Adaptive algorithms for hypergraph learning

Aikaterini Chassapi, Constantine Kotropoulos, and Konstantinos Pliakos

Department of Informatics, Aristotle University of Thessaloniki, GREECE

chasapia@csd.auth.gr,{costas,kpliakos}@aiia.csd.auth.gr

March 23rd, 2016

Outline

1. Introduction

Outline

1. Introduction
2. Unified hypergraph framework

Outline

1. Introduction
2. Unified hypergraph framework
3. Hyperedge weight estimation scheme

Outline

1. Introduction
2. Unified hypergraph framework
3. Hyperedge weight estimation scheme
4. Adaptive hyperedge weight estimation

Outline

1. Introduction
2. Unified hypergraph framework
3. Hyperedge weight estimation scheme
4. Adaptive hyperedge weight estimation
5. Experimental results

Outline

Motivation

- The amount of images uploaded nowadays to the web is overwhelmingly huge and thus, the need for efficient tagging methods has risen.

Introduction

Motivation

- The amount of images uploaded nowadays to the web is overwhelmingly huge and thus, the need for efficient tagging methods has risen.
- 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.

Introduction

Motivation

- The amount of images uploaded nowadays to the web is overwhelmingly huge and thus, the need for efficient tagging methods has risen.
- 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 in terms of high-order relations.

Introduction

Motivation

- The amount of images uploaded nowadays to the web is overwhelmingly huge and thus, the need for efficient tagging methods has risen.
- 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 in terms of high-order relations.
- Hypergraphs can model such high-order relations between their vertices by hyperedges whose influence can be assessed by properly estimating their weights.

Introduction

What is the paper about?

- An efficient adaptive hypergraph weight estimation scheme is proposed for image tagging by
  1. enforcing both equality and inequality constraints during hypergraph learning

Introduction

What is the paper about?

- An efficient adaptive hypergraph weight estimation scheme is proposed for image tagging by
    1. enforcing both equality and inequality constraints during hypergraph learning
    2. and employing an efficient adaptation step using the Armijo rule.

[a] `http://www.flickr.com`

What is the paper about?

- An efficient adaptive hypergraph weight estimation scheme is proposed for image tagging by
  1. enforcing both equality and inequality constraints during hypergraph learning
  2. and employing an efficient adaptation step using the Armijo rule.
- Experiments conducted on a dataset crawled from *Flickr*[a] demonstrate the superior performance of the proposed approach compared to the state-of-the-art.

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

Unified Hypergraph Framework

Hypergraphs consist of a set of:

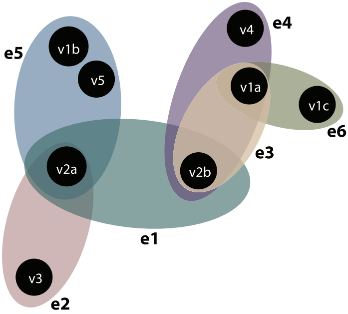
- vertices (v) made by concatenating different kind of objects.
- hyperedges (e), linking these vertices.

Unified Hypergraph Framework

Hypergraphs consist of a set of:

- vertices (v) made by concatenating different kind of objects.
- hyperedges (e), linking these vertices.

v1a, v1b, v1c : images v2a, v2b : users v3 : groups v4 : geo-tags v5 : tags	e1: friendship relations e2: social group relations e3: possession relations e4: geo-tag relations e5: tag relations e6: image visual relations



Unified Hypergraph Framework

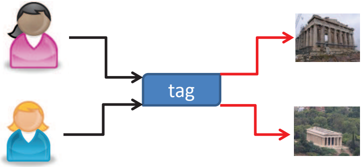
Why not simple graphs?

- Simple graphs model only pairwise relations between the items.
- Hyperedges capture high-order relations. Thus, triplet relations among different types of objects (i.e., users-images-tags or users-images-geotags) can be represented.

Unified Hypergraph Framework

Why not simple graphs?

- Simple graphs model only pairwise relations between the items.
- Hyperedges capture high-order relations. Thus, triplet relations among different types of objects (i.e., users-images-tags or users-images-geotags) can be represented.



Simple graph cannot answer who has tagged what.

Unified Hypergraph Framework

Construction of the Unified Hypergraph

- Different relations correspond to different types of hyperedges.
- 6 types of hyperedges ($E^{(1)}, E^{(2)}, E^{(3)}, E^{(4)}, E^{(5)}, E^{(6)}$), linking 5 types of objects (images, users, user groups, geo-tags, and tags).

Table : The structure of the hypergraph incidence matrix **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

Unified Hypergraph Framework

$E^{(1)}$	$E^{(2)}$	$E^{(3)}$	$E^{(4)}$	$E^{(5)}$	$E^{(6)}$
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

Different types of hyperedges:

- $E^{(1)}$: represents a pairwise friendship relation between users.
- $E^{(2)}$: represents a group of users.
- $E^{(3)}$: represents a user-image possession relation.
- $E^{(4)}$: contains an image, its owner, and its geo-location.
- $E^{(5)}$: contains a triplet of an image, a user, and a tag, representing a tagging relation.
- $E^{(6)}$: contains pairs of vertices, which represent two visually similar images.

Notations

- $G(V, E, w)$ a hypergraph of a set of vertices V and hyperedges E, with a weight function $w : E \rightarrow \mathbb{R}$ assigned.
- $\mathbf{D}_u = \mathrm{diag}(\mathbf{Hw})$ the vertex degree matrix, $\mathbf{D}_e = \mathrm{diag}(\mathbf{1}^T\mathbf{H})$ the hyperedge degree matrix, the weight matrix $\mathbf{W} = \mathrm{diag}(\mathbf{w})$ containing the hyperedge weights w_i, $\|.\|_2$ the ℓ_2 norm of a vector and $\mathbf{I}$ the identity matrix.
- $\mathbf{y}$ and $\mathbf{f} \in \mathbb{R}^{|V|}$ the query and ranking vectors, respectively.
- ϑ a regularizing parameter.
- $\mathbf{L} = \mathbf{I} - \mathbf{A}$ the positive semi-definite hypergraph Laplacian matrix, where $\mathbf{A} = \mathbf{D}_u^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_u^{-1/2}$.

Hyperedge weight estimation scheme

Last year ICASSP optimization problem

- In a hypergraph clustering problem, one seeks to minimize $\Omega(\mathbf{f}) = \mathbf{f}^T\mathbf{L}\mathbf{f}$ requiring all vertices with the same value in $\mathbf{f}$ to be strongly connected[a].
- The ℓ_2 regularization norm between the ranking vector $\mathbf{f}$ and the query vector $\mathbf{y}$ could be included to address a ranking problem[b].
- Hypergraph ranking was enhanced by optimizing the hyperedge weights $\mathbf{w} = (w_1, w_2, \cdots, w_n)^T$ so that[c]:

$$\underset{\mathbf{f},\mathbf{w}}{\operatorname{argmin}} \left\{\Omega(\mathbf{f}) + \vartheta\,||\mathbf{f} - \mathbf{y}||^2 + \kappa||\mathbf{w}||^2\right\} \qquad \text{s.t. } \mathbf{1}_n^T\mathbf{w} = 1. \tag{1}$$

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

[b] 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.

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

Adaptive hyperedge weight estimation

ICASSP 2016 optimization problem

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

where $\Psi(\mathbf{f}) = \mathbf{f}^T \mathbf{L} \mathbf{f} + \vartheta ||\mathbf{f} - \mathbf{y}||^2$ and κ is a positive regularization parameter.

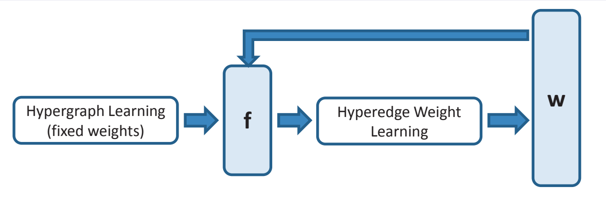
Adaptive hyperedge weight estimation

ICASSP 2016 optimization problem

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

where $\Psi(\mathbf{f}) = \mathbf{f}^T \mathbf{L} \mathbf{f} + \vartheta ||\mathbf{f} - \mathbf{y}||^2$ and κ is a positive regularization parameter.

Alternating optimization



Adaptive hyperedge weight estimation

Optimization wrt. **f** when **w** is fixed

Closed-form solution[a]:

$$\mathbf{f}^* = \frac{\vartheta}{1+\vartheta}\left(\mathbf{I} - \frac{1}{1+\vartheta}\mathbf{A}\right)^{-1}\mathbf{y}. \tag{3}$$

[a] 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.

Adaptive hyperedge weight estimation

Optimization wrt. f when w is fixed

Closed-form solution[a]:

$$\mathbf{f}^* = \frac{\vartheta}{1+\vartheta}\left(\mathbf{I} - \frac{1}{1+\vartheta}\mathbf{A}\right)^{-1}\mathbf{y}. \quad (3)$$

[a] 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.

Optimization wrt. w when f is fixed

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

where $P(\mathbf{w}) = \mathbf{f}^T\mathbf{L}\mathbf{f} + \kappa||\mathbf{w}||^2$.

Adaptive hyperedge weight estimation

- The new optimization problem (4) is solved by employing gradient descent.

Adaptive hyperedge weight estimation

- The new optimization problem (4) is solved by employing gradient descent.
- Lagrangian of the optimization problem:

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

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

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

Adaptive hyperedge weight estimation

- The new optimization problem (4) is solved by employing gradient descent.
- Lagrangian of the optimization problem:

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

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

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

- with G_1 being an equality constraint and for $j > 1$: $2 \leq \nu_j \leq n + 1$, such that $\nu_j - 1 \in [1, n]$ being an index of a hyperedge weight.

Adaptive hyperedge weight estimation

- The Kuhn-Tucker theorem 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.,

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

Adaptive hyperedge weight estimation

- The Kuhn-Tucker theorem 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.

Adaptive hyperedge weight estimation

Structure of $\mathbf{\Gamma} = [\nabla G_1 | \nabla G_2 | \cdots | \nabla G_{\wp}]$

- its first column is $\mathbf{1}_n$;

Adaptive hyperedge weight estimation

Structure of $\mathbf{\Gamma} = [\nabla G_1 | \nabla G_2 | \cdots | \nabla G_\wp]$

- its first column is $\mathbf{1}_n$;
- The remaining columns have 1 at the row $\nu_j - 1$ and zero otherwise.

Solution of (7) wrt. $\mathbf{c}$

Adaptive hyperedge weight estimation

Structure of $\mathbf{\Gamma} = [\nabla G_1 | \nabla G_2 | \cdots | \nabla G_{\wp}]$

- its first column is $\mathbf{1}_n$;
- The remaining columns have 1 at the row $\nu_j - 1$ and zero otherwise.

Solution of (7) wrt. $\mathbf{c}$

- $\mathbf{c} = -(\mathbf{\Gamma}^T \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^T \nabla P$.

Adaptive hyperedge weight estimation

Structure of $\mathbf{\Gamma} = [\nabla G_1 | \nabla G_2 | \cdots | \nabla G_{\wp}]$

- its first column is $\mathbf{1}_n$;
- The remaining columns have 1 at the row $\nu_j - 1$ and zero otherwise.

Solution of (7) wrt. $\mathbf{c}$

- $\mathbf{c} = -(\mathbf{\Gamma}^T \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^T \nabla P$.
- Let $S_{\text{inactive}} = \sum_{\substack{i=1 \\ w_i \neq 0}}^{n} (\nabla P)_i$, where $(\nabla P)_i$ denotes the i-th element of ∇P.

Adaptive hyperedge weight estimation

Structure of $\mathbf{\Gamma} = [\nabla G_1 | \nabla G_2 | \cdots | \nabla G_{\wp}]$

- its first column is $\mathbf{1}_n$;
- The remaining columns have 1 at the row $\nu_j - 1$ and zero otherwise.

Solution of (7) wrt. $\mathbf{c}$

- $\mathbf{c} = -(\mathbf{\Gamma}^T \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^T \nabla P$.
- Let $S_{\text{inactive}} = \sum_{\substack{i=1 \\ w_i \neq 0}}^{n} (\nabla P)_i$, where $(\nabla P)_i$ denotes the i-th element of ∇P.
-
$$\mathbf{c} = \begin{bmatrix} \frac{-S_{\text{inactive}}}{n-\wp+1} \\ \frac{S_{\text{inactive}}}{n-\wp+1} - (\nabla P)_{\nu_2 - 1} \\ \vdots \\ \frac{S_{\text{inactive}}}{n-\wp+1} - (\nabla P)_{\nu_{\wp} - 1} \end{bmatrix}. \tag{9}$$

Adaptive hyperdge weight updating

i-th element of ∇Q

$$(\nabla Q)_i = \begin{cases} 0 & i : w_i = 0 \\ (\nabla P)_i - \frac{S_{\text{inactive}}}{n - \wp + 1} & \text{otherwise}, \end{cases} \tag{10}$$

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

$$(\nabla P)_i = -\mathbf{f}^T \left(D_e^{-1}(i,i) \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^T - \boldsymbol{\Xi}_i \right) \mathbf{f} + 2\kappa w_i \tag{11}$$

with

- $\boldsymbol{\Lambda}_i \in \mathbf{R}^{|V|}$ being the i-th column of $\boldsymbol{\Lambda} = \mathbf{D}_v^{-1/2} \mathbf{H}$;

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

Adaptive hyperdge weight updating

i-th element of ∇Q

$$(\nabla Q)_i = \begin{cases} 0 & i : w_i = 0 \\ (\nabla P)_i - \frac{S_{\text{inactive}}}{n - \wp + 1} & \text{otherwise}, \end{cases} \tag{10}$$

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

$$(\nabla P)_i = -\mathbf{f}^T \left(D_e^{-1}(i,i) \mathbf{\Lambda}_i \mathbf{\Lambda}_i^T - \mathbf{\Xi}_i \right) \mathbf{f} + 2\kappa w_i \tag{11}$$

with

- $\mathbf{\Lambda}_i \in \mathbf{R}^{|V|}$ being the i-th column of $\mathbf{\Lambda} = \mathbf{D}_v^{-1/2} \mathbf{H}$;
- $\mathbf{\Xi}_i = \text{diag}(\mathbf{\Lambda}_i) \mathbf{D}_v^{-1/2} \mathbf{A}$ being a a $|V| \times |V|$ symmetric matrix;

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

Adaptive hyperdge weight updating

i-th element of ∇Q

$$(\nabla Q)_i = \begin{cases} 0 & i : w_i = 0 \\ (\nabla P)_i - \frac{S_{\text{inactive}}}{n-\wp+1} & \text{otherwise,} \end{cases} \qquad (10)$$

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

$$(\nabla P)_i = -\mathbf{f}^T \left(D_e^{-1}(i,i)\mathbf{\Lambda}_i \mathbf{\Lambda}_i^T - \mathbf{\Xi}_i \right) \mathbf{f} + 2\kappa w_i \qquad (11)$$

with

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

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

Adaptive hyperdge weight updating

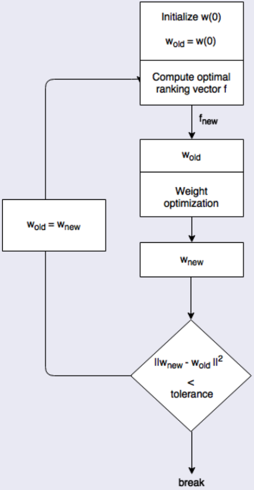
Gradient descent

$\mathbf{w}^{\mathrm{new}} = \mathbf{w}^{\mathrm{old}} - \mu \nabla Q$. That is,

$$w_i^{\mathrm{new}} = \begin{cases} 0 & \text{if } w_i^{\mathrm{old}} = 0 \\ w_i^{\mathrm{old}} - \mu (\nabla P)_i + \mu \frac{S_{\mathrm{inactive}}}{n - \wp + 1} & \text{otherwise.} \end{cases}$$

Adaptive hyperdge weight updating

Flowchart



Adaptive hyperdge weight updating

Armijo Rule

- An arbitrary fixed small adaptation step μ can be used as in the classical gradient descent[a].

[a] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.
[b] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2006.

Adaptive hyperdge weight updating

Armijo Rule

- An arbitrary fixed small adaptation step μ can be used as in the classical gradient descent[a].
- To achieve a sufficient decrease, the Armijo rule is employed to properly select the adaptation step μ.

[a] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.
[b] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 2006.

Adaptive hyperdge weight updating

Armijo Rule

- An arbitrary fixed small adaptation step μ can be used as in the classical gradient descent[a].
- To achieve a sufficient decrease, the Armijo rule is employed to properly select the adaptation step μ.
- At iteration k: $Q(\mathbf{w}(k)) = \mathbf{f}^T \mathbf{L} \mathbf{f} + \kappa ||\mathbf{w}(k)||^2 + \sum_{j=1}^{\wp} c_j \, G_j$

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

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

Adaptive hyperdge weight updating

Armijo Rule

- An arbitrary fixed small adaptation step μ can be used as in the classical gradient descent[a].
- To achieve a sufficient decrease, the Armijo rule is employed to properly select the adaptation step μ.
- At iteration k: $Q(\mathbf{w}(k)) = \mathbf{f}^T \mathbf{L}\mathbf{f} + \kappa ||\mathbf{w}(k)||^2 + \sum_{j=1}^{\wp} c_j \, G_j$
- Adaptation step: $\mu_k = \varrho \, \mu_{k-1}$, for $\varrho \in (0, 1]$ until the condition

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

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[b].

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

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

Experimental Results

Table : 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

Evaluation metrics

- 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, i.e.,

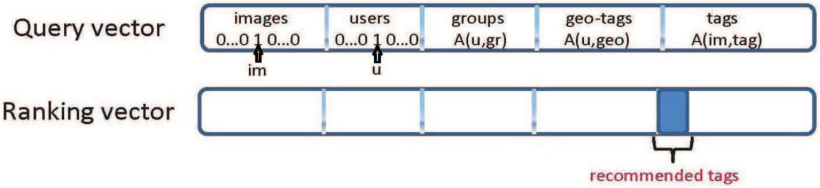
$$F_1 = 2\frac{\mathit{Precision} \cdot \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}}.$$

Experimental results

Protocol

- Test set containing the 25% of the tags
- Training set containing the remaining 75%.
- 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.

Figure : The structure of query and result ranking vectors.



Experimental results

Algorithms compared

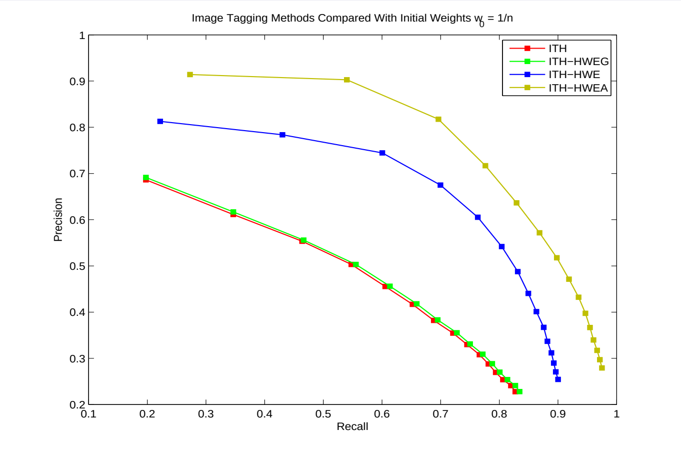
- ITH: Image Tagging on Hypergraph[a]
- ITH-HWE: Image tagging on Hypergraph with Hyperedge Weight Estimation (ICASSP 2015 algorithm)[b]
- ITH-HWEG: Adaptive hypergraph weight estimation with gradient descent and fixed adaptation step (ICASSP 2016 1st proposal)
- ITH-HWEA Adaptive hypergraph weight estimation with gradient descent and Armijo rule (ICASSP 2016 2nd proposal)

[a] 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.

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

Experimental results

Averaged recall-precision curves with initial hyperedge weights $\frac{1}{n}$

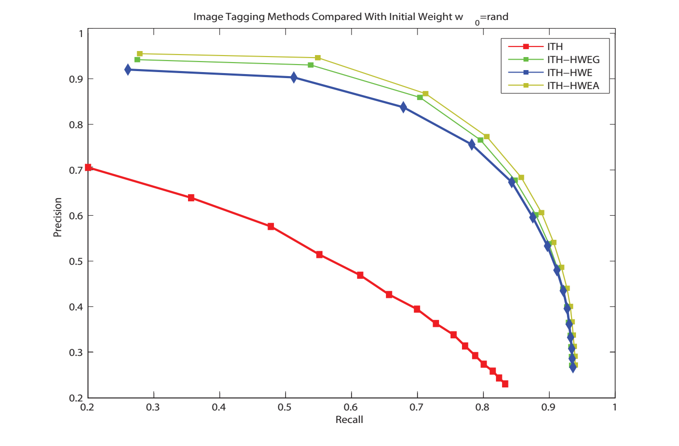


F_1 measure at various ranking positions when the hyperedge weights are initialized as $\frac{1}{n}$

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	0.307	0.444	0.520	0.440
ITH-HWE	0.349	0.556	0.675	0.517
ITH-WHEG	0.317	0.458	0.541	0.445
ITH-HWEA	**0.420**	**0.676**	**0.720**	**0.560**

Experimental results

Averaged recall-precision curves when the initial hyperedge weights are randomly initialized.



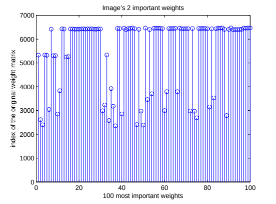
Experimental results

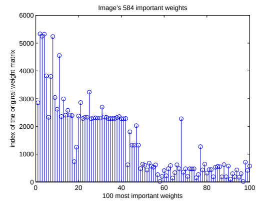
F_1 measure for ITH-HWEG and ITH-HWEA curves when the initial hyperedge weights are randomly initialized.

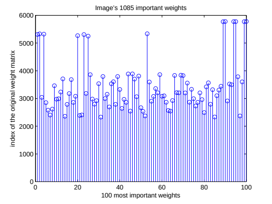
Random initial weights	F_1@1	F_1@2	F_1@5	F_1@10
ITH-WHEG	0.425	0.682	0.753	0.558
ITH-HWEA	**0.431**	**0.695**	**0.760**	**0.560**

Experimental results

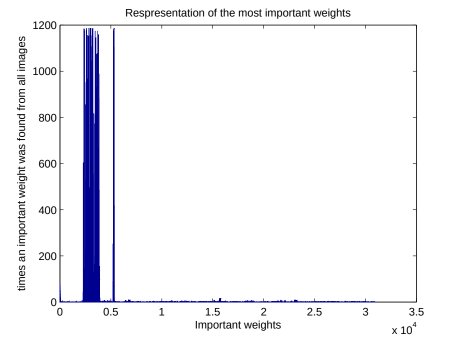
100 important weights for 3 images

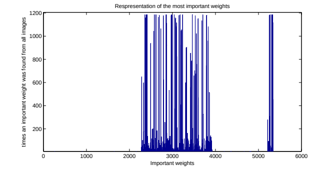


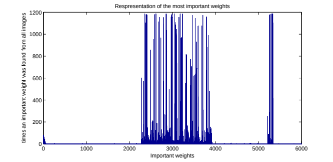




Histograms of hyperedge activations







Conclusions

- All the hyperedges do not hold the same importance as others. Some of these hyperdges are more informative and thus should have higher impact on the learning procedure.

Conclusions

- All the hyperedges do not hold the same importance as others. Some of these hyperdges are more informative and thus should have higher impact on the learning procedure.
- Both the proposed methods ITH-HWEG and ITH-HWEA outperform the baseline techniques.

Conclusions

- All the hyperedges do not hold the same importance as others. Some of these hyperdges are more informative and thus should have higher impact on the learning procedure.
- Both the proposed methods ITH-HWEG and ITH-HWEA outperform the baseline techniques.
- The most effective tagging method employs randomly initialized hyperedge weights and the Armijo rule for determining the adaptation step.

Conclusions

- All the hyperedges do not hold the same importance as others. Some of these hyperdges are more informative and thus should have higher impact on the learning procedure.
- Both the proposed methods ITH-HWEG and ITH-HWEA outperform the baseline techniques.
- The most effective tagging method employs randomly initialized hyperedge weights and the Armijo rule for determining the adaptation step.
- The aforementioned choices reduce significantly the time needed for algorithm convergence.

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: 11SYN-10-1730-ATLAS.

Ministry of Education and Religious Affairs
GSRT - Management and Implementation Agency for RTD and Innovation Activities

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: 11SYN-10-1730-ATLAS.

Ministry of Education and Religious Affairs
GSRT - Management and Implementation Agency for RTD and Innovation Activities

Questions?

Thank You! Any Questions?