWE over both the ITH and the HG-WE. It is seen that the initialization of the algorithm affects the image tagging results.

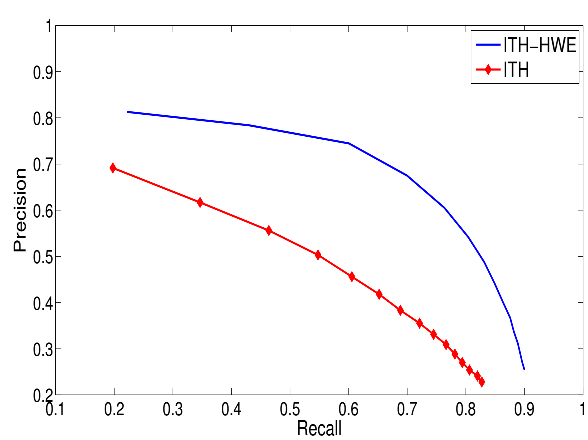

Fig. 2. Averaged Recall-Precision curves for the ITH and ITH-HWE.

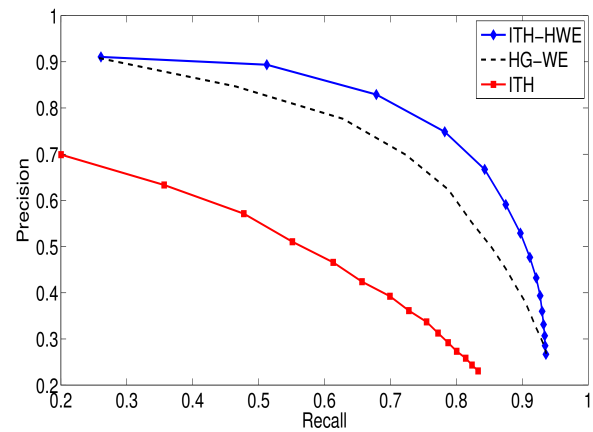

Fig. 3. Averaged Recall-Precision curves for the compared methods.

In Table 3, the averaged F_1 measure at ranking positions 1, 2, 5, 10 and the MAP are listed for the ITH and the ITH-HWE. Additional experiments were conducted on personalized image recommendation (IRH) and geo-referenced image recommendation (GIRH), as were described in [20]. In all experiments, by employing the ITH-HWE the results are significantly improved, validating the merits of the proposed hyperedge weight learning scheme.

Table 3. MAP and F_1 measure for the compared methods.

	$F_1@1$	$F_1@2$	$F_1@5$	$F_1@10$	MAP
ITH	0.307	0.444	0.520	0.440	0.679
ITH-HWE	0.349	0.556	0.675	0.517	0.829
IRH	0.422	0.590	0.481	0.338	0.734
IRH-HWE	0.527	0.801	0.587	0.379	0.897
GIRH	0.301	0.434	0.448	0.390	0.609
GIRH-HWE	0.522	0.828	0.720	0.574	0.983

7. CONCLUSION AND FUTURE WORK

In this paper, an efficient hyperedge weight learning scheme has been proposed. Image tagging has been addressed within a unified hypergraph learning framework, exploiting hypergraph structure multi-link relations. The experiments conducted on a collection of images related to Greek sites have demonstrated the effectiveness of the proposed method. Needless to say that the developed framework can be accommodated in the tagging, retrieval, or recommendation of any multimedia (e.g., music, video) or even the fusion between them. The incremental update of an already trained hypergraph learning model, reducing the $O(|E|^3)$ complexity of the method, could be a topic of future research.

8. ACKNOWLEDGMENTS

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: 11SYN-10-1730-ATLAS.

9. REFERENCES

[1] 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.

[2] H. Ma, J. Zhu, M. R. T. Lyu, and I. King, "Bridging the semantic gap between image contents and tags," *IEEE Trans. Multimedia*, vol. 12, no. 5, pp. 462–473, 2010.

[3] K. Pliakos and C. Kotropoulos, "Simultaneous image tagging and geo-location prediction within hypergraph ranking framework," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2014, pp. 6944–6948.

[4] F. Wu, Y. Han, Q. Tian, and Y. Zhuang, "Multi-label boosting for image annotation by structural grouping sparsity," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 15–24.

[5] Y. Huang, Q. Liu, S. Zhang, and D. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, 2010, pp. 3376–3383.

[6] J. Yu, D. Tao, and M. Wang, "Adaptive hypergraph learning and its application in image classification," *IEEE Trans. Image Processing*, vol. 21, no. 7, pp. 3262–3272, 2012.

[7] Y. Gao, M. Wang, Z. J. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Trans. Image Processing*, vol. 22, no. 1, pp. 363–376, 2013.

[8] Y. Zhuang, Y. Liu, F. Wu, Y. Zhang, and J. Shao, "Hypergraph spectral hashing for similarity search of social image," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1457–1460.

[9] Y. Liu, J. Shao, J. Xiao, F. Wu, and Y. Zhuang, "Hypergraph spectral hashing for image retrieval with heterogeneous social contexts," *Neurocomputing*, vol. 119, pp. 49–58, 2013.

[10] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, Z. Lijun, and X. He, "Music recommendation by unified hypergraph: Combining social media information and music content," in *Proc. ACM Conf. Multimedia*, 2010, pp. 391–400.

[11] L. Li and T. Li, "News recommendation via hypergraph learning: Encapsulation of user behavior and news content," in *Proc. ACM Int. Conf. on Web Search and Data Mining*, 2013, pp. 305–314.

[12] C. Berge and E. Minieka, *Graphs and Hypergraphs*, vol. 7, North-Holland, Amsterdam, 1973.

[13] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Advances in Neural Information Processing Systems*, 2006, pp. 1601–1608.

[14] Z. Yu, S. Tang, Y. Zhang, and J. Shao, "Image ranking via attribute boosted hypergraph," in *Proc. 13th Pacific-Rim Conf. Advances in Multimedia Information Processing*, 2012, pp. 779–789.

[15] J. Xu, V. Singh, Z. Guan, and B. S. Manjunath, "Unified hypergraph for image ranking in a multimodal context," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2012, pp. 2333–2336.

[16] S. Agarwal, K. Branson, and S. Belongie, "Higher order learning with graphs," in *Proc. 23rd Int. Conf. Machine Learning*, 2006, pp. 17–24.

[17] E. S. Ristad and P. Yianilos, "Learning string-edit distance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, 1998.

[18] A. Oliva and A. Torralba, "Building the GIST of a scene: The role of global image features in recognition," *Progress in Brain Research*, vol. 155, pp. 23–36, 2006.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] K. Pliakos and C. Kotropoulos, "Personalized and georeferenced image recommendation using unified hypergraph learning and group sparsity optimization," in *Proc. IEEE Int. Symp. Communications, Control, and Signal Processing*, 2014, pp. 323–326.