Master Thesis

TAGGING AND RECOMMENDATION ALGORITHMS ON HYPERGRAPHS

Pliakos Konstantinos

Supervisor: Prof. Kotropoulos Constantine

ARISTOTLE UNIVERSITY OF THESSALONIKI

DEPARTMENT OF INFORMATICS

ACADEMIC YEAR 2015-2016

Contents

List of Figures

List of Tables

Abstract

Nowadays, advanced web technologies and 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 and recommendation. Despite the many efforts undertaken so far, accuracy or efficiency still remain open problems.

The main goal of this master thesis is to develop efficient tagging and recommendation methods. These methods are based on hypergraph learning and they are further enhanced using social media information and probabilistic models, such as the Probabilistic Latent Semantic Analysis (PLSA). Moreover, the hypergraph learning is advanced by applying a hyperedge weight tuning scheme. In this way, the information captured by the hyperedges is efficiently distilled.

Acknowledgments

I would like to thank my supervisor, Professor Constantine Kotropoulos for his continuous advice and help during this master thesis. I would also want to thank my parents for their support.

Chapter 1

Subject of Study

1.1 Objectives

Nowadays, the continuously rising popularity of photo-sharing web applications has led to a huge amount of uploaded images. This boundless volume of image data has triggered the interest in image tagging and recommendation. The image tagging is a very critical procedure, as it is responsible for search engine retrieval accuracy and contributes to the organization of the images uploaded to the web. It aims at bridging the gap between the semantic and visual content of an image. However, it suffers from several limitations, such as spam, lack of uniformity, and noise. Therefore, an effective tagging system is of paramount importance. Recently, besides tagging, much progress has been made toward developing new recommender systems. Recommender systems typically produce a list of recommendations through collaborative or content-based filtering. They are a useful alternative to search algorithms since they provide users with items they might not have found by themselves. However, achieving satisfactory efficiency or accuracy remains still an open problem.

The main goal of this master thesis is to develop methods in order to address effectively image tagging and recommendation. The proposed methods are based on hypergraph learning. The hypergraph theory is among the research areas of study that have both considerable theoretical interest and an important range of practical applications. Text information distilled from image tags, social media information, and image visual content are exploited, modeling relations between the images in a unified hypergraph framework. Moreover, the hypergraph learning is enhanced by applying an updating hyperedge weight tuning scheme.

This way, the information is distilled as each hyperedge of the unified hypergraph is treated in a different manner. The proposed methods are based on thorough mathematical analyses which are presented in this master thesis in full detail. Experiments were conducted and satisfactory results were presented, demonstrating the efficiency of the proposed methods.

1.2 Outline

The outline of the master thesis is as follows:

- In Chapter 2, the main concept behind the image tagging and the image recommendation is detailed.

- In Chapter 3, the motivation behind the use of hypergraphs is highlighted and the general hypergraph learning model is presented.

- In Chapter 4, an efficient model for image annotation and tourism places of interest (POI) recommendation is detailed. Tourism related information has been distilled from a large collection of images using clustering and classification methods. The model consists of two main parts. First, a PLSA-driven image annotation method is proposed where tourism related images are annotated geographically, semantically, and visually, harnessing visual attributes and text information. Second, a tourism recommendation approach is proposed based on hypergraph ranking and enforcing group sparsity constraints.

- In Chapter 5, image tagging has been addressed within a unified hypergraph learning framework, exploiting image visual similarities, metadata information, and social media information. The approach is further enhanced by employing a hyperedge weight updating scheme. The developed framework can also be applied to other problems, such as retrieval, or recommendation.

- In Chapter 6, the hyperedge weight updating method is revisited and further analyzed. A novel iterative optimization scheme is proposed. In particular, both equality and inequality constraints enforced during hypergraph learning are taken into account. A thorough mathematical analysis is provided and experimental results are disclosed.

- In Chapter 7, conclusions are drawn, establishing a broad perspective of the whole work done in this master thesis. Hypergraph learning is an interesting research field posing difficult research challenges. Some of these challenges are described in this chapter and several topics of future research are proposed.

Chapter 2

Preliminaries

2.1 Image Tagging

A tag is a keyword or term assigned to an item, such as a digital image. This kind of metadata helps us to describe an item and leverages browsing or searching in large collections of data. Given a particular object or resource, tagging is a process where a user assigns a tag to an object. Tags are generally chosen informally and personally by the item's creator or by its viewer, depending on the system.

Tagging has gained wide popularity due to the growth of social networking, photographing, sharing, and bookmarking sites [1]. These sites allow users to create and manage tags that categorize content using simple keywords. The use of keywords as part of an identification and classification system long predates computers. In the early days of the web, keyword meta tags were used by web page designers to tell search engines what the web page was about. Nowadays modern tagging systems exploit the user-provided tags by linking several items having the same tag. This way, the browsing and content description is facilitated. Social tagging became popular with the launch of sites like `Delicious`[1] and `Flickr`[2]. In 2003, the social bookmarking website Delicious provided a way for its users to add tags to their bookmarks. Delicious also provided browse-able aggregated views of the bookmarks of all users featuring a particular tag. Flickr allowed its users to add their own text tags to each of their pictures, constructing flexible and easy metadata that made the pictures easy to be searched and found. The success of Flickr and the influence of Delicious

[1] https://delicious.com/

[2] https://www.flickr.com/

popularized the concept, and other social software websites, such as YouTube[3] and Last.fm[4] also implemented tagging.

Nowadays, more and more images are uploaded on the web and the task of finding an image in such huge image collections is a real challenge for an ordinary user. Browsing through this vast volume of data resorts primarily to search engines and image annotation is a crucial procedure, as it affects considerably the retrieval accuracy of search engines. Large amount of researches on image annotation were carried out in the past. Most of them were focused on the image visual content. However, there is a semantic gap between content-based image retrieval and image semantics understandable by humans. Image tagging aims at bridging the semantic gap between the semantic and the visual content of an image. During the recent years, social media information has been used extensively, apart from visual content or text information provided by tags. In this thesis, image tagging is addressed as a hypergraph ranking problem, exploiting social media information, metadata, and image visual content in a unified hypergraph framework.

2.2 Recommender Systems

Recommender systems have changed the way people find products, information, and even other people. They study patterns of behavior to know what someone will prefer from among a collection of items he has never discovered [2]. During the last years new technology behind recommender systems has been developed leading to a rich collection of tools. These new tools and methods offer to the researcher the possibility of developing effective recommender systems. Modern recommender systems increase the interaction between the user and the machine rather than providing a static experience in which users search for and select items [3]. Recommender systems identify recommendations autonomously for each user based both on the preferences of the specific user and the behavior of other users. In the research field of recommendation there are mainly three basic approaches, collaborative filtering, content-based filtering, and hybrid. Collaborative filtering methods are based on the retrieval and analysis of the behavior of several users. A huge amount of data from various users' past activities and preferences are collected and the final recommendation to the user is based on

[3]https://www.youtube.com/

[4]http://www.last.fm/

the similarity between this specific user and other users. Content-based filtering methods use a variety of properties and characteristics of an item and they are based on retrieval methods. The recommendation items to the user are items having similar properties to a former user selection. Hybrid recommender systems are based on the combination of the two aforementioned approaches. In this thesis, the recommendation problem is based on hypergraph learning, exploiting social media information, metadata, and image visual content in a unified hypergraph framework.

Chapter 3

Hypergraph Ranking

3.1 Introduction

Hypergraphs are a generalization of graphs having edges, called hyperedges, that connect more than two vertices. This way, hypergraphs can capture high-order relations (i.e., triple or higher). There are various kinds of hypergraphs. Hypergraphs that include clutters, having no hyperedges containing another hyperedge (i.e., no hyperedge appears as a subset of another hyperedge). Other hypergraphs have abstract complexes, containing all subsets of every hyperedge. Hypergraphs are also separated into uniform and non-uniform, and into directed and undirected.

Hypergraphs have a solid mathematical background [4] and they have been used in a variety of social, biological, and technological fields, including data mining [5], very large scale integration circuits (VLSI) [6], and biomedicine [7]. Hypergraph clustering can be cast into a non-cooperative multi-player game, where the notion of a cluster is equivalent to a classical game-theoretic equilibrium concept [8]. Furthermore, spectral clustering on graphs is extended to hypergraph spectral clustering by optimizing a real-valued relaxation of the hypergraph normalized cut criterion. In computational geometry, a hypergraph may sometimes be called a *range space* and the hyperedges are called ranges [9]. In cooperative game theory, hypergraphs are called *simple games* (voting games), a notion applied to solve problems in social choice theory. Hyperedges are often called *hyperlinks* or *connectors*.

Hypergraph ranking on multimedia is the heart of a general class of problems, such as retrieval, recommendation, and annotation. Ranking methods based only on multimedia

content fail to yield satisfactory results due to the well known problem of the semantic gap. By using hypergraphs one can construct high-order relations between multimedia content, such as the visual information contained in images and videos or the audio information contained in music tracks, and information distilled from the web, such as social media information and multimedia metadata. Nowadays, social media sharing platforms let multimedia convey additional metadata information, including tags and geo-tags. Moreover, by using social media sharing sites, users can upload data and make friendship relations with other users forming this way social networks. To this end, rich, complementary, and worth exploiting information is provided. Problems such as annotation or recommendation can be addressed as ranking on hypergraphs in a unified manner by jointly analyzing the content and its associated context defined by the metadata and the social relations between the users. In the research field of multimedia recommendation, hypergraphs were employed modeling the various objects (i.e., users, user groups, tags, images, videos, music tracks) and the relations among them, such as friendship relations, membership relations, and tagging relations [8]. Then, recommendation can be treated as an optimization problem, containing a cost function that guarantees that vertices contained in many common hyperedges have similar scores in addition to an ℓ_2 norm between the resulting ranking vector and the initial query vector. A query vector is initially set and the result of the ranking procedure is the final ranking vector containing the scores of the vertices. The vertices in the hypergraph are ranked according to their relevance to the query. By altering the query vector, one can also address other problems such as tagging or retrieval.

3.2 General Hypergraph Model

Hereafter, set cardinality is denoted by $|\cdot|$, the ℓ_2 norm of a vector appears as $\|.\|$ and $\mathbf{I}$ is the identity matrix of compatible dimensions. Let $G(V, E, w)$ denotes a hypergraph with set of vertices $V = \{v_1, v_2, \cdots, v_{|V|}\}$ and set of hyperedges $E = \{e_1, e_2, \cdots, e_{|E|}\}$ to which a weight function $w : E \to \mathbb{R}$ is assigned. V consists of sets of objects of different type and E consists of hyper-relations between these objects. As it can be conducted for a standard volume of vertices an enormously large volume of hyperedges can be generated, forming a very sparse matrix having size $|V| \times |E|$, called incidence matrix $\mathbf{H}$. $\mathbf{H}$ is formed having as

elements:

$$H(v,e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

Based on $\mathbf{H}$, the vertex and hyperedge degrees are defined as:

$$\delta(v) = \sum_{e \in E} w(e) H(v,e) \tag{3.2}$$

$$\delta(e) = \sum_{v \in V} H(v,e). \tag{3.3}$$

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. The rank of a hypergraph is defined as $\max_{e \in E} \delta(e)$, while the anti-rank is defined as $\min_{e \in E} \delta(e)$. If the rank of a hypergraph equals its anti-rank, then the hypergraph is called *uniform*. The hypergraph is called κ-*uniform* if the degree of each hyperedge is constant i.e., κ. To this end, a 2-uniform hypergraph is a graph, a 3-uniform hypergraph is a collection of unordered triples, and so on. Clearly, in undirected κ-uniform hypergraphs, the ordering of vertices in a hyperedge does not matter. Hypergraph learning techniques are separated into two main classes. The first class uses hypermatrices for clustering, such as the high-order extension of spectral methods for graphs and the second class deals with arbitrary hypergraphs, which model mixed higher-order relationships.

Let $\mathbf{A} = \mathbf{D}_u^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_u^{-1/2}$ be the adjacency matrix $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$. $\mathbf{A}$ is a symmetric matrix as the diagonal matrices $\mathbf{W}$ and $\mathbf{D}_e^{-1}$ commute in multiplication. $\mathbf{A}$ has size $|V| \times |V|$ and its elements, $A(j,i)$, indicate the relatedness between the vertices j and i. Then, $\mathbf{L} = \mathbf{I} - \mathbf{A}$ is known as *Zhou's normalized Laplacian* of the hypergraph [10].

Chapter 4

An Efficient Annotation and Recommendation Model Based on PLSA and Hypergraph Learning

4.1 Introduction

Nowadays, the overwhelming number of images uploaded on the web in conjuction with the constantly rising popularity of social media sharing platforms has led to an indisputable need for efficient annotation and recommender systems. Browsing through this vast volume of data resorts primarily to search engines, which harness the text information provided by the image tags or the image title. Image annotation is a crucial procedure, as it affects considerably both the retrieval accuracy of search engines and the organization of the images uploaded on the web. Several photo sharing websites like `Flickr`[1], `Instagram`[2] or `PicasaWebAlbum`[3] enable the users to annotate images, facilitating image search and content description, bridging this way the gap between the semantic and the visual content of an image. However, quite often, the user provided tags are far from being accurate, some are redundant, or even contain false information. Consequently, an accurate and efficient annotation model is of crucial importance.

Tourism is a vital economic sector for Greece as well as other countries. Nowadays, the

[1]https://www.flickr.com/

[2]http://instagram.com/

[3]http://picasa.google.com/

sector of e-tourism is thriving, launching a plethora of initiatives or entrepreneurial plans. The way people choose their tourist destination has evolved. Traditional ways, such as brochures or a simple web search have been substituted by tourist recommender systems. Despite the effort made so far, there are persisting problems in achieving satisfactory efficiency and accuracy to be addressed, amplifying the need of efficient tourism recommender systems. In this Chapter, a tourism recommender system is proposed and experimental results are disclosed, demonstrating its great potential.

In the past, many efforts have been made toward handling large scale tourism-related image datasets. In [11], the problem of the vast amount of images was handled by building an Internet landmark recognition engine, resorting to efficient object recognition and unsupervised clustering techniques. In [12], a cluster-based landmark and event detection scheme was presented that was based on clustering performed on both visual and tag similarity graphs. In [13], the problem of landmark classification in large-scale image datasets was addressed. Models for these landmarks were learned by a multiclass support vector machine, using vector-quantized interest point descriptors as features. The classification rate was significantly improved by using temporal constraints and textual information distilled from image tags.

Image annotation has been studied extensively in the past using probabilistic models or graph-based methods. In [14], an image and video annotation model was proposed based on the joint probability distribution of tags and image feature vectors. The tag probabilities were computed, using a multiple Bernoulli model and the probabilities of image features were obtained, using non-parametric kernel density estimates. In [15] and [16], image auto-annotation models were implemented by applying the probabilistic latent semantic analysis (PLSA) [17]. In [18], image annotation was refined by integrating the PLSA with a Random Walk model. The PLSA was employed to estimate the posterior probabilities of each tag for an image. Next, a label similarity graph was constructed by a weighted linear combination of labels and visual similarities. A random walk process over the label graph was performed, mining the correlation of the tags and further refining this way the image annotation process. A joint probabilistic model was proposed for simultaneous image classification and annotation in [19]. It was based on a multi-class extension of the supervised Latent Dirichlet Analysis (sLDA) introduced in [20]. Graph-based methods were proposed for tagging in [21] and [22], capturing the information from multi-type interrelated objects. In [23], a k-NN

sparse graph-based semi-supervised approach was proposed for image annotation. A label training refinement strategy was proposed within this k-NN graph-based learning framework, handling the noise in the training labels by using a dual regularization for both the quantity and sparsity of the noise.

Hypergraph models have been exploited in many works, dealing with recommendation problems. A unified hypergraph model was proposed for music recommendation [24]. Music recommendation was treated as a hypergraph ranking problem, using both social media information and music acoustic-based content. A hypergraph model was employed for personalized image recommendation, harnessing the high-order relations among the users, the tags, and the images [25]. The recommendation results were improved by enforcing group sparsity constraints. Image tagging and geo-location prediction were treated simultaneously, modelling the users, the user social groups, the images, the image tags, and the geo-tags within a unified hypergraph framework [26]. The hypergraph learning process was accelerated, using Singular Value Decomposition (SVD) analysis [27]. News recommendation was formulated as a ranking problem on fined-grained hypergraphs and a transductive inference approach was proposed to solve the so-called *cold-start* problem, (i.e., the problem of recommending news on a new user without much reading history) [28]. Ranking on hypergraphs has been also employed in [29, 30]. A related work in tourism recommendation is that of L. Cao *et al.* [31], where recommendation was based on clustering of geotagged images by location and visual matching. Moreover, hypergraphs were applied to exploit visual duplicates for video re-ranking using a random walk algorithm in [32]. In [33], a new transductive hypergraph learning framework for image retrieval was proposed. Images were modeled as hypergraph vertices and the problem of image retrieval was treated as a ranking problem on a probabilistic hypergraph.

The main contribution of this work is in the development of a complete image annotation and tourism recommender system, extending and complementing the work presented in [34]. In particular, the problems of semantic image annotation, content-based image classification, and tourism recommendation are addressed in a unified framework. To begin with, geotagged images crawled from `Flickr` are clustered by means of the GPS coordinates (latitude, longitude), forming several geographical clusters, called *geo-clusters* hereafter. The geo-clusters are then sorted according to their density (i.e., the number of images they contain). This way, places of interest (POIs) are defined. The underlying rationale is that popular

tourist destinations attract more visitors, who upload more geo-tagged photos to the web. The text information (e.g., titles, tags) associated to the images that belong to each geo-cluster is concatenated, forming a geo-cluster derived document. A term-document matrix is then created, and the PLSA [17, 35, 36], properly initialized, is applied to it. The PLSA is used to represent the documents as probability distributions of topics treated as unobserved class variables. By applying the PLSA to a term-document matrix, the relations between the terms and the documents are captured by observing the probability distribution between the documents and the generated topics as well as between the topics and terms. Here, the PLSA is the heart of the method proposed for semantic image annotation and pertains the hypergraph ranking employed for tourism recommendation. The experimental results obtained by replacing the PLSA with the LDA [37], did not show any improvement. Thus, the PLSA is preferred due to its simplicity. The semantic annotation is performed by assigning the most strongly related terms with the geo-cluster derived document to the associated geo-cluster as well as all images belonging to it.

Semantic image annotation is complemented by exploiting the visual attributes of images. The visual content annotation is based on visual content classification, treating the class label as a global image description. Here, the Support Vector Machine (SVM) classifier [38] is fed by GIST descriptors [39] in order to classify each image to a predefined number of classes, outperforming the simple method used in [34]. Among the most common kernel functions used in the SVM are the linear, the polynomial, the Gaussian radial basis function (RBF), and the sigmoid. It was found that the RBF kernel outperforms the other ones.

Tourism recommendation is addressed using a hypergraph. A hypergraph is defined as a set of vertices made by concatenating different kinds of objects (e.g., documents, topics, terms) and hyperedges linking these vertices. In contrast to simple graphs, multi-link relations between the vertices are captured by hypergraphs [4]. Here, the hypergraph vertices are the annotation terms, the geo-cluster derived documents, and the latent topics derived by the PLSA. This way, the relations computed by the PLSA between the geo-cluster derived documents and the topics as well as the vocabulary terms are modeled. Tourism recommendation is treated as a hypergraph ranking problem and the top ranked geo-clusters are recommended as tourist destinations. That is, the recommendation resorts to the hypergraph ranking enhanced by enforcing group sparsity constraints, such as the group Least Absolute Shrinkage and Selection Operator (LASSO) [40, 41, 42]. This way, both the spar-

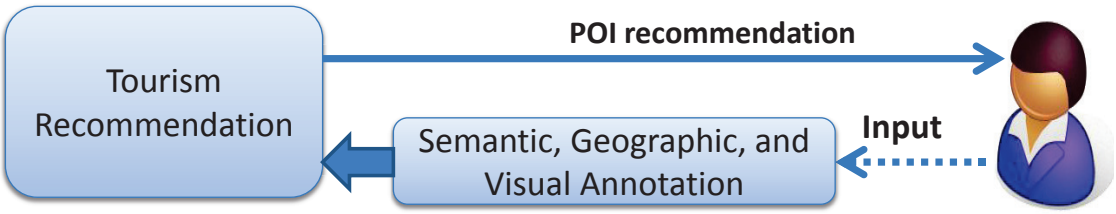

Figure 4.1: Annotation and recommender system.

sity and the group structure of the data are exploited. The effect of each object group in the recommendation process is controlled separately by assigning them different weights. Accordingly, a more advanced method is developed than that in [34]. In order to accelerate the solution of the optimization problem, an efficient method is proposed, using the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [43]. This algorithm is an extension of the Iterative Shrinkage-Thresholding Algorithm (ISTA), preserving its computational simplicity, but achieving a significantly faster rate of convergence.

The block diagram of the proposed model is depicted in Fig. 4.1. The user gives a test image as input to the system and the image is annotated semantically, geographically, and visually, as described in Sections 4.3.1 and 4.3.3. Proceeding to tourism recommendation, the query vector is set, as in Section 4.4.2. The hypergraph model is formed by the geo-cluster derived documents, topics, and terms, exploiting the results of the annotation process. Hypergraph ranking is applied to geo-cluster derived documents, topics, and terms and the top ranked geo-cluster documents are recommended as tourism POIs.

Promising experimental results are disclosed. In particular, an average precision of 92% at 10% recall is reported for semantic image annotation. The accuracy of content-based image classification of 310 test images over 14 classes is $82,6\%$. For tourism recommendation, an average precision of 92% is measured at 1% recall, indicating the effectiveness of the proposed recommendation method.

The structure of this chapter is organized as follows. The dataset and the tourism POI identification method are described in Section 4.2. In Section 4.3, the image annotation is detailed. Hypergraph construction is explained in Section 4.4.1. The hypergraph ranking model is analyzed in Section 4.4.2. The optimization solution enforcing group sparsity is detailed in Sections 4.4.3 and 4.4.4. The outline of the proposed system is presented in Section 4.5. In Section 4.6, experimental results are presented, demonstrating the effectiveness

of the proposed method. Conclusions are drawn and topics for future research are suggested in Section 4.7.

4.2 Dataset and Tourism POI Identification

Popular tourist destinations attract more visitors, who upload more geo-tagged images on social media sharing platforms. To properly organize such geo-tagged images into geographical clusters, an hierarchical clustering algorithm was applied, that is based on geographical distances computed by the "Haversine formula"[4]. Thus, from 50000 geo-tagged randomly selected images related to Greece, that were collected from `Flickr`, 4660 geo-clusters were formed. From these geo-clusters, only the 500 most dense were considered as tourism POIs, containing 31814 images. The number of 500 was chosen taking into account the trade-off between the computational complexity and the information preservation. Next, a document was created for each geo-cluster by concatenating the text information (e.g., title, tags) available in all the images assigned to the geo-cluster.

Text information related to 150000 images, not including the 50000 images mentioned above, was also crawled from `Flickr` in order to properly capture the context of the tourism application. All characters were converted to lower case. Unreadable symbols and redundant information were removed. A vocabulary of unique words was generated along with their frequencies. Terms appearing with frequency less than a threshold (e.g., 100) were eliminated, yielding a vocabulary of 1901 terms.

4.3 Image Annotation

4.3.1 Image Annotation Using Semantic Topics

The PLSA performs a probabilistic mixture decomposition, which associates an unobserved class variable to co-occurrences of terms and documents. The PLSA models each term in a document as a sample from a mixture model. The mixture components are multinomial random variables that can be interpreted as topic representations. The data generation process can be described as [17, 35]: 1) select a document d with probability $P(d)$, 2) pick

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

a latent topic z with probability $P(z|d)$ and, 3) generate a term t with probability $P(t|z)$.

Let $t \in T = \{t_1, t_2, \cdots, t_k\}$ be a vocabulary term and $d \in D = \{d_1, d_2, \cdots, d_m\}$ denote a document. The joint probability model is defined by the mixture:

$$P(t,d) = P(d)\ P(t|d) \quad \text{where} \tag{4.1}$$

$$P(t|d) = \sum_{z \in Z} P(t|z)\ P(z|d) \tag{4.2}$$

and $z \in Z = \{z_1, z_2, \cdots, z_n\}$ is an unobserved class variable representing the topics. As is indicated in (4.2), the document specific term distribution $P(t|d)$ is a convex combination of the n topic dependent distributions $P(t|z)$. In order to determine $P(d)$, $P(z|d)$, and $P(t|z)$, the log-likelihood function

$$\mathcal{L} = \sum_{d \in D} \sum_{t \in T} \tilde{n}(d,t)\ \log P(d,t) \tag{4.3}$$

has to be maximized with respect to all the aforementioned probabilities. In (4.3), $\tilde{n}(d,t)$ denotes the term-document frequency. That is, the number of times t occurs in d. The estimation of $P(d)$ can be carried out independently resulting in $P(d) = \frac{\tilde{n}(d)}{\sum_{d' \in D} \tilde{n}(d')}$. The conditional probabilities $P(z|d)$ and $P(t|z)$ are estimated by means of the EM algorithm [44, 45], which alternates between the *Expectation (E)-step:*

$$\hat{P}(z|d,t) = \frac{P(t|z)P(z|d)}{\sum_{z' \in Z} P(t|z')P(z'|d)}. \tag{4.4}$$

and the *Maximization (M)-step:*

$$P(t|z) = \frac{\sum_{d \in D} \tilde{n}(d,t)\hat{P}(z|d,t)}{\sum_{d \in D} \sum_{t' \in T} \tilde{n}(d,t')\hat{P}(z|d,t')} \tag{4.5}$$

$$P(z|d) = \frac{\sum_{t \in T} \tilde{n}(d,t)\hat{P}(z|d,t)}{\tilde{n}(d)}. \tag{4.6}$$

By alternating (4.4) with (4.5)-(4.6), a convergent procedure is obtained to a local maximum of the log-likelihood. The annotation procedure is performed as follows:

1 The PLSA is applied to the term-document matrix $\mathbf{\Theta} \in \mathbb{R}^{k \times m}$. Here, the documents are formed by concatenating any terms in the tags or the title of the images that belong to a geo-cluster, forming the columns of $\mathbf{\Theta}$. That is, a document $d \in D$ is represented by a vector of size k, having as elements the frequency of occurrence of each term in d.

2 For each document to be annotated, the most related topic is chosen, that with the highest probability, i.e., $z^* = \arg\max_{z \in Z} P(z|d)$.

3 The $k' << k$ most related terms to z^* are identified by sorting $P(t|z^*)$ in decreasing order of magnitude. k' is the same for any hidden topic.

Here, the number of hidden topics is set empirically. However, there are techniques, such as those based on information criteria (e.g., the Bayesian Information Criterion or the method in [46]) that could be adapted to PLSA. The term document matrix $\mathbf{\Theta}$ is of size 1901×500. Among the most descriptive terms of a document, those providing geographical information are identified using geo-gazetteers[5]. Thus, a complete annotation model is built, which provides geographic information in addition to the semantic information.

4.3.2 PLSA Initialization

The PLSA depends on proper initialization method. In addition to the common random initialization, there are many other schemes, e.g., the Random C (RC) or the Random Acol (RAcol) [35]. A variant of RC is the Dense Random C (DRC) summarized in Algorithm 1. The DRC treats the columns of $\mathbf{\Theta}$ unequally. Only the most dense columns are chosen, as they provide more valuable information. The reduction of the number of columns makes the method less time consuming. The DRC was found to be more effective than the RC and the RAcol in the experiments conducted.

4.3.3 Classification-based Visual Content Annotation

The visual features of an image provide valuable, complementary, information about its content. Image annotation is strongly related to image classification, considering the class label as a global description of the image, while the tags are treated as local description of the individual image parts. In order to construct a proper visual dataset, an image subset made of images without occlusion or unwanted noise was manually extracted from the aforementioned dataset of 31814 images and annotated using l different class labels. Here, the different visual classes have been manually defined, capturing the various themes,

[5]http://www.geonames.org

Algorithm 1 Dense Random C Initialization

Input: matrix $\mathbf{\Theta} \in \mathbb{R}^{k \times m}$ with $\Theta(i,j) \geq 0$.

Output: matrix $\mathbf{S} \in \mathbb{R}^{k \times n}$, containing the conditional probabilities $P(t|d)$.

1 Count the non-zero elements of each column of $\mathbf{\Theta}$.

2 Compute the mean document vector $\overline{\mu}$.

3 Find the ξ columns of $\mathbf{\Theta}$, having more non-zero elements than $\overline{\mu}$.

4 Average $\tilde{\zeta}$ randomly chosen columns out of the ξ and set the average column vector as a column of $\mathbf{S}$. Repeat 3-4 for all columns of $\mathbf{S}$.

pertaining the image dataset. Each visual topic (class), was assigned manually one label and a few representative tags, e.g., clouds, sky, sea, sunset, defining the image visual content.

The GIST descriptor [39], was extracted from any image $\phi \in \Phi = \{\phi_1, \phi_2, \cdots, \phi_N\}$, where Φ is a smaller image subset, made of images without occlusion or unwanted noise. Φ is manually extracted and annotated using the l class labels. The GIST descriptors are global descriptors initially proposed in [39]. Each image ϕ is segmented by a 4-by-4 grid for which orientation histograms are extracted, producing a final vector of low dimensionality, the GIST descriptor. The GIST descriptors have recently demonstrated good results for image search tasks on large databases [47]. Local descriptors, such as SIFT [48] represented in a bag of features approach, were not used due to their heavy computational complexity and their memory needs.

An SVM classifier was trained using the GIST descriptors G as input vectors. Each test image was classified into one among the l classes by applying the SVM classifier fed by the GIST descriptors of the test images. Here, the RBF kernel was employed. It was found that it outperforms the other commonly used kernels (i.e., the linear, the polynomial, and the sigmoid). The multiclass problem was treated by applying the "one against one" strategy, also known as pairwise coupling. It is based on the construction of one SVM for each pair of classes. Thus, for a problem with l classes, $l(l-1)/2$ SVMs are trained to recognize the samples of one class from the samples of its rivals. Each SVM votes for one class. The classification of a test sample is achieved by majority voting. Representative images assigned to several classes are shown in Fig. 4.2.

Figure 4.2: A sample of 16 images, representing several visual classes. Top raw: First image belongs to visual class sea; Second image belongs to visual class mills; Third image belongs to visual class sky; Fourth image belongs to visual class White Tower. Second raw: First image belongs to visual class Meteora; Second image belongs to visual class Byzantine; Third image belongs to visual class artifacts; Fourth image belongs to visual class Parthenon. Third raw: Second image belongs to visual class city scene. Fourth raw: Second image belongs to visual class plants, and Fourth image belongs to image class sunset.

4.4 Tourism Recommendation

The second part of the proposed system consists of a hypergraph model representing the multi-link relations between terms of the vocabulary, documents (geo-clusters), and topics as they were computed in Sec. 4.3.1.

4.4.1 Hypergraph Construction

A hypergraph $\mathbf{H}$ having size of 2751×1000 elements was formed by concatenating 500 documents associated to the geo-clusters, 350 topics, z, and 1901 vocabulary terms, t. The vertex set is defined as $V = D \cup Z \cup T$. The structure of the hypergraph incidence matrix is summarized in Table 4.1. For each document d_j associated to a geo-cluster, a hyperedge e_1 is inserted as a column of $\mathbf{De}_1, \mathbf{Ze}_1, \mathbf{Te}_1$, containing 1 in the j-th entry of $\mathbf{De}_1$, 1 for the most related topic to d_j, z^* in $\mathbf{Ze}_1$, and 30 ones for the 30 most descriptive terms t for z^*, in $\mathbf{Te}_1$. The weight for this hyperedge is $w(e_1) = P(z^*|d_j)$. To capture the geographical proximity, hyperedges $e_2 \in E_2$ are created. For each d_j corresponding to a specific geo-cluster, one hyperedge e_2 is inserted as a column of $\mathbf{De}_2$ having 1 to its j-th entry, as well as to the entries corresponding to geo-clusters being at a geographical distance less than 150 km. The weight for this hyperedge is set to 1.

Table 4.1: Structure of $\mathbf{H}$.

	e_1	e_2
D	$\mathbf{De}_1$	$\mathbf{De}_2$
Z	$\mathbf{Ze}_1$	0
T	$\mathbf{Te}_1$	0

4.4.2 Hypergraph Model

Let $G(V, E, w)$ denotes a hypergraph with set of vertices V and set of hyperedges E to which a weight function $w : E \to \mathbb{R}$ is assigned. V consists of sets of objects of different type (i.e., documents, topics, terms). A $|V| \times |E|$ incidence matrix $\mathbf{H}$ is formed as in (3.1). Based on $\mathbf{H}$, the vertex and hyperedge degrees are defined as in (3.2) and (3.3) respectively. 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. $\mathbf{A}$ has size $|V| \times |V|$ and its elements, $A(j,i)$, indicate the relatedness between the vertices j and i. Considering j, i as two vertices of the same type (e.g., document(j), document(i)) the element $A(i,j)$ indicates the similarity of the two documents. In case j, i corresponds to two vertices of different type (e.g., document(j), term(i)) then the element $A(j,i)$ indicates how much the document(j) is related to the term(i). Then, $\mathbf{L} = \mathbf{I} - \mathbf{A}$ is known as Zhou's normalized Laplacian of the hypergraph [10]. 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:

$$\Omega(\mathbf{f}) = \mathbf{f}^T\mathbf{L}\mathbf{f}. \tag{4.7}$$

in terms of the ranking vector $\mathbf{f}$. That is, one requires all vertices with the same value in the ranking vector $\mathbf{f}$ to be strongly connected [49]. $\Omega(\mathbf{f})$ is small, if vertices with high affinities have the same label. For instance, two documents are probably similar, if they are linked with many common topics and textual terms. The aforementioned optimization problem was extended to solve a recommendation problem by including the ℓ_2 regularization norm between the ranking vector $\mathbf{f}$ and a query vector $\mathbf{y} \in \mathbb{R}^{|V|}$ in [24]. This guarantees that the ranking vector does not differ too much from the initial query vector $\mathbf{y}$. The function to be minimized is then expressed as

$$Q(\mathbf{f}) = \Omega(\mathbf{f}) + \vartheta \, ||\mathbf{f} - \mathbf{y}||^2 \tag{4.8}$$

where ϑ is a regularizing parameter. The best ranking vector, $\mathbf{f}^* = \arg\min_{\mathbf{f}} Q(\mathbf{f})$, is found to be [24]:

$$\mathbf{f}^* = \frac{\vartheta}{1+\vartheta}\left(\mathbf{I} - \frac{1}{1+\vartheta}\mathbf{A}\right)^{-1}\mathbf{y}. \tag{4.9}$$

Let d'_j be the geo-cluster where the test image ϕ_{test} belongs to with respect to its geo-tag. The query vector $\mathbf{y} \in \mathbb{R}^{|V|}$ is defined as:

$$y(v) = \begin{cases} 1, & \text{if } v = d'_j \\ A(d'_j, v), & \text{otherwise} \end{cases} \tag{4.10}$$

treating $A(d'_j, v)$ as a measure of relatedness between the vertices of the hypergraph.

The ranking vector $\mathbf{f}^* \in \mathbb{R}^{|V|}$ has the same size and structure as $\mathbf{y}$. It corresponds to a significant query of the system. In other terms, for every query vector $\mathbf{y}$ the optimization procedure presented in (4.8) is performed and the values of the ranking vector $\mathbf{f}$ corresponding to the entries associated to geo-cluster documents are used as rankings for tourist destination recommendation. The top ranked geo-cluster documents are recommended as tourism POIs to the user, who has imported the test image ϕ_{test}.

4.4.3 Group Sparse Regularization

The hypergraph vertices are split into Δ non-overlapping object groups (geo-cluster derived documents, topics, terms). Undoubtedly, each object group contributes differently to the ranking procedure. A Group Lasso regularizing term is more appropriate than the ℓ_2 norm in this kind of problems [41]. Accordingly, different weights γ_δ, $\delta = 1, 2, \ldots, \Delta$ are assigned to each object group, yielding the following objective function to be minimized:

$$\tilde{Q}(\mathbf{f}) = \Omega(\mathbf{f}) + \vartheta \sum_{\delta=1}^{\Delta} \sqrt{\gamma_\delta \, (\mathbf{f}-\mathbf{y})^T \mathbf{K}_\delta (\mathbf{f}-\mathbf{y})}. \tag{4.11}$$

In (4.11), ϑ is a positive regularization parameter and $\mathbf{K}_\delta$ is the $|V| \times |V|$ diagonal matrix with elements equal to 1 for the vertices, which belong to the δ-th object group.

4.4.4 FISTA Solution

In order to solve the minimization problem (4.11) by applying the FISTA algorithm, one needs to transform the cost function $\Omega(\mathbf{f})$ to the following ℓ_2 norm [43]:

$$\Omega(\mathbf{f}) = \left|\left|\tilde{\mathbf{A}}\mathbf{x} - \mathbf{b}\right|\right|^2. \tag{4.12}$$

By introducing the auxiliary variable $\mathbf{x} = \mathbf{f} - \mathbf{y}$, the cost function (4.7) is rewritten as:

$$\Omega(\mathbf{f}) = (\mathbf{x}+\mathbf{y})^T \mathbf{L} (\mathbf{x}+\mathbf{y}). \tag{4.13}$$

The fact that $\mathbf{L}$ is a symmetric and positive definite matrix implies that its Cholesky factor $\mathbf{L}^{\frac{1}{2}}$ is also a symmetric positive definite matrix. Thus, the cost function (4.13) is rewritten

as:

$$\begin{aligned}\Omega(\mathbf{f}) &= (\mathbf{x}+\mathbf{y})^T\mathbf{L}^{\frac{1}{2}}\mathbf{L}^{\frac{1}{2}}(\mathbf{x}+\mathbf{y}) \\ &= \left[\mathbf{L}^{\frac{1}{2}}(\mathbf{x}+\mathbf{y})\right]^T\left[\mathbf{L}^{\frac{1}{2}}(\mathbf{x}+\mathbf{y})\right] \\ &= \left|\left|\mathbf{L}^{\frac{1}{2}}(\mathbf{x}+\mathbf{y})\right|\right|^2. \end{aligned} \tag{4.14}$$

From Equations (4.12) and (4.14), it is deduced that:

$$\tilde{\mathbf{A}} = \mathbf{L}^{\frac{1}{2}}, \tag{4.15}$$

$$\mathbf{b} = -\mathbf{L}^{\frac{1}{2}}\mathbf{y}. \tag{4.16}$$

The optimization problem (4.11) takes the form:

$$\tilde{Q}(\mathbf{x}) = g(\mathbf{x}) + q(\mathbf{x}) \tag{4.17}$$

where $g(\mathbf{x}) = \left|\left|\mathbf{L}^{\frac{1}{2}}(\mathbf{x}+\mathbf{y})\right|\right|^2$, $q(\mathbf{x}) = \vartheta \sum_{\delta=1}^{\Delta}\sqrt{\gamma_\delta}||\mathbf{x}_\delta||$, and $\mathbf{x}_\delta$ refers to the entries of $\mathbf{x}$ related to group δ.

Given a point $\mathbf{x}$, the ISTA approach applied to $\tilde{Q}(\mathbf{x})$ minimizes the sum of $q(\mathbf{x})$ and a quadratic approximation of $g(\mathbf{x})$ at each iteration, i.e.

$$\begin{aligned}\mathbf{x}_{new} &= \arg\min_{\tilde{\mathbf{z}}}\left\{g(\mathbf{x}) + \nabla^T g(\mathbf{x})(\tilde{\mathbf{z}}-\mathbf{x}) + \frac{1}{2\varrho}||\tilde{\mathbf{z}}-\mathbf{x}||^2 + q(\tilde{\mathbf{z}})\right\} \\ &= \arg\min_{\tilde{\mathbf{z}}}\left\{\sum_\delta\left\{\frac{1}{\varrho}||\tilde{\mathbf{z}}_\delta - \tilde{\mathbf{d}}(\mathbf{x})_\delta||^2 + \vartheta\sqrt{\gamma_\delta}||\tilde{\mathbf{z}}_\delta||\right\}\right\} \end{aligned} \tag{4.18}$$

where ϱ is the step-length and $\tilde{\mathbf{d}}(\mathbf{x}) = \mathbf{x} - 2\varrho\nabla g(\mathbf{x})$.

The optimization problem (4.18) is separable. The solution of each of the Δ subproblems given a soft-thresholding operation

$$\mathbf{x}_{new} = \frac{\tilde{\mathbf{d}}(\mathbf{x})_\delta}{||\tilde{\mathbf{d}}(\mathbf{x})_\delta||}\max\left\{0, ||\tilde{\mathbf{d}}(\mathbf{x})_\delta|| - \vartheta\sqrt{\gamma_\delta}\varrho\right\} \tag{4.19}$$

The step-length ϱ satisfies the inequality $\varrho \leq \frac{1}{\eta}$, where η is the maximal eigenvalue of $\mathbf{L}$ (i.e., the Lipschitz constant for $\nabla g(\mathbf{x})$).

Algorithm 2 summarizes the steps of the FISTA for $\varrho = \frac{1}{\eta}$. $\mathbf{x}^\infty$ denotes the vector $\mathbf{x}$ upon the convergence of the while loop in Algorithm 2.

Algorithm 2 Solving (4.11) with FISTA.

Input: The Laplacian matrix $\mathbf{L} \in \mathbb{R}^{|V|\times|V|}$, the query vector $\mathbf{y} \in \mathbb{R}^{|V|}$, $\gamma_\delta, \delta = 1, 2, \cdots, \Delta$, parameter ϑ, and maximum eigenvalue of $\mathbf{L}$, η.

Output: The ranking vector $\mathbf{f} \in \mathbb{R}^{|V|}$.

Initialize $\tau = 1,\ \zeta^{(1)} = 1,\ \mathbf{x}^{(0)},\ \tilde{\mathbf{z}}^{(1)} = \mathbf{x}^{(0)}$.

while not converged do

  $\tilde{\mathbf{d}}(\tilde{\mathbf{z}}^{(\tau)}) = \tilde{\mathbf{z}}^{(\tau)} - \frac{2}{\eta}\mathbf{L}(\tilde{\mathbf{z}}^{(\tau)} + \mathbf{y})$

  for $\delta = 1 \cdots \Delta$ do

    $\mathbf{x}_\delta^{(\tau)} = \frac{\tilde{\mathbf{d}}(\tilde{\mathbf{z}}^{(\tau)})_\delta}{||\tilde{\mathbf{d}}(\tilde{\mathbf{z}}^{(\tau)})_\delta||} \max\left\{0, ||\tilde{\mathbf{d}}(\tilde{\mathbf{z}}^{(\tau)})_\delta|| - \vartheta\frac{\sqrt{\gamma_\delta}}{\eta}\right\}$

  end

  $\zeta^{(\tau+1)} = \frac{1+\sqrt{1+4(\zeta^{(\tau)})^2}}{2}$

  $\tilde{\mathbf{z}}^{(\tau+1)} = \mathbf{x}^{(\tau)} + \frac{\zeta^{(\tau)}-1}{\zeta^{(\tau+1)}}(\mathbf{x}^{(\tau)} - \mathbf{x}^{(\tau-1)})$

  $\tau \leftarrow \tau + 1$

end while

$\mathbf{f} = \mathbf{x}^{\infty} + \mathbf{y}$

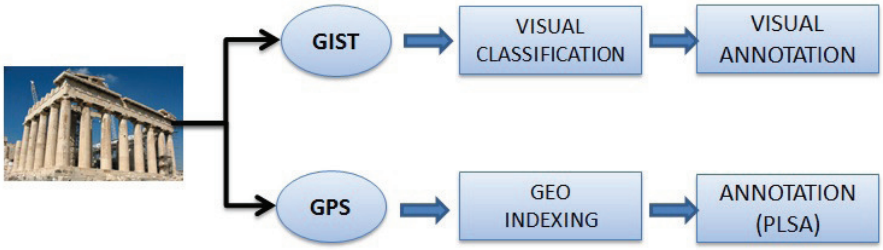

Figure 4.3: Annotation system.

4.5 System Outline

Given an image as input, the distances between image geo-location captured using the GPS technology and the geo-cluster centers are computed. The input image is then assigned to the nearest geo-cluster. Simultaneously, the image visual content is classified by means of an SVM classifier fed by the image GIST descriptor. Next, the class label and the predefined representative tags offer a visual content annotation. Simultaneously, the vocabulary terms assigned to the closest geo-cluster derived document by the PLSA, offer geographic and semantic annotation for the image, as was described in Sec. 4.3.1. Fig. 4.3 demonstrates the proposed annotation system. Proceeding to tourism recommendation, the query vector $\mathbf{y}$ is initialized, as was suggested in Sec. 4.4.2. Hypergraph ranking is applied and the top ranked geo-clusters are recommended as tourist destinations.

4.6 Experimental Results

The averaged Recall-Precision curve, the F_1 measure, and the Mean Average Precision (MAP) are used as figures of merit. Precision is defined as the number of correctly recommended objects divided by the number of all recommended objects. Recall is defined as the number of correctly recommended objects divided by the number of all objects. The F_1 measure is the weighted harmonic mean of precision and recall, which measures the effectiveness of retrieval when treating precision and recall as equally important, i.e.,

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall}. \tag{4.20}$$

The MAP is the mean value of the Average Precision (AP) of all the queries. The AP is defined as the average of precisions computed at the point of each correctly retrieved item,

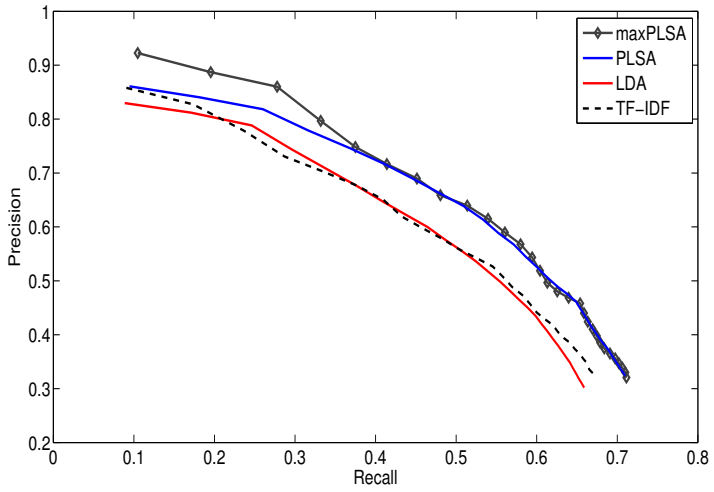

Figure 4.4: Recall-precision for the PLSA, the LDA, and the TF-IDF.

as is shown below:

$$AP = \frac{\sum_{i=1}^{Num} Precision@i \cdot true_i}{cNum} \tag{4.21}$$

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.

For evaluation purposes, a test set containing 310 images was randomly chosen and removed from the training set along with their text information. The PLSA performance in semantic image annotation has been compared to that of the LDA [37] and the term frequency-inverse document frequency (TF-IDF) [50]. As is demonstrated in Fig. 4.4, the PLSA outperforms both the LDA and the TF-IDF method. The recall-precision curves were obtained by averaging recall-precision pairs in 100 repetitions using different initializations. The curve corresponding to the best results at the top ranking positions obtained by using the PLSA is denoted as *maxPLSA*. An average precision of 92% at 10% recall is reported, using the PLSA. It is worth noting, that the PLSA is much simpler than the LDA. In Table 4.2, the averaged F_1 measure corresponding to 7 different ranking positions is listed for the PLSA, the LDA, and the TF-IDF, for semantic image annotation. It is seen that the PLSA outperforms the LDA and the TF-IDF.

In Fig. 4.5, recall-precision curves are plotted for the PLSA, having been initialized by the DRC, the RC, the RAcol, and the Random initialization coined as (Rnd) for 2 and 4 iterations of the Expectation Maximization (EM) algorithm. The average recall precision

Table 4.2: F_1 measure at 7 ranking positions.

	F_1@1	F_1@2	F_1@5	F_1@10	F_1@15	F_1@20	F_1@25
PLSA	0.1704	0.2987	0.4926	0.5707	**0.5546**	**0.5213**	**0.4816**
LDA	0.1601	0.2829	0.4583	0.5275	0.5204	0.4884	0.4510
TF-IDF	0.1641	0.2867	0.4529	0.5261	0.5260	0.4993	0.4613
maxPLSA	**0.1882**	**0.3202**	**0.4996**	**0.5749**	0.5487	0.5175	0.4779

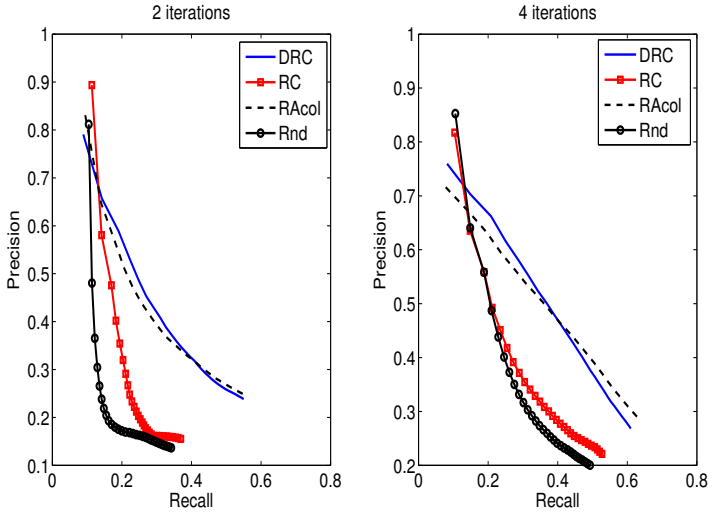

Figure 4.5: DRC, RC, RAcol, and Rnd recall-precision curves for 2 and 4 iterations of the EM.

curves over 100 repetitions of the PLSA training for each initialization are shown. The results indicate that the precision falls with a slower rate for the DRC.

As is depicted in Fig. 4.6, the MAP improves as the number of the topics increases. A plausible compromise between the annotation accuracy and the additional complexity caused by large number of topics, suggests fixing the number of topics to 350. In Fig. 4.7, recall-precision curves are demonstrated for several number of topics. The recall-precision curves were obtained by averaging recall-precision pairs in 100 repetitions.

For visual image classification, the same test set was used. Each test image was assigned into one of 14 representative classes manually in order to form the ground truth. For each image one GIST descriptor is extracted. Visual classification accuracy for the proposed

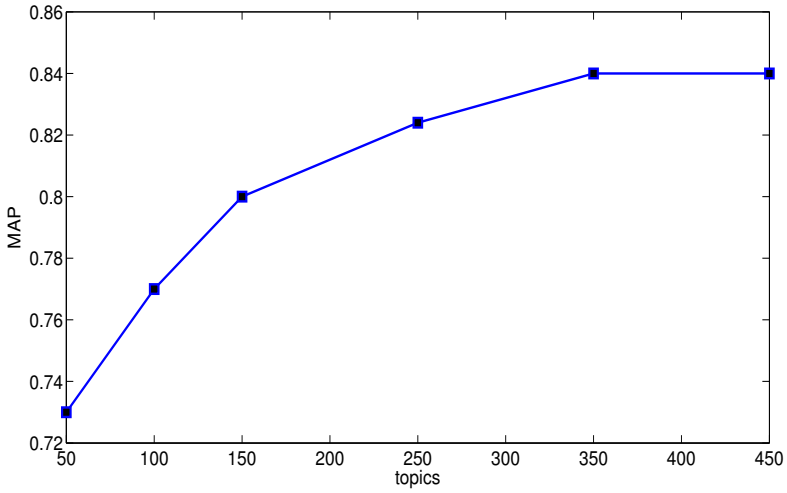

Figure 4.6: MAP vs. the number of topics.

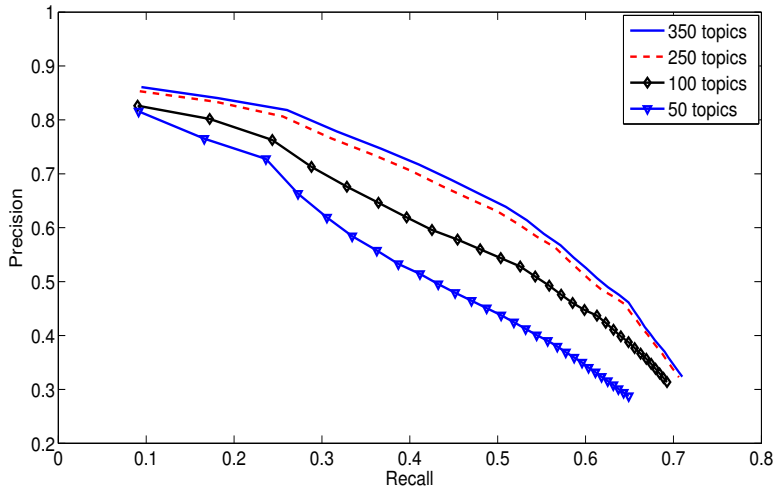

Figure 4.7: PLSA recall-precision curves for several number of topics.

Table 4.3: Accuracy results for the 3 compared methods.

	GS	*GK*	*GMJ*
classification accuracy	**0.826**	0.700	0.607

Table 4.4: Accuracy results for 4 different SVM kernels.

	linear	*polynomial*	*sigmoid*	*RBF*
classification accuracy	0.784	0.784	0.439	**0.826**

approach (GS), the approach presented in [34] (GK) and a simple approach employing GIST descriptors and majority voting (GMJ) is shown in Table. 4.3. The best results were obtained by using the SVM classifier. Across the 310 test set images, the accuracy of content-based image classification over 14 classes is $82,6\%$. In Table 4.4, the classification accuracy is demonstrated for 4 different SVM kernels, the linear, the polynomial, the sigmoid, and the RBF. The RBF kernel clearly outperforms the other 3. The width parameter $\tilde{\gamma}$ of the RBF kernel function $\exp(-\tilde{\gamma}||\chi_i - \chi_j||^2)$ was set equal to 1. In Fig. 4.8, the classification rates over the regularization parameter C that weighs the sum of slack variables [51] are displayed. The classification accuracy is reduced greatly for small values of the cost parameter C. A small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. Both C and $\tilde{\gamma}$ were estimated in a grid search, performing 5-fold cross validation on the training set. In Fig. 4.9, the confusion matrix is shown, demonstrating the classification rates across the classes.

Moreover, experiments were conducted to assess tourism POI recommendation. In order to form the ground truth, relations were established manually among the geo-clusters, taking into account the distance, common geographical entities (e.g., mainland, island) and leisure activities. For this, various tourist related web sources were exploited, such as `TripAdvisor`[6] and `TravelMuse`[7]. Firstly, only the hyperedges $e_1 \in E_1$ were taken into account in hypergraph creation, as in [34]. Secondly, all the hyperedges were considered. Let us denote the first approach as *HR1* and the one exploiting also the geographic information as *HR*. The associated recall-precision curves are plotted in Fig. 4.10. It is seen that the results are increased when both types of hyperedges are considered.

[6]http://www.tripadvisor.com.gr/

[7]http://www.travelmuse.com/

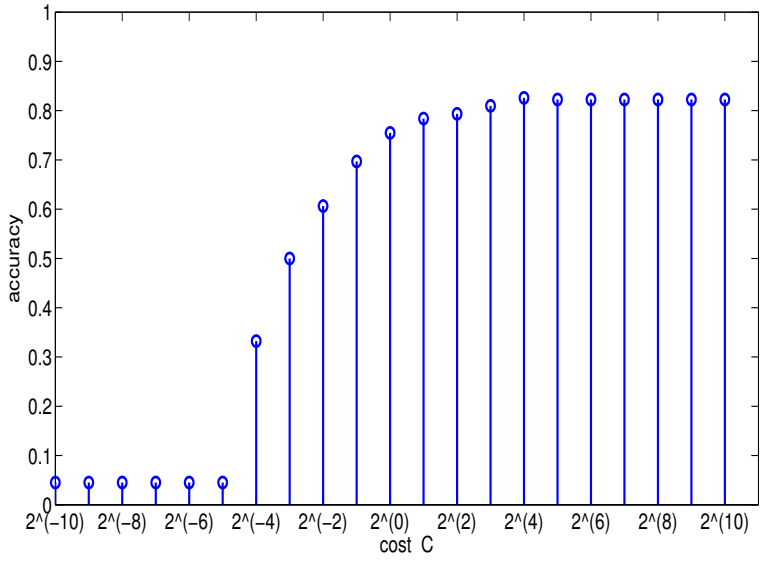

Figure 4.8: Classification accuracy results per cost parameter C.

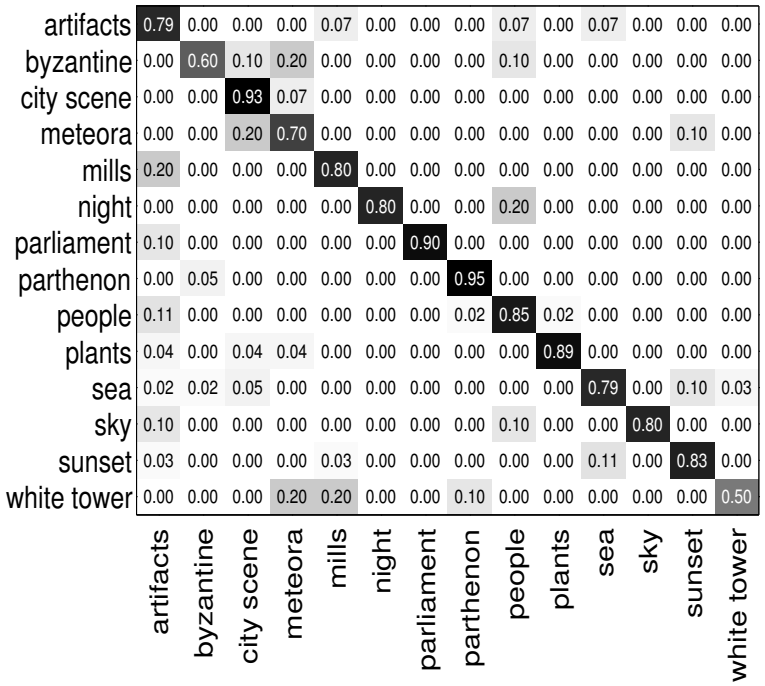

Figure 4.9: Confusion matrix for the visual image classification.

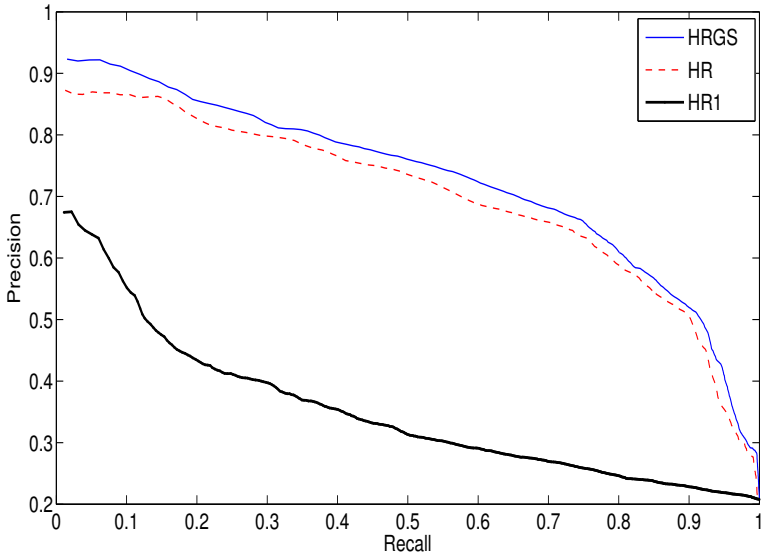

Figure 4.10: Averaged Recall-precision curves for tourism POI recommendation.

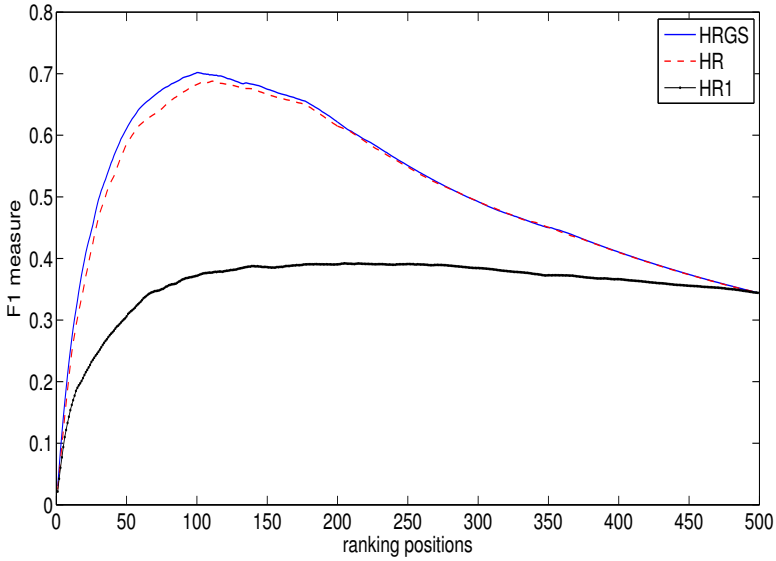

Figure 4.11: Averaged F_1 measure at several ranking positions.

Furthermore, the recommendation is improved by enforcing group sparsity constraints. The tourism POI recommendation enforcing group sparsity is coined as *HRGS*. In Fig. 4.10, the recall-precision curves for tourism POI recommendation are plotted revealing the superiority of the approach proposed here. As is demonstrated in Fig. 4.10, a precision rate of 92% and 91% is achieved for 1% and 10% recall, respectively. The complete curves of the averaged F_1 measure per ranking position for all algorithms compared are displayed in Fig. 4.11. The maximum average F_1 measure equals 0.7019. It is obtained at the ranking position 101. The MAP results for the HRGS, the HR, and the HR1 are 0.741, 0.710, and 0.3647, respectively. Here, the objects were split in 2 object groups, the first is formed by the geo-cluster derived documents and the second consists of the concatenation of both topics and terms. This choice was made by observing the sparsity distribution over the incidence matrix $\mathbf{H}$ of the hypergraph. Indeed, by applying spectral analysis to the hypergraph Laplacian matrix $\mathbf{L}$, these 2 object groups are clearly discriminated as can be seen in Fig. 4.12. The columns of $\mathbf{L}$, are projected into the 2 dimensional space by employing spectral analysis and taking the 2 eigenvectors corresponding to the 2 non-zero smallest eigenvalues of $\mathbf{L}$. The weights for the 2 different object groups were empirically set to 0.9 for the geo-cluster derived documents and to 0.2 for the other group. In Fig. 4.13, a visual example of the proposed recommendation method on the Google maps is demonstrated where the user has inserted as a query an image of Parthenon and the proposed system recommended the Zappeion and the temple of Zeus.

4.7 Conclusions and Future Work

An efficient image annotation and tourism POI recommender system has been proposed and described in full detail in this chapter. After organizing large collections of images by using clustering and classification methods, tourism related images have been annotated geographically, semantically, and visually, harnessing visual attributes and text information. The PLSA has been enhanced by an effective initialization method and used in order to extract semantic information from image metadata. The annotation procedure has been complemented by exploiting the image visual attributes. Furthermore, a tourism recommendation approach has been proposed based on hypergraph ranking and enforcing group sparsity constraints. The solution of the optimization problem was significantly accelerated

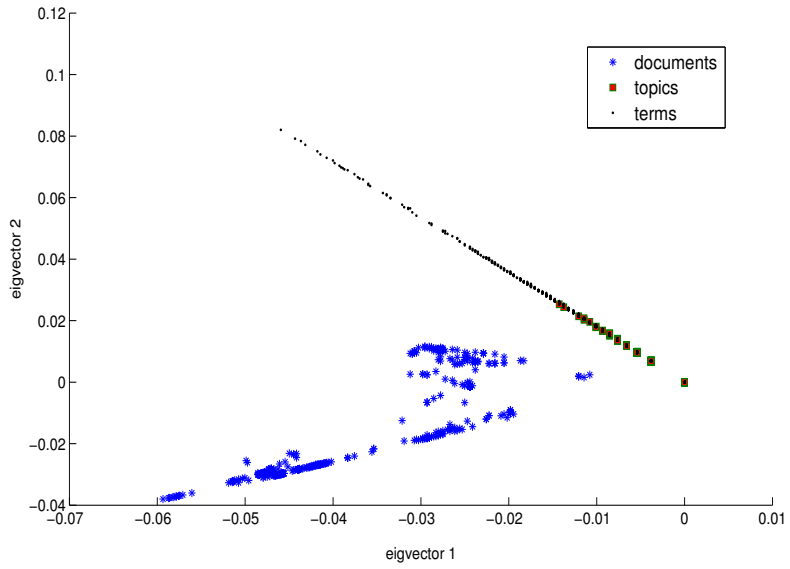

Figure 4.12: Spectral analysis on the hypergraph Laplacian.

via the gradient accelerated method. Subjects of future research might be the exploitation of personalized user information and any other social media information and the on-line updating of the hyperedge weights. Another interesting approach would be the employment of the majorization-minimization algorithm proposed in [52], enhancing the efficiency of the ranking in the experiments.

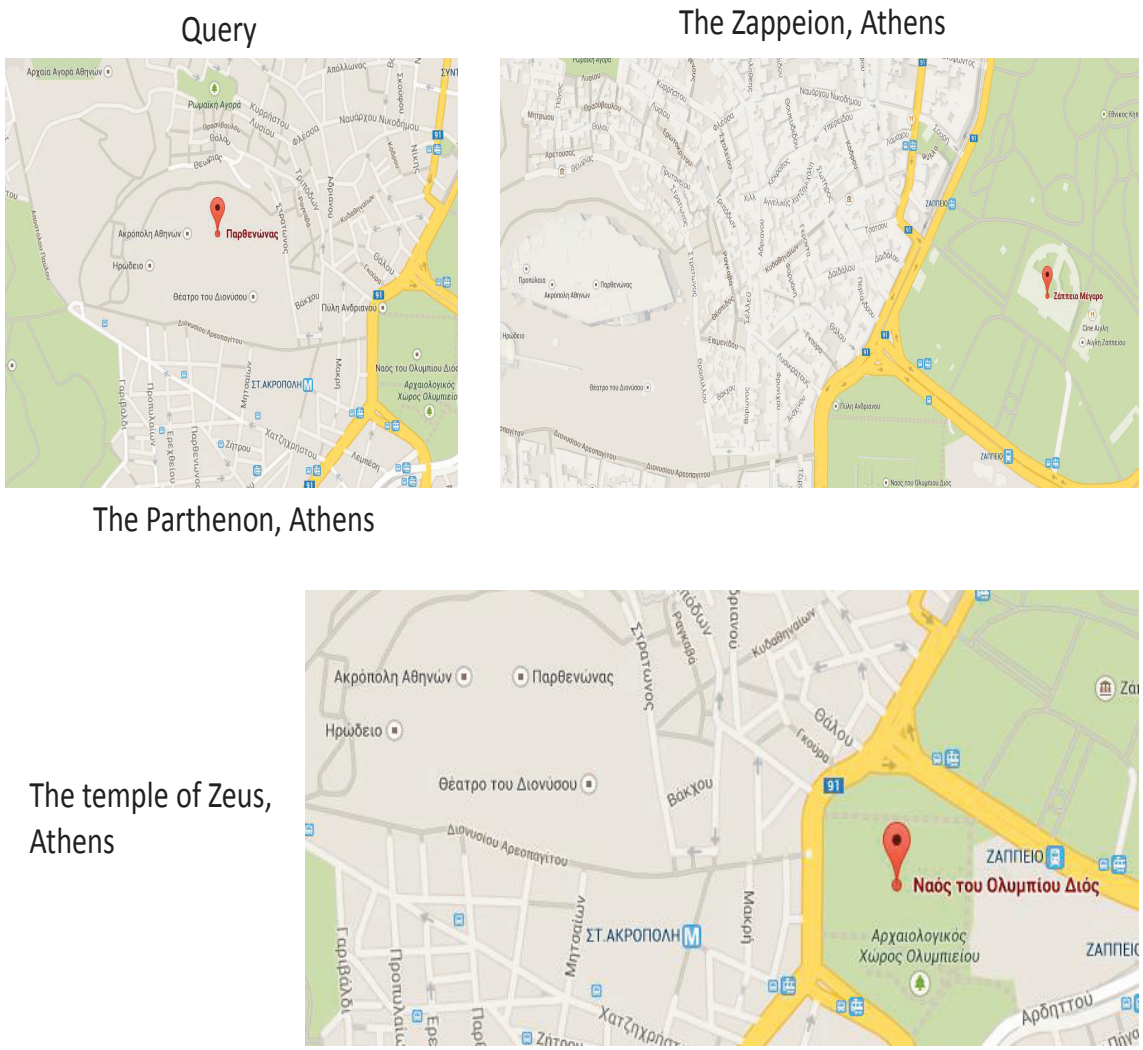

Figure 4.13: A visual example of the recommendation method (Google maps).

Chapter 5

Image Tagging Using Unified Hypergraph Learning Framework

5.1 Introduction

Image tagging is a crucial procedure, affecting considerably both the retrieval accuracy and the organization of the images uploaded on the web, as it was mentioned in the previous chapter. Popular social media sharing platforms, 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. Consequently, an accurate and efficient image tagging model is of crucial importance. In this chapter, a novel image tagging method is detailed, exploiting multiple sources of information, such as social, image-related metadata, or similarities based on visual attributes. The several relations are modeled in a unified hypergraph framework. The method is further improved by employing a hyperedge weight updating scheme.

Graphs and hypergraphs have been employed on various machine learning tasks, exploiting multimodal information. A visual image similarity graph and an image-tag bipartite graph were fused in a unified graph in [53]. 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. Multi-label image annotation was formulated as a regression model with a regularized penalty, exploiting the structural group sparsity in [54]. Furthermore, in [55], a hypergraph was used for classification, modelling the images by their visual attributes. Hypergraph learning was also applied to social

image search [56, 57, 58].

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 [4, 10, 26, 8]. 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 [56], an effective hyperedge weight learning scheme is proposed, treating each hyperedge in a different manner. The novelty of this method is in the analytical solution of the optimization problem, which leads to the weight estimation. Here, the method in [56] is revisited by taking into account the vertex degree matrix that was omitted in [56], 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 [26]. 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 this chapter is as follows. In Section 5.2, the general hypergraph model is introduced and the ranking on a hypergraph is detailed. The hyperedge weight learning method is addressed in Section 5.3. In Section 5.4, the dataset is described. The hypergraph construction is explained in Section 5.5. Experimental results are presented in Section 5.6, demonstrating the merits of the proposed method. Conclusions are drawn in Section 5.7.

5.2 Hypergraph Model

Here, the theoretical background detailed in the previous chapter is revisited for clarity purposes. $G(V, E, w)$ denotes 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.

Let $\mathbf{D}_u$ of size $|V| \times |V|$ be the vertex degree matrix, the $\mathbf{D}_e$ of size $|E| \times |E|$ be the hyperedge degree matrix, and $\mathbf{W}$ be the $|E| \times |E|$ matrix, containing the hyperedge weights.

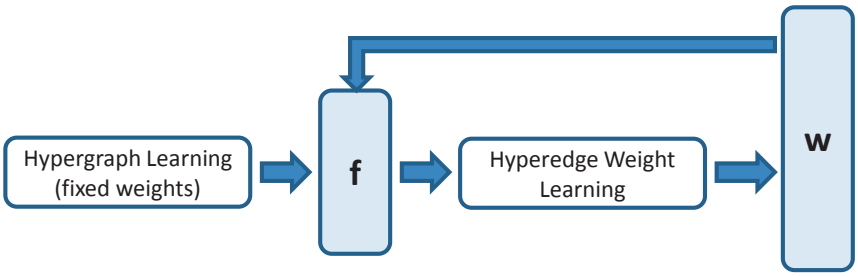

Figure 5.1: Description of the hyperedge weight learning method.

Let also $\mathbf{A} = \mathbf{D}_u^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_u^{-1/2}$ and $\mathbf{L} = \mathbf{I} - \mathbf{A}$ be the Zhou's normalized Laplacian of the hypergraph [10]. The elements of $\mathbf{A}$, $A(j,i)$, indicate the relatedness between the vertices j and i. As has been said in Section 4.4.2, 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:

$$\Omega(\mathbf{f}) = \mathbf{f}^T\mathbf{L}\mathbf{f}. \tag{5.1}$$

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 $\mathbf{y} \in \mathbb{R}^{|V|}$ [24]. Then, the function to be minimized is expressed as

$$Q(\mathbf{f}) = \Omega(\mathbf{f}) + \vartheta \, ||\mathbf{f} - \mathbf{y}||^2 \tag{5.2}$$

where ϑ is a positive regularizing parameter. The best ranking vector, $\mathbf{f}^* = \arg\min_{\mathbf{f}} Q(\mathbf{f})$, is found to be [24]:

$$\mathbf{f}^* = \frac{\vartheta}{1+\vartheta}\left(\mathbf{I} - \frac{1}{1+\vartheta}\mathbf{A}\right)^{-1}\mathbf{y}. \tag{5.3}$$

5.3 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:

$$\underset{\mathbf{f},\mathbf{w}}{\operatorname{argmin}} \left\{ Q(\mathbf{f}) + \kappa ||\mathbf{w}||^2 \right\} \qquad \text{s.t. } \mathbf{1}_n^T \mathbf{w} = 1. \tag{5.4}$$

The method is illustrated in Fig.5.1. An alternating minimization of (5.4) starts with a fixed $\mathbf{w}$ and optimizes $\mathbf{f}$ as in (5.3). Next, $\mathbf{f}$ is fixed and $\mathbf{w}$ is optimized. Having fixed $\mathbf{f}$, the optimization w.r.t. $\mathbf{w}$ is read as:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \left\{ \mathbf{f}^T \mathbf{L} \mathbf{f} + \kappa ||\mathbf{w}||^2 \right\} \qquad \text{s.t. } \mathbf{1}_n^T \mathbf{w} = 1. \tag{5.5}$$

The Lagrangian function of the optimization problem is:

$$\begin{aligned} \Psi(\mathbf{w}, c) &= \mathbf{f}^T \mathbf{L} \mathbf{f} + \kappa \mathbf{w}^T \mathbf{w} + c(\mathbf{1}_n^T \mathbf{w} - 1) \\ &= \mathbf{f}^T \left(\mathbf{I} - \mathbf{D}_u^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_u^{-1/2} \right) \mathbf{f} \\ &\quad + \kappa \mathbf{w}^T \mathbf{w} + c(\mathbf{1}_n^T \mathbf{w} - 1). \end{aligned} \tag{5.6}$$

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

$$\frac{\partial \Psi}{\partial w_i} = \frac{\partial}{\partial w_i} \left(-\mathbf{f}^T \mathbf{\Lambda} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{\Lambda}^T \mathbf{f} \right) + 2\kappa w_i + c = 0. \tag{5.7}$$

Solving (5.7) w.r.t. w_i, we obtain:

$$w_i = \frac{1}{2\kappa} \left[\frac{\partial}{\partial w_i} \left(\mathbf{f}^T \mathbf{\Lambda} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{\Lambda}^T \mathbf{f} \right) - c \right] \tag{5.8}$$

and by substituting (5.8) into the constraint $\mathbf{1}_n^T \mathbf{w} = 1$, the Lagrange multiplier is determined:

$$c = \frac{1}{n} \left[\mathbf{1}_n^T \frac{\partial}{\partial w_i} \left(\mathbf{f}^T \mathbf{\Lambda} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{\Lambda}^T \mathbf{f} \right) - 2\kappa \right]. \tag{5.9}$$

The partial derivative in (5.8) and (5.9) is analysed as:

$$\begin{aligned} \frac{\partial}{\partial w_i} \left(\mathbf{f}^T \mathbf{\Lambda} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{\Lambda}^T \mathbf{f} \right) &= \mathbf{f}^T \frac{\partial (\mathbf{\Lambda})}{\partial w_i} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{\Lambda}^T \mathbf{f} + \mathbf{f}^T \mathbf{\Lambda} \frac{\partial (\mathbf{W})}{\partial w_i} \mathbf{D}_e^{-1} \mathbf{\Lambda}^T \mathbf{f} \\ &\quad + \mathbf{f}^T \mathbf{\Lambda} \mathbf{W} \mathbf{D}_e^{-1} \frac{\partial \left(\mathbf{\Lambda}^T \right)}{\partial w_i} \mathbf{f} \\ &= \mathbf{f}^T D_e^{-1}(i,i) \mathbf{\Lambda}_i \mathbf{\Lambda}_i{}^T \mathbf{f} - \mathbf{f}^T \mathbf{\Xi}_i \mathbf{f}, \end{aligned} \tag{5.10}$$

where $\mathbf{\Lambda}_i \in \mathbf{R}^{|V|}$ is the i-th column of $\mathbf{\Lambda}$. $\mathbf{\Xi}_i = \operatorname{diag}(\mathbf{\Lambda}_i) \mathbf{D}_u^{-1/2} \mathbf{A}$. Observe that $\mathbf{\Xi}_i$ is a symmetric matrix and $\operatorname{diag}(\mathbf{\Lambda}_i)$ is a $|V| \times |V|$ diagonal matrix having $\mathbf{\Lambda}_i$ in its main diagonal. By substituting (5.10) into (5.9) and (5.8), we obtain the following closed expressions:

$$c = \frac{1}{n} \left\{ \mathbf{f}^T \sum_{i=1}^{n} \left[D_e^{-1}(i,i) \mathbf{\Lambda}_i \mathbf{\Lambda}_i{}^T - \mathbf{\Xi}_i \right] \mathbf{f} - 2\kappa \right\}$$

$$w_i = \frac{1}{2\kappa} \left\{ \mathbf{f}^T \left[D_e^{-1}(i,i) \mathbf{\Lambda}_i \mathbf{\Lambda}_i{}^T - \mathbf{\Xi}_i \right] \mathbf{f} - c \right\}. \tag{5.11}$$

Having optimized the hyperedge weights, $\mathbf{f}$ is re-optimized. The proposed method is summarized in Algorithm 3.

Algorithm 3 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 (5.3).

4 Optimize Eq.(5.5) to find the hyperedge weights as in (5.11).

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 $\mathbf{f}$.

5.4 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*[1], 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 5.1.

[1] `http://www.flickr.com/services/api`

Table 5.1: 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

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 [59].

Having computed pairwise distances according to the "Haversine formula"[2], geo-tags were clustered into 125 distinct clusters using hierarchical clustering.

5.5 Hypergraph Construction

The hypergraph structure is displayed in Table 5.2. 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.

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

Table 5.2: 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

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 [39] and they were reduced to the 5 most similar images to the reference image, by using scale-invariant feature transform (SIFT) [48]. 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.

5.6 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 (4.20). 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 (4.21). 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 (5.3) as Image Tagging on Hypergraph (ITH). The ranking obtained by the approach in [56] 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. 5.3, 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.5.2, 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.5.3, image tagging was performed by initializing the hyperedge weights unequally, as in [26]. 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.

In Table 5.3, 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 [26]. In all experiments, by employing the ITH-HWE the results are significantly improved, validating the merits of the proposed hyperedge weight learning scheme.

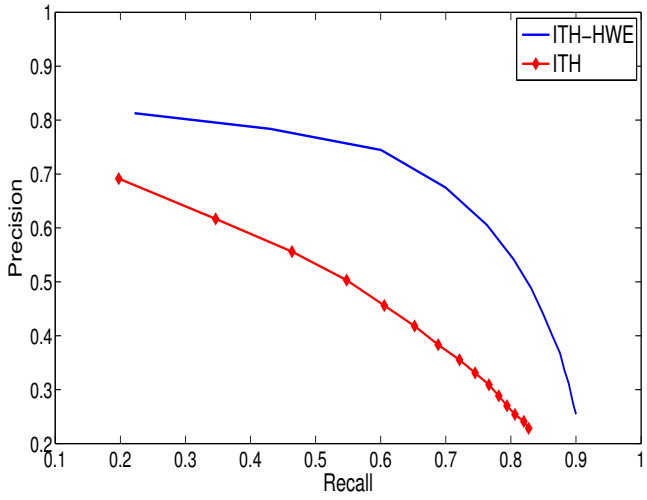

Figure 5.2: Averaged Recall-Precision curves for the ITH and ITH-HWE.

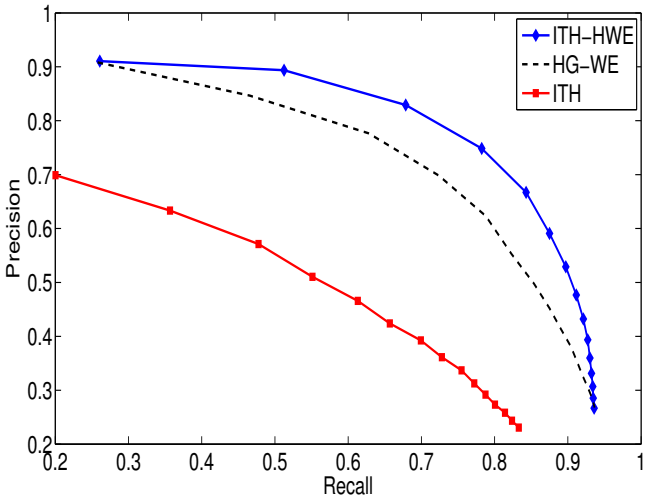

Figure 5.3: Averaged Recall-Precision curves for the compared methods.

5.7 Conclusion and Future Work

In this chapter, an efficient hyperedge weight learning scheme has been detailed. 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 $O(|E|^3)$

Table 5.3: 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

complexity of the method, could be a topic of future research.

Chapter 6

A Hyperedge Weight Updating Scheme Using Descent Gradient and Multiple Constraints

6.1 Introduction

Here, the optimization problem of the hyperedge weight updating scheme, presented in the previous chapter, is further elaborated. A novel iterative optimazation algorithm is presented in this chapter and the disclosed experimental results demonstrate the efficacy of the method.

The new optimization method is based on descent gradient algorithm. The new approach can be further improved by adding different constraints and by adjusting the various parameters of the problem.

6.2 General Solution with Descent Gradient

The descent gradient method can be employed to many optimization problems. By initializing randomly the parameters of the problem (initial "guess") the algorithm aims at finding a minimum in an iterative manner. This iterative optimization is achieved by taking steps in the negative direction of the function gradient.

As presented in Chapter 5, the cost function that needs to be minimized in the hyperedge

weight updating scheme is the (5.4):

$$\underset{\mathbf{f},\mathbf{w}}{\operatorname{argmin}} \left\{ Q(\mathbf{f}) + \kappa ||\mathbf{w}||^2 \right\} \qquad \text{s.t. } \mathbf{1}_n^T \mathbf{w} = 1. \tag{6.1}$$

An alternating minimization of (5.4) starts with a fixed $\mathbf{w}$ and optimizes $\mathbf{f}$ as in (5.3). Next, $\mathbf{f}$ is fixed and $\mathbf{w}$ is optimized. Having fixed $\mathbf{f}$, the optimization w.r.t. $\mathbf{w}$ is read as:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \left\{ \mathbf{f}^T \mathbf{L} \mathbf{f} + \kappa ||\mathbf{w}||^2 \right\} \qquad \text{s.t. } \mathbf{1}_n^T \mathbf{w} = 1. \tag{6.2}$$

Next the Lagrangian function of the aforementioned optimization problem is employed yielding (5.11). However, an alternative solution can be achieved by using descent gradient solving this way the optimization problem in an iterative manner. More precisely, the derivative is:

$$\frac{\partial \Psi}{\partial w_i} = \frac{\partial}{\partial w_i} \left(-\mathbf{f}^T \mathbf{\Lambda} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{\Lambda}^T \mathbf{f} \right) + 2\kappa w_i + c \tag{6.3}$$

where $\mathbf{\Lambda} = \mathbf{D}_u^{-1/2} \mathbf{H}$ and the weight of each hyperedge i is estimated as:

$$w_i(iter + 1) = w_i(iter) - \mu \frac{\partial \Psi}{\partial w_i} \tag{6.4}$$

where $iter$ is the number of the iteration.

The optimization problem presented above has only one equality constraint. By taking this problem one step further, the equality constraint of the aforementioned optimization problem can be complemented by other inequality constraints in order to improve the results.

6.3 Hyperedge Weight Updating with Multiple Constraints

Given a differential scalar field $\Psi(\mathbf{w})$ and an initial w_0, gradient descent iteratively moves the guess toward lower values of Ψ by taking steps in the direction of the negative gradient. As an initialization step all the weights w_i are set to be fuzzy numbers such as $0 \leq w_i \leq 1$ and $1 \leq i \leq n$, where $n = |E|$. Then a search is performed for the active constraints. From the initial statement of the problem it can be deducted that the equality constraint is always included as first constraint and it is always active. Let G_1 be the equality constraint as follows:

$$G_1 : \sum_{i=1}^{n} w_i - 1 = 0 \tag{6.5}$$

For $i > 1$, G_i is an active constraint if $w_{i-1} = 0$ or $w_{i-1} = 1$. Let ν_1, ν_2,$\cdots$,ν_ℓ be ℓ active constraints. Since $\nu_1 = 1$, for $j > 1$, $2 \leq \nu_j \leq n + 1$, where ν_j is a pointer to a hyperedge weight such that $\nu_j - 1 \in [1, \cdots, n]$, $j > 1$. Therefore, it is supposed that:

$$Q = P + \sum_{j=1}^{\ell} c_j G_{\nu_j} \tag{6.6}$$

where c_j, $j = 1, 2, \cdots, \ell$ are the Lagrange multipliers and G_{ν_j} is

$$G_{\nu_j} = \begin{cases} \mathbf{1}_n^T \mathbf{w} - 1, & \text{if } \nu_j = 1 \\ \mathbf{w}_{\nu_j - 1}, & \text{otherwise.} \end{cases} \tag{6.7}$$

Based on the Karush Kuhn Tucker (KKT) conditions, it is required that $\nabla Q \perp \nabla G_{\nu_j}$, $j = 1, 2, \cdots, \ell$ yielding:

$$\nabla G_{\nu_j}^T \nabla Q = 0. \tag{6.8}$$

It can be shown that

$$\nabla Q = \nabla P + \sum_{j=1}^{\ell} c_j \nabla G_{\nu_j} = \nabla P + \mathbf{\Gamma c}. \tag{6.9}$$

where c_j are the Lagrange multipliers and ℓ is the variable of the number of the active constraints. Based on the Karush Kuhn Tucker (KKT), $\nabla Q \perp \nabla G_{\nu_j}$, $j = 1, \cdots, \ell$ yielding

$$\nabla {G_{\nu_j}}^T \nabla Q = 0 \Leftrightarrow \mathbf{\Gamma}^T \nabla Q = 0. \tag{6.10}$$

By substituting (6.9) to (6.10), we obtain:

$$\mathbf{c} = -(\mathbf{\Gamma}^T \mathbf{\Gamma})^{-1} \mathbf{\Gamma}^T \nabla P \tag{6.11}$$

Matrix $\mathbf{\Gamma}$ has size $|n| \times |\ell|$. It can be seen that the first column is $\mathbf{1}_{n \times 1}$, while the remaining columns $j > 1$ have 1 at the row $\nu_j - 1$ and zero otherwise. Its j-th column is ∇G_{ν_j}, i.e. $\mathbf{\Gamma} = [\nabla G_1 | \nabla G_{\nu_2} | \cdots | \nabla G_{\nu_\ell}]$. More analytically, the matrices $\mathbf{\Gamma}^T \mathbf{\Gamma}$ and $(\mathbf{\Gamma \Gamma})^{-1}$ have the

following structure:

$$\mathbf{\Gamma}^T\mathbf{\Gamma} = \begin{bmatrix} n & 1 & \cdots & \cdots & \cdots & 1 \\ 1 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & & \ddots & & 0 & \vdots \\ \vdots & 0 & & \ddots & & \vdots \\ 1 & & & & \ddots & 0 \\ 1 & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \tag{6.12}$$

$$(\mathbf{\Gamma}^T\mathbf{\Gamma})^{-1} = \frac{1}{n-l+1}\begin{bmatrix} 1 & -1 & \cdots & \cdots & \cdots & -1 \\ -1 & (n-l+2) & 1 & \cdots & \cdots & 1 \\ \vdots & & \ddots & & 1 & \vdots \\ \vdots & 1 & & \ddots & & \vdots \\ \vdots & & & & 1 & 1 \\ -1 & 1 & \cdots & \cdots & 1 & (n-l+2) \end{bmatrix} \tag{6.13}$$

A closed form expression can be obtained. Let

$$S_{active} = \sum_{i\,=\,active} \nabla P_i \tag{6.14}$$

and

$$S_{inactive} = \sum_{i\,=\,inactive} \nabla P_i. \tag{6.15}$$

We can obtain the Lagrange multipliers $\mathbf{c}$ by the following equation:

$$\begin{aligned}
\mathbf{c} &= -\frac{1}{n-\ell+1}\begin{bmatrix} 1 & -1 & \cdots & -1 \\ -1 & n-\ell+2 & & 1 \\ \vdots & & \ddots & \\ -1 & 1 & & n-\ell+2 \end{bmatrix}\begin{bmatrix} \mathbf{1}^T \\ \nabla G_{\nu_2}^T \\ \nabla G_{\nu_\ell}^T \end{bmatrix}\begin{bmatrix} \nabla P_1 \\ \nabla P_2 \\ \vdots \\ \nabla P_n \end{bmatrix} \\
&= -\frac{1}{n-\ell+1}\begin{bmatrix} 1 & -1 & \cdots & -1 \\ -1 & n-\ell+2 & & 1 \\ \vdots & & \ddots & \\ -1 & 1 & & n-\ell+2 \end{bmatrix}\begin{bmatrix} S_{inactive}+S_{active} \\ \nabla P_{\nu_2} \\ \vdots \\ \nabla P_{\nu_\ell} \end{bmatrix} \\
&= -\frac{1}{n-\ell+1}\begin{bmatrix} S_{inactive}+S_{active}-S_{active} \\ -S_{inactive}-S_{active}+(n-\ell+2)\nabla P_{\nu_2}+(S_{active}-\nabla P_{\nu_2}) \\ -S_{inactive}-S_{active}+(n-\ell+2)\nabla P_{\nu_\ell}+(S_{active}-\nabla P_{\nu_\ell}) \end{bmatrix} \\
&= -\frac{1}{n-\ell+1}\begin{bmatrix} S_{inactive} \\ -S_{inactive}+(n-\ell+1)\nabla P_{\nu_2} \\ \vdots \\ -S_{inactive}+(N-\ell+1)\nabla P_{\nu_\ell} \end{bmatrix} = \begin{bmatrix} \frac{-S_{inactive}}{n-\ell+1} \\ \frac{S_{inactive}}{n-\ell+1}-\nabla P_{\nu_2} \\ \frac{S_{inactive}}{n-\ell+1}-\nabla P_{\nu_\ell} \end{bmatrix} \qquad (6.16)
\end{aligned}$$

If (6.16) is substituted into (6.9) we obtain:

$$\nabla Q = \begin{bmatrix} \nabla P_1 \\ \nabla P_2 \\ \vdots \\ \nabla P_n \end{bmatrix} + \begin{bmatrix} 1 & |\nabla G_{\nu_2}| & \cdots & |\nabla G_{\nu_\ell}| \end{bmatrix}\begin{bmatrix} \frac{S_{inactive}}{n-\ell+1} \\ -\frac{S_{inactive}}{n-\ell+1}-\nabla P_{\nu_2} \\ \vdots \\ -\frac{S_{inactive}}{n-\ell+1}-\nabla P_{\nu_\ell} \end{bmatrix} \qquad (6.17)$$

and finally

$$\nabla Q_i = \begin{cases} 0 & \text{if } w_i = 0 \text{ or } w_i = 1 \\ \nabla P_i - \frac{S_{inactive}}{n-\ell+1}, & \text{otherwise.} \end{cases} \qquad (6.18)$$

For w_i inactive, ∇P_i is obtained as follows:

$$\begin{aligned}\nabla P_i &= \frac{\partial}{\partial w_i}(-\mathbf{f}^T\mathbf{\Lambda}W\mathbf{D}_e^{-1}\mathbf{\Lambda}^T\mathbf{f}) + 2\kappa w_i \\ &= -\mathbf{f}^T D_e^{-1}(i,i)\mathbf{\Lambda}_i{\mathbf{\Lambda}_i}^T\mathbf{f} + \mathbf{f}^T\mathbf{\Xi}_i\mathbf{f} + 2\kappa w_i \\ &= -\mathbf{f}^T(D_e^{-1}(i,i)\mathbf{\Lambda}_i{\mathbf{\Lambda}_i}^T - \mathbf{\Xi}_i)\mathbf{f} + 2\kappa w_i \end{aligned} \tag{6.19}$$

where $\mathbf{\Lambda}_i \in \mathbf{R}^{|V|}$ is the i-th column of $\mathbf{\Lambda}$.

Finally, the hyperedge weights for the inactive constraints are updated as in (6.20), while if w_i becomes 0 or 1 in any iteration, then it becomes frozen to 0 or 1 in the subsequent iterations.

$$\begin{aligned} w_i(iter+1) &= w_i(iter) - \mu\nabla Q_i \\ &= w_i(iter) - \mu\nabla P_i + \mu\frac{S_{inactive}}{n-l+1}, \end{aligned} \tag{6.20}$$

This way, the hyperedge weight updating scheme is efficiently optimized w.r.t. the constraints given. It is very rationale for someone to seek various constraints in order to speed up the convergence procedure or obtain more accurate results. Here, except for the initial equality constraint $\mathbf{1}_n^T\mathbf{w} = 1$, 2 inequality constraints, $0 \leq w_i \leq 1$ were applied smoothing the convergence and treating the hyperedge weights in a probabilistic point of view. The final hyperedge weights are derived as follows:

$$w_i(iter+1) = \begin{cases} 0, & \text{if} \quad w_i(iter) \leqq 0 \\ w_i(iter) - \mu\nabla P_i + \mu\frac{S_{inactive}}{n-l+1}, & \text{otherwise} \\ 1, & \text{if} \quad w_i(iter) \geqq 1 \end{cases} \tag{6.21}$$

6.4 Experiments

The averaged Recall-Precision 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. Let us refer to the ranking obtained by the proposed approach as Image Tagging with Hyperedge Weight Updating (IT-HWU) and that obtained by (5.3) as Image Tagging on Hypergraph (ITH).

The dataset used here for experimental evaluation is the same as described in chapter 5. 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. 6.1, 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. The hyperedge weights were initialized to $\frac{1}{n}$. It is demonstrated that by applying the hyperedge weight iterative updating scheme. The IT-HWU outperforms the ITH significantly, reaffirming the effectiveness of the proposed method.

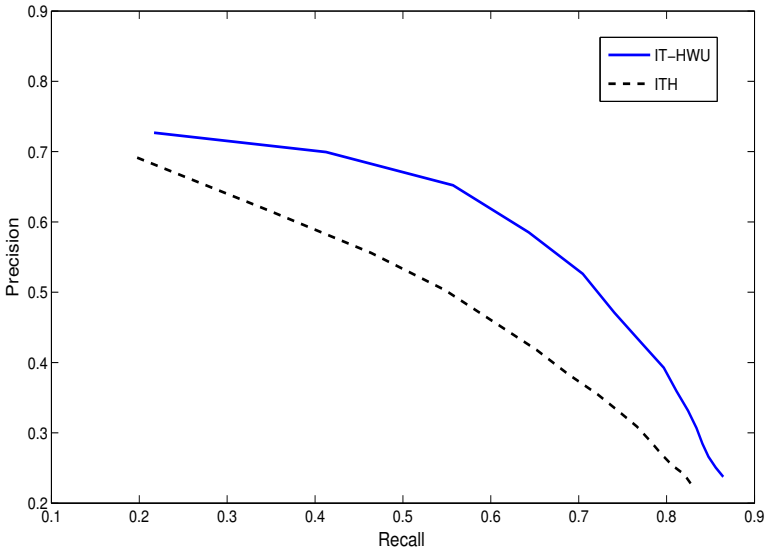

Figure 6.1: Averaged Recall-Precision curves for the ITH and IT-HWU.

Chapter 7

Final Conclusions

7.1 Introduction

Here, this master thesis is concluded providing a broad perspective on the whole work done. Firstly, the main contributions of the scientific research made are highlighted and the experimental results are summarized. Next, potential topics for future research are discussed.

7.2 Conclusions

In this master thesis, the hypergraph utility of modeling high-order relations was exploited and image tagging and recommendation were addressed as hypergraph ranking problems. Hypergraph learning has been used to implement efficient image tagging and recommendation models and their efficacy was improved by several proposed approaches.

First, a novel and efficient annotation and POI recommender system was proposed. A method to organize large collections of images was developed based on clustering and classification. Next, tourism related images were annotated geographically, semantically, and visually, harnessing visual attributes and text information. The heart of the aforementioned methods was the PLSA. PLSA was enhanced by an effective initialization method and used in order to extract semantic information from image metadata. Moreover, thanks to hypergraph learning, a tourism recommender system was implemented. The efficacy of the recommender system was enhanced by applying an optimization approach based on hypergraph ranking and enforcing group sparsity constraints. The solution of the optimization

problem was significantly accelerated via the gradient accelerated method. The experiments were conducted on a dataset of Greek landmarks and the obtained results reaffirmed the efficiency and the accuracy of the proposed method. In particular, an average precision of 92% at 10% recall is reported for semantic image annotation. The accuracy of content-based image classification of 310 test images over 14 classes is $82,6\%$. For tourism recommendation, an average precision of 92% is measured at 1% recall, indicating the effectiveness of the proposed recommendation method. Second, an effective method of image tagging based on hypergraph learning was proposed and an efficient hyperedge weight learning scheme was employed. Thanks to hypergraph learning, image tagging was addressed as a hypergraph ranking problem and improved by employing the proposed hyperedge weight learning scheme. Image tagging addressed within hypergrapgh learning fully exploited the image content, the context, and the social media information. The hyperedge weight learning scheme was analyzed mathematically and presented in full detail in this master thesis. The 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. Finally, the optimization problem of the hyperedge weight updating scheme, was revisited and further analyzed. A novel iterative optimization algorithm was proposed based on descent gradient algorithm. Both equality and inequality constraints were taken into account and the disclosed experimental results demonstrated the potential of the method.

7.3 Future Work

Here, hypergraphs were broadly employed and hypergraph learning proved to be undoubtedly an important research field with great potential. The incremental update of an already trained hypergraph learning model can be a very interesting topic of future research. Here, the merits of the proposed methods were demonstrated experimentally on collections of Greek landmark images. However, further experimental evaluation could also take place in the future using different and larger datasets. A very interesting topic of future research would be the application of the proposed methods to large-scale datasets and the integration with other methods derived from the research field of Big Data. Moreover, the proposed methods were applied mainly to image tagging and recommendation but they can also accommodate tagging or recommendation for any multimedia (e.g. music, video) or even the

fusion between them. Another very interesting topic would be their application to friendship recommendation between members of social networks. Finally, applications, such as travel guideline systems based on images and their geographic coordinates or query-based systems recommending touristic destinations can also benefit from the proposed methods.

Bibliography

[1] M. Gupta, R. Li, Z. Yin, and J. Han, Survey on social tagging techniques, ACM SIGKDD Explorations Newsletter, vol. 12, no. 1, pp. 58–72, 2010.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutierrez, Recommender systems survey, Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.

[3] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, 2005.

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

[5] D. Gunopulos, H. Mannila, R. Khardon, and H. Toivonen, Data mining, hypergraph transversals, and machine learning, in Proc. ACM Int. Symp. SIGACT-SIGMOD-SIGART Principles of Database Systems, pp. 209–216, 1997.

[6] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, Multilevel hypergraph partitioning: applications in VLSI domain, IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 7, no. 1, pp.69–79, 1999.

[7] H. Liu, D. Dou, R. Jin, P. LePendu, and N. Shah, Mining biomedical ontologies and data using RDF hypergraphs, in IEEE Int. Conf. Machine Learning and Applications (ICMLA), vol. 1, pp. 141–146, 2013.

[8] C. Kotropoulos, Multimedia social search based on hypergraph learning, in I. Pitas (Eds), Graph-Based Social Media Analysis, pp. 221–281, 2015.

[9] D. Haussler, and E. Welzl, ε-nets and simplex range queries, Discrete and Computational Geometry, vol. 2, no. 1, pp. 127–151, 1987.

[10] D. Zhou, J. Huang, and B. Schokopf, Learning with hypergraphs: Clustering, classification, and embedding, in Proc. Advanced Neural Information Processing Systems, pp. 1601–1608, 2007

[11] Y. T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T. S. Chua, and H. Neven, Tour the world: building a web-scale landmark recognition engine, in Proc. IEEE Computer Vision and Pattern Recognition, pp. 1085–1092, 2009.

[12] S. Papadopoulos, C. Zigkolis, Y. Kompatsiaris, and A. Vakali, Cluster-based landmark and event detection on tagged photo collections, IEEE Multimedia, pp. 52–63, 2011

[13] L. Yunpeng, D. J. Crandall, and D. P. Huttenlocher, Landmark classification in large-scale image collections, in Proc. Int. Conf. Computer Vision, pp. 1957–1964, 2009.

[14] S. L. Feng, R. Manmatha, and V. Lavrenko, Multiple Bernoulli relevance models for image and video annotation, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 1002–1009, 2004.

[15] F. Monay and D. Gatica-Perez, On image auto-annotation with latent space models, in Proc. ACM Int. Conf. Multimedia, pp. 275–278, 2003.

[16] F. Monay and D. Gatica-Perez, PLSA-based image auto-annotation: constraining the latent space, in Proc. ACM Int. Conf. Multimedia, pp. 348–351, 2004.

[17] T. Hofmann, Probabilistic latent semantic analysis, in Proc. 15th Conf. Uncertainty in Artificial Intelligence, pp. 289–296, 1999.

[18] D. Tian, X. Zhao, and Z. Shi, Refining image annotation by integrating PLSA with random walk model, in Advances in Multimedia Modeling, vol. 7732, pp. 13–23, 2013

[19] C. Wang, D. M. Blei, and L. Fei-Fei, Simultaneous image classification and annotation, in Proc. IEEE Computer Vision and Pattern Recognition, pp. 1903–1910, 2009.

[20] D. M. Blei and J. D. McAuliffe, Supervised topic models, in Proc. Conf. Neural Information Processing Systems, vol. 7, pp. 121–128, 2007.

[21] 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.

[22] 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 Int. ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 540–547, 2009.

[23] J. Tang, R. Hong, S. Yan, T. S. Chua, G. J. Qi, and R. Jain, Image annotation by K-NN-sparse graph-based label propagation over noisily tagged web images, ACM Trans. Intelligent Systems and Technology, vol. 2, no. 2, pp 111–126, 2014.

[24] 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, pp. 391–400, 2010.

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

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

[27] G. H. Golub and C. F. Van Loan, Matrix Computations, Johns Hopkins Studies in Mathematical Sciences, 1996.

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

[29] 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, pp. 779–789, 2012.

[30] 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, pp. 2333–2336, 2012.

[31] L. Cao, J. Luo, and A. Gallagher, A worldwide tourist recommendation system based on geotagged web photos, in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, pp. 2274–2277, 2010.

[32] H. K. Tan, C. W. Ngo and X. Wu, Modeling video hyperlinks with hypergraph for web video reranking, in Proc. ACM Int. Conf. Multimedia, pp. 659–662, 2008.

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

[34] K. Pliakos and C. Kotropoulos, Simultaneous image clustering, classification and annotation for tourism recommendation, in Proc. 8th Hellenic Conf. Artificial Intelligence, pp. 630–640, 2014.

[35] N. Bassiou and C. Kotropoulos, On-line PLSA: Batch updating techniques including out of vocabulary words, IEEE Trans. Neural Networks and Learning Systems, 2014.

[36] N. Bassiou and C. Kotropoulos, RPLSA: A novel updating scheme for probabilistic latent semantic analysis, Computer, Speech, and Language, vol. 25, no. 4, pp. 741–760, 2011.

[37] D. M. Blei, A. Y. Ng, and M. I. Jordan, Latent Dirichlet allocation, Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.

[38] C. J. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery, vol. 2, no.2, pp. 121–167, 1998.

[39] A. Oliva and A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. Journal Computer Vision, vol. 42, no. 3, pp. 145–175, 2001.

[40] J. Friedman, T. Hastie, and R. Tibshirani, A note on the group LASSO and a sparse group LASSO, Arxiv preprint arXiv:1001.0736, 2010.

[41] M. Yuan and Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 68, no. 1, pp. 49–67, 2006.

[42] J. Huang and T. Zhang, The benefit of group sparsity, The Annals of Statistics, vol. 38, no. 4, pp. 1978–2004, 2010

[43] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal Imaging Sciences, vol. 2, no. 1, pp.183–202, 2009

[44] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Machine Learning, vol. 42, no. 12, pp. 177–196, 2001.

[45] A. Dempster, N. Laird, and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm (with discussion), Journal of the Royal Statistical Society, Series B, vol. 39, no. 1, pp. 1–38, 1977.

[46] G. Tzortzis and A. Likas, The global kernel k-means clustering algorithm, IEEE Trans. Neural Networks, vol. 20, no. 7, pp. 1181–1194, 2009.

[47] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and C. Schmid, Evaluation of GIST descriptors for web-scale image search. in Proc. ACM Int. Conf. Image and Video Retrieval, pp. 19–26, 2009.

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

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

[50] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Communications ACM, vol. 18, no. 11, pp. 613–620, 1975.

[51] C. Cortes and V. Vapnik, Support-vector networks, Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[52] J. Bioucas-Dias, M. Figueiredo, and J. Oliveira, Adaptive total-variation image deconvolution: A majorization-minimization approach, in Proc. European Signal Processing Conf., pp. 1–4, 2006.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.