ADAPTIVE ALGORITHMS FOR HYPERGRAPH LEARNING

Aikaterini Chasapi, Constantine Kotropoulos, and Konstantinos Pliakos

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

ABSTRACT

Social media sharing platforms enable image content as well as context information (e.g., user friendships, geo-tags assigned to images) to be jointly analyzed in order to achieve accurate image annotation or successful image recommendation. The context information is expressed frequently in terms of high-order relations, such as the relations among users, tags, and images. Hypergraphs can model the aforementioned high-order relations between their vertices (i.e., users, user social groups, tags, geo-tags, and images) by hyperedges, whose influence can be assessed by properly estimating their weights. Here, an efficient adaptive hypergraph weight estimation is proposed for image tagging. In particular, both equality and inequality constraints enforced during hypergraph learning are taken into account and an efficient adaptation step selection using the Armijo rule is proposed. Experiments conducted on a dataset demonstrate the superior performance of the proposed approach compared to the state-of-the-art.

Index Terms— Image tagging, Hypergraph learning, Gradient search algorithms.

1. INTRODUCTION

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 tags provided by the users are inaccurate or redundant. The aforementioned fact has motivated research toward automated image tagging. Despite the research effort made so far, achieving satisfactory efficiency and accuracy still remain open issues.

In image tagging, the context information (e.g., user friendships, geo-tags assigned to images) is of great importance and can be expressed in terms of high-order relations. For example, the relation engaging a user, an image, and a tag is a third-order relationship. Hypergraphs are suitable to model high-order relations between heterogeneous vertices, obtained by concatenating different kinds of objects (i.e., users, user social groups, tags, geo-tags, and images), with hyperedges [1, 2]. The influence of each hyperedge can be assessed by properly estimating its weight [3, 4]. The hypergraphs enjoy theoretical interest and find a wide range of applications in mathematics [5], databases, data mining, biology, complex network modeling, multimedia search to mention a few [6].

Image tagging was treated in a "query and ranking" manner and a graph-based reinforcement algorithm for interrelated multi-type objects was proposed in [7]. A random walk model was proposed in [8], employing a fusion parameter to regularize the influence between the visual and textual information. In [9], 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 [10]. Hypergraph learning was also applied to social image search [11, 12].

Here, an efficient *adaptive* hypergraph weight estimation scheme is proposed for image tagging, extending the previous work [4]. The novelty of this paper is in the incorporation of equality and inequality constraints within the optimization problem related to hypergraph learning and the derivation of a gradient search method for its solution from first principles. In addition, an efficient adaptation step selection is proposed, using the Armijo rule. Experiments conducted on a dataset of images related to popular Greek landmarks demonstrate the superior performance of the proposed approach compared to the state-of-the-art.

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

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 $G(V, E, w)$ 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 $\delta(v) = \sum_{e \in E} w(e)H(v, e)$ and $\delta(e) = \sum_{v \in V} H(v, e)$, respectively. The following diagonal matrices are defined: the vertex degree matrix $\mathbf{D}_v$ 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 $\mathbf{A} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}$. $\mathbf{A}$ 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 $\Omega(\mathbf{f}) = \mathbf{f}^T\mathbf{L}\mathbf{f}$. That is, one requires all vertices with the same value in the ranking vector $\mathbf{f}$ to be strongly

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

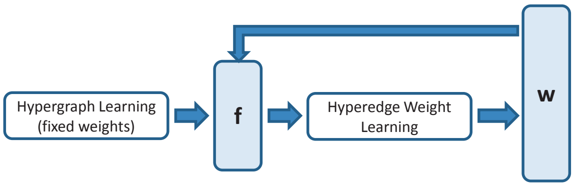

Fig. 1. Description of the hyperedge weight learning method.

connected [14]. For instance, two images are probably similar, if they are linked with many common tags.

The just described optimization problem was extended to a ranking problem by including the ℓ_2 regularization norm between the ranking vector $\mathbf{f}$ and a query vector $\mathbf{y} \in \mathbb{R}^{|V|}$ [15, 16] This guarantees that the ranking vector does not differ too much from the initial query. The function to be minimized is then expressed as

$$\Psi(\mathbf{f}) = \Omega(\mathbf{f}) + \vartheta \, ||\mathbf{f} - \mathbf{y}||^2 \tag{1}$$

where ϑ is a positive regularizing parameter. The best ranking vector, $\mathbf{f}^* = \arg\min_{\mathbf{f}} \Psi(\mathbf{f})$, is found to be [15, 16]:

$$\mathbf{f}^* = \frac{\vartheta}{1+\vartheta} \left(\mathbf{I} - \frac{1}{1+\vartheta}\mathbf{A}\right)^{-1} \mathbf{y}. \tag{2}$$

Image tagging can be cast as a ranking problem [4].

3. ADAPTIVE HYPEREDGE WEIGHT UPDATING

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 $\mathbf{W}$. In addition to enforcing the inequality constraint $\mathbf{1}_n^T\mathbf{w} = 1$ as in [4], we also require w_i, $i = 1, 2, \ldots, n$, to be non-negative. The latter inequality constraints are collectively referred to as $\mathbf{w} \geq \mathbf{0}$. That is, the following minimization problem is defined:

$$\underset{\mathbf{f},\mathbf{w}}{\text{argmin}} \left\{\Psi(\mathbf{f}) + +\kappa ||\mathbf{w}||^2\right\} \qquad \text{s.t. } \mathbf{1}_n^T\mathbf{w} = 1 \text{ and } \mathbf{w} \geq \mathbf{0} \tag{3}$$

where κ is a positive regularization parameter. The minimization problem (3) is solved in alternating fashion, as is illustrated in Fig. 1. Obviously, when $\mathbf{w}$ is fixed, the solution for $\mathbf{f}$ is given by (2). Therefore, we shall elaborate on the solution for $\mathbf{w}$, when $\mathbf{f}$ is fixed. Let $P(\mathbf{w}) = \mathbf{f}^T\mathbf{L}\mathbf{f} + \kappa||\mathbf{w}||^2$. In this case, the optimization w.r.t. $\mathbf{w}$ is read as:

$$\underset{\mathbf{w}}{\text{argmin}} \; P(\mathbf{w}) \qquad \text{s.t. } \mathbf{1}_n^T\mathbf{w} = 1 \text{ and } \mathbf{w} \geq \mathbf{0}. \tag{4}$$

The new optimization problem (4) is solved by employing gradient descent. By omitting the dependence on $\mathbf{w}$ for notation simplicity, the Lagrangian of the optimization problem is given by

$$Q = P + \sum_{j=1}^{\wp} c_j \, G_j, \tag{5}$$

where c_j, $j = 1, 2, \ldots, \wp$ are the Lagrange multipliers associated to the $\wp$ active constraints G_j defined as

$$G_j : \begin{cases} \mathbf{1}_n^T \, \mathbf{w} - 1 = 0 & \text{for } j = 1 \\ w_{\nu_j - 1} = 0 & \text{for } j > 1. \end{cases} \tag{6}$$

That is, the first constraint G_1 is the equality constraint, which is always active. For the remaining active constraints (i.e., for $j > 1$), we have $2 \leq \nu_j \leq n+1$, such that $\nu_j - 1 \in [1, n]$ is an index of a hyperedge weight. The Kuhn-Tucker theorem [17] requires the Lagrange multipliers to be determined by demanding that ∇Q to be orthogonal to $\nabla G_j = \frac{\partial G_j}{\partial \mathbf{w}}$, i.e.,

$$\nabla G_j^T \, \nabla Q = 0, \quad j = 1, 2, \ldots, \wp \tag{7}$$

It can be shown that:

$$\nabla Q = \nabla P + \sum_{j=1}^{\wp} c_j \, \nabla G_j = \nabla P + \mathbf{\Gamma} \, \mathbf{c} \tag{8}$$

where $\mathbf{c} \in \mathbb{R}^{\wp}$ and $\mathbf{\Gamma}$ is a matrix of size $n \times \wp$ having a special structure. In particular, its first column is $\mathbf{1}_n$, while the remaining columns have 1 at the row $\nu_j - 1$ and zero otherwise. Its j-th column is simply ∇G_j, i.e. $\mathbf{\Gamma} = [\nabla G_1 | \nabla G_2 | \cdots | \nabla G_{\wp}]$. Accordingly, the system of equations (7) can be rewritten as:

$$\mathbf{\Gamma}^T \, \nabla Q = \mathbf{\Gamma}^T \, (\nabla P + \mathbf{\Gamma}\mathbf{c}) = \mathbf{0} \Leftrightarrow \mathbf{c} = -(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\nabla P, \tag{9}$$

yielding a closed-form expression for $\mathbf{c}$ that can be further simplified. Let

$$S_{\text{inactive}} = \sum_{\substack{i=1 \\ w_i \neq 0}}^{n} (\nabla P)_i, \tag{10}$$

where $(\nabla P)_i$ denotes the i-th element of ∇P. By exploiting the structure of $\mathbf{\Gamma}$, it can be found that

$$\mathbf{c} = \begin{bmatrix} \frac{-S_{\text{inactive}}}{n-\wp+1} \\ \frac{S_{\text{inactive}}}{n-\wp+1} - (\nabla P)_{\nu_2 - 1} \\ \vdots \\ \frac{S_{\text{inactive}}}{n-\wp+1} - (\nabla P)_{\nu_{\wp} - 1} \end{bmatrix}. \tag{11}$$

Substituting (11) in (8), we obtain the i-th element of ∇Q, i.e.,

$$(\nabla Q)_i = \begin{cases} 0 & i : w_i = 0 \\ (\nabla P)_i - \frac{S_{\text{inactive}}}{n-\wp+1} & \text{otherwise}, \end{cases} \tag{12}$$

where $(\nabla P)_i$ is given by [4]:

$$(\nabla P)_i = -\mathbf{f}^T \left(D_e^{-1}(i,i)\mathbf{\Lambda}_i\mathbf{\Lambda}_i{}^T - \mathbf{\Xi}_i\right) \mathbf{f} + 2\kappa w_i. \tag{13}$$

In (13), $\mathbf{\Lambda}_i \in \mathbf{R}^{|V|}$ is the i-th column of $\mathbf{\Lambda} = \mathbf{D}_v^{-1/2} \, \mathbf{H}$ and $\mathbf{\Xi}_i = \text{diag}(\mathbf{\Lambda}_i)\mathbf{D}_v^{-1/2}\mathbf{A}$. Observe that $\mathbf{\Xi}_i$ is a symmetric matrix and $\text{diag}(\mathbf{\Lambda}_i)$ is a $|V| \times |V|$ diagonal matrix having $\mathbf{\Lambda}_i$ in its main diagonal.

The gradient descent requires $\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} - \mu\nabla Q$. That is, if w_i^{old} was active (i.e., $w_i^{\text{old}} = 0$), then $w_i^{\text{new}} = 0$. Otherwise,

$$w_i^{\text{new}} = w_i^{\text{old}} - \mu(\nabla P)_i + \mu\frac{S_{\text{inactive}}}{n - \wp + 1}. \tag{14}$$

Algorithm 1 summarizes the adaptive weight estimation.

In (14), an arbitrary fixed small adaptation step μ is used as in the classical gradient descent [17]. In order to achieve a sufficient decrease in the objective function between successive iterations and speed up in this way the convergence of the algorithm, the Armijo rule [18] is employed to properly select the adaptation step μ. The Armijo rule states that for sufficiently decreasing the objective function at iteration k

$$Q(\mathbf{w}(k)) = \mathbf{f}^T\mathbf{L}\mathbf{f} + \kappa||\mathbf{w}(k)||^2 + \sum_{j=1}^{\wp} \mathbf{c}_j \, G_j \tag{15}$$

Algorithm 1 Image tagging via hyperedge weight learning with gradient descent

Inputs: The objects (i.e., users, groups, tags, geo-tags and images) and their relations. Set the regularization parameters ϑ and κ.
Output: Optimized weights $\mathbf{w}$ and the ranking vector $\mathbf{f}$.

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

2 Compute the affinity matrix $\mathbf{A} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T \mathbf{D}_v^{-1/2} \in \mathbb{R}^{|V|\times|V|}$. Set the query vector $\mathbf{y} \in \mathbb{R}^{|V|}$.

3 Find result ranking vector $\mathbf{f} \in \mathbb{R}^{|V|}$, using (2).

4 Compute the gradient $\nabla P(\mathbf{w})$ with elements as in (13).

5 Update the non-zero weights, using (14). If a weight becomes zero, it remains zero for ever.

6 Having found the new hyperedge weights update $\mathbf{A}$, $\mathbf{D}_v$, and $\mathbf{W}$.

7 Repeat the steps 2 - 6 until convergence. Find the final ranking vector $\mathbf{f}$.

the adaptation step can be updated as $\mu_k = \varrho\, \mu_{k-1}$, for $\varrho \in (0, 1]$ until the condition

$$Q(\mathbf{w}(k) + \mu_k\, \mathbf{d}_k) \leq Q(\mathbf{w}(k)) + \eta_1\, \mu_k \nabla Q^T(\mathbf{w}(k))\, \mathbf{d}(k) \quad (16)$$

is fulfilled for some $\eta_1 \in (0, 1)$ (e.g., $\eta_1 = 10^{-4}$) with $\mathbf{d}(k) = -\nabla Q(\mathbf{w}(k))$ being the search direction in the steepest descent method [18].

4. DATASET DESCRIPTION AND HYPERGRAPH CONSTRUCTION

The image dataset used in [4] is exploited here as well. The dataset was collected from *Flickr*. It contains both indoor and outdoor medium sized photos of popular Greek landmarks, including city scenes and landscapes. Using *Flickr API*[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 [19].

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

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

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. 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*[6] 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 [20] and they were reduced to the 5 most similar images to the reference image, by using scale-invariant feature transform (SIFT) [21]. 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.

5. EXPERIMENTS

The averaged Recall-Precision and the 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

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

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

[6] A hypergraph is needed, indeed.

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

is the weighted harmonic mean of precision and recall, which measures the effectiveness of tagging when treating precision and recall as equally important, i.e., $F_1 = 2\frac{Precision \cdot Recall}{Precision+Recall}$. The F_1 measure is also measured at several ranking positions.

Let us refer to the ranking obtained by the proposed adaptive weight estimation method with steepest descent as ITH-HWEG when a fixed adaptation step is used and ITH-HWEA when the Armijo rule is employed. The method proposed in [4] is referred to as ITH-HWE and the ranking obtained by (2) is denoted as ITH. In the aforementioned acronyms, ITH stands for Imaging Tagging on Hypergraph, HWE reads as Hypergraph Weight Estimation, and the final G and A signals whether a fixed step size or the Armijo rule has been used in the proposed steepest descent algorithm.

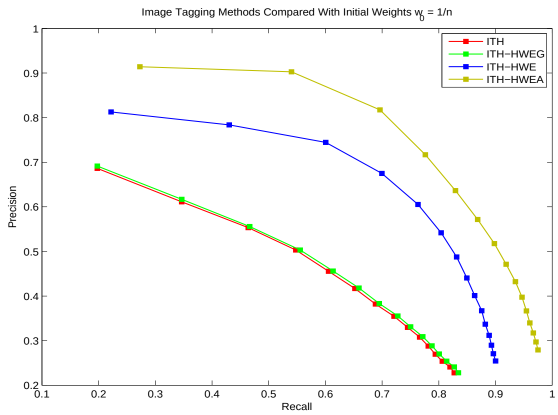

Fig. 2. Averaged Recall-Precision curves for all methods with initial hyperedge weights $\frac{1}{n}$.

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 when the initial hypergraph weights are initialized as $\frac{1}{n}$. 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. It is seen that the proposed method ITH-HWEA outperforms all the methods it is compared to, validating its effectiveness for hyperedge weight learning. The ITH-HWEG that employs a fixed adaptation does not yield a performance improvement. The F1 measure at various ranking positions is summarized in Table 3.

Experiments were also conducted with random initial hyperedge weights. The averaged Recall-Precision curves are plotted in Fig. 3. It is seen that both methods ITH-HWEG and ITH-HWEA outperform the baseline techniques ITH-HWE [4] and ITH [16].

Table 3. F_1 measure at various ranking positions for the compared methods when the hyperedge weights are initialized as $\frac{1}{n}$.

Initial weight set to $\mathbf{w}(0) = \frac{1}{n}\mathbf{1}_n$	F_1@1	F_1@2	F_1@5	F_1@10
ITH [16]	0.307	0.444	0.520	0.440
ITH-HWE[4]	0.349	0.556	0.675	0.517
ITH-HWEG	0.317	0.458	0.541	0.445
ITH-HWEA	**0.420**	**0.676**	**0.720**	**0.560**

Table 4. F_1 measure for ITH-HWEG and ITH-HWEA.

Random initial weights	F_1@1	F_1@2	F_1@5	F_1@10
ITH-HWEG	0.425	0.682	0.753	0.558
ITH-HWEA	**0.431**	**0.695**	**0.760**	**0.560**

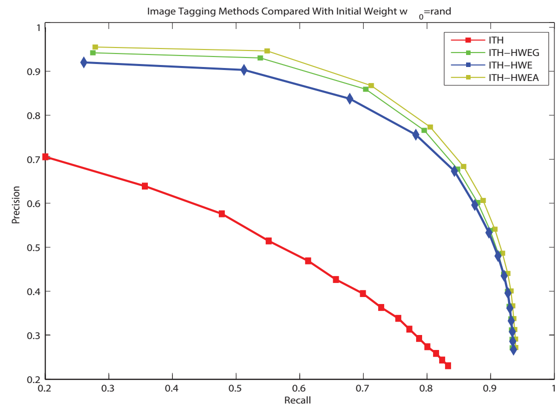

Fig. 3. Averaged Recall-Precision curves for all methods when the initial hyperedge weights are randomly initialized.

The F1 measure at ranking positions 1, 2, 5, and 10 for ITH-HWEG and ITH-HWEA is listed in Table 4.

The experimental findings indicate that the initialization of the hyperedge weights affects the image tagging efficiency. The use of random hyperedge weight initialization combined with the Armijo rule for step size selection is highly recommended, as it guarantees the best performance.

6. CONCLUSIONS AND FUTURE WORK

In this paper, efficient adaptive hyperedge weight learning algorithms have been proposed for image tagging. The experiments conducted on a collection of images related to Greek sites have demonstrated the superiority of the proposed algorithms. The incremental update of an already trained hypergraph learning model could be a topic of future research.

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.

7. REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] K. Pliakos and C. Kotropoulos, "Weight estimation in hypergraph learning," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2015, pp. 1161–1165.

[5] C. Berge, *Hypergraphs*, North Holland, Amsterdam, The Netherlands, 1989.

[6] C. Kotropoulos, "Multimedia social search based on hypergraph learning," in *Graph-Based Social Media Analysis*, I. Pitas, Ed., pp. 215–273. CRC Press, Boca Raton, FL, 2016.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Advances in Neural Information Processing Systems*, 2007, vol. 19, pp. 1601–1608.

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

[15] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, 2004, vol. 16, pp. 321–328.

[16] 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.

[17] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[18] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2006.

[19] 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.

[20] 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.

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