Unified Tag Analysis With Multi-Edge Graph[*]

Dong Liu
School of Computer Sci.& Tec.
Harbin Institute of Technology
Harbin 150001, P. R. China
dongliu.hit@gmail.com

Shuicheng Yan
Department of ECE
National Univ. of Singapore
Singapore 117576, Singapore
eleyans@nus.edu.sg

Yong Rui
Microsoft China R&D Group
49 Zhichun Road
Beijing 100080, P. R. China
yongrui@microsoft.com

Hong-Jiang Zhang
Microsoft Adv. Tech. Center
49 Zhichun Road
Beijing 100080, P. R. China
hjzhang@microsoft.com

ABSTRACT

Image tags have become a key intermediate vehicle to organize, index and search the massive online image repositories. Extensive research has been conducted on different yet related tag analysis tasks, e.g., tag refinement, tag-to-region assignment, and automatic tagging. In this paper, we propose a new concept of *multi-edge graph*, through which a unified solution is derived for the different tag analysis tasks. Specifically, each vertex of the graph is first characterized by a unique image. Then each image is encoded as a *region bag* with multiple image segmentations, and the thresholding of the pairwise similarities between regions naturally constructs the *multiple edges* between each vertex pair. The unified tag analysis is then generally described as the tag propagation between a vertex and its edges, as well as between all edges cross the entire image repository. We develop a core vertex-vs-edge tag equation unique for multi-edge graph to unify the image/vertex tag(s) and region-pair/edge tag(s). Finally, unified tag analysis is formulated as a constrained optimization problem, where the objective function characterizing the cross-patch tag consistency is constrained by the core equations for all vertex pairs, and the cutting plane method is used for efficient optimization. Extensive experiments on various tag analysis tasks over three widely used benchmark datasets validate the effectiveness of our proposed unified solution.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Experimentation, Performance

Keywords

Multi-edge Graph, Tag Refinement, Tag-to-Region Assignment, Automatic Tagging

1. INTRODUCTION

The prevalence of image/video capture devices and growing practice of photo sharing with online websites have resulted in proliferation of image data on the Internet. Some photo sharing websites, such as Flickr [1] and Picasa [2], are becoming part of our daily lives. Take Flickr as example– more than $2,000$ images are being uploaded by grassroot Internet users every minute. Therefore, how to manage, index and search for these images effectively and efficiently is becoming an increasingly important research task.

Existing approaches for Internet image search usually build upon keyword queries against the texts around the images within webpages, such as anchor texts, texts in the bodies of webpages and file names. Beyond simply harnessing the indirect surrounding texts of web images for text query matching, a more desirable technique is to annotate the images with their associated semantic labels, namely tags. The underlying principle is that tags can capture the essential semantic contents of images more precisely, and thus are better for organizing and searching image repositories. Nowadays, those popular photo sharing websites provide the convenient interfaces for users to manually add tags to the photos they browsed, which also attracts a variety of research efforts in analyzing these user-provided tags. The main focuses of previous tag analysis research have been put on three aspects:

Tag Refinement. The purpose of tag refinement is to refine the unreliable user-provided tags associated with the images. Recent studies reported in [3, 4, 5] reveal that the user-provided tags associated with those social images are rather imprecise, with only about 50% precision rate. Moreover, the average number of tags for each social image is relatively small [6], which is far below the number required to fully describe the contents of an image. Therefore, effective methods to refine these unreliable tags have become emerging needs. Content-based Annotation Refinement (CBAR) [7] is one such method, which re-ranks the

[*] This work was performed at Learning and Vision Group of National University of Singapore.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'10, October 25–29, 2010, Firenze, Italy.
Copyright 2010 ACM 978-1-60558-933-6/10/10 ...$10.00.

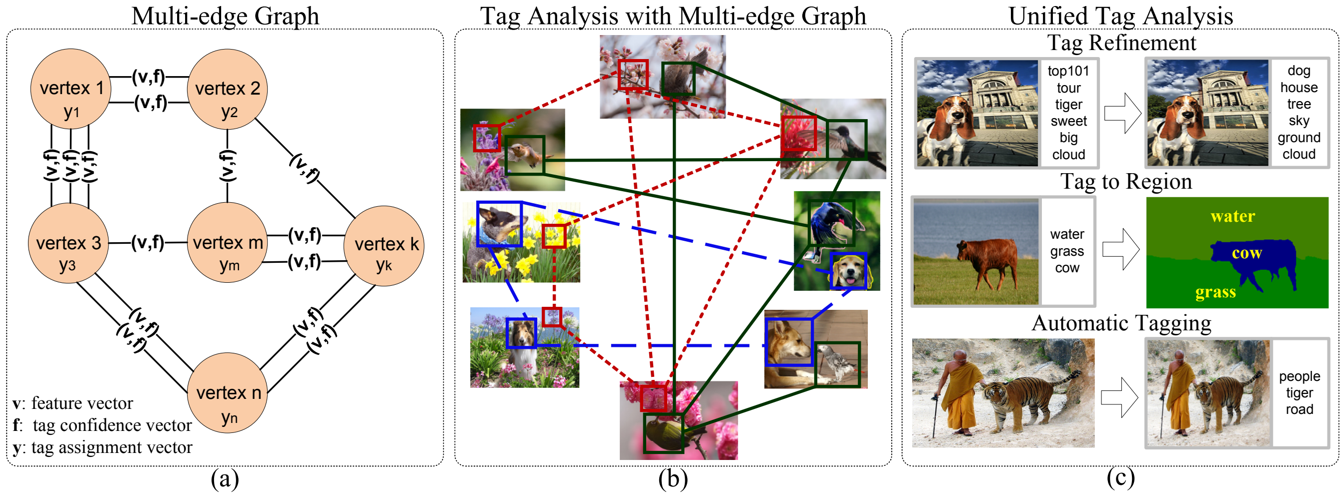

Figure 1: (a) An illustration of *multi-edge graph*, a graph with multiple edges between each vertex pair. (b) For image tag analysis, each edge connects two segmented image regions from two unique images/vertices, and thus each image/vertex pair is generally connected with multiple edges, where the number of edges for each image/vertex pair may be different. (c) Based on multi-edge graph, various image tag analysis tasks can be formulated within a unified framework. For better viewing, please see original color pdf file.

tags of an image and only reserves the top ones as the refined results. However, this method focuses on selecting a coherent subset of tags from existing tags associated with each image, and thus is incapable of "inventing" new tags for a particular image if the tags are not initially provided by users. To address this issue, Liu et al. [8] propose to refine the tags based on the visual and semantic consistency residing in the social images, and assign similar tags to visually similar images. The refined and possibly enriched tags can then better describe the visual contents of the images and in turn improve the performances in tag-related applications.

Tag-to-Region Assignment. Even after we obtain reliable tags annotated at the image-level, we still need a mechanism to assign tags to regions within an image, which is a promising direction for developing reliable and visible content-based image retrieval systems. There exist some related works [9, 10, 11] in computer vision community, known as simultaneous object recognition and image segmentation, which aims to learn explicit detection model for each class/tag, and thus inapplicable for real applications due to the difficulties in collecting precisely labeled image regions for each tag. In multimedia community, a recent work is proposed by Liu et al. [12], which accomplishes the task of tag-to-region assignment by using a bi-layer sparse coding formulation for uncovering how an image or semantic region could be reconstructed from the over-segmented image patches of the entire image repository. In [12], the atomic representation is based on over-segmented patches, which are not descriptive enough and may be ambiguous across tags, and thus the algorithmic performance is limited.

Multi-label Automatic Tagging. Accurate manual tagging is generally laborious when the image repository volume becomes large, while a large portion of images uploaded onto those photo sharing websites are unlabeled. Therefore, an automatic process to predict the tags associated with an image is highly desirable. Generally, popular machine learning techniques are applied to learn the relationship between the image contents and semantic tags based on a collection of manually labeled training images, and then use the learnt models to predict the tags of those unlabeled images. Many algorithms have been proposed for automatic image tagging, varying from building classifiers for individual semantic labels [13, 14] to learning relevance models between images and keywords [15, 16]. However, these algorithms generally rely on sufficient training samples with high quality labels, which are however difficult, if not impossible, to be obtained in many real-world applications. Fortunately, the semi-supervised learning technique can well relieve the above difficulty by leveraging both labeled and unlabeled images in the automatic tagging process. The most frequently applied semi-supervised learning method for automatic image tagging is *graph-based label propagation* [17]. A core component of the algorithms for label propagation on graph is the graph construction. Most existing algorithms simply construct a graph to model the relationship among individual images, and a single edge is connected between two vertices for capturing the image-to-image visual similarity. Nevertheless, using a single edge to model the relationship of two images is inadequate in practice, especially for the real-world images typically associated with multiple tags.

Although many tag related tasks have been exploited, there exist several limitations for these existing algorithms. (1) Most existing algorithms for tag refinement and automatic tagging tasks typically infer the correspondence between the images and their associated tags only at the image level, and utilize the image-to-image visual similarity to refine or predict image tags. However, two images with partial common tags may be considerably different in terms of holistic image features, and thus image level similarity may be inadequate to describe the similarity of the underlying concepts among the images. (2) The only work [12] for tag-to-region assignment task, despite working at the region level, suffers from the possible ambiguities among the smaller-size patches, which are expected to sparsely reconstruct the desired semantic regions.

It is worth noting that the various tasks on tag analysis are essentially closely related, and the common goal is to discover the tags associated with the underlying semantic regions in the images. Therefore, we naturally believe that there exists a unified framework which can accomplish these tasks in a coherent way, which directly motivates our work in this paper. Towards this goal, we propose a new concept of *multi-edge graph*, upon which we derive a unified formulation and solution to various tag analysis tasks. We characterize each vertex in the graph with a unique image, which is encoded as a "bag-of-regions" with multiple segmentations, and the thresholding of the pairwise similarities between the individual regions naturally constructs the *multiple edges* between each two vertices. The foundation of the unified tag analysis framework is *cross-level tag propagation* which propagates and adapts the tags between a vertex and its connected edges, as well as between all edges in the graph structure. A core vertex-vs-edge tag equation unique for multi-edge graph is derived to bridge the image/vertex tags and the region-pair/edge tags. That is, the maximum confidence scores over all the edges between two vertices indicate the shared tags of these two vertices (see Section 2). Based on this core equation, we formulate the unified tag analysis framework as a constrained optimization problem, where the objective characterizing the cross-region tag consistency is constrained by the core equations for all vertex pairs, and the cutting plane method is utilized for efficient optimization. Figure 1 illustrates the unified tag analysis framework based on the proposed multi-edge graph.

The main contributions of this work can be summarized as follows:

- We propose a unified formulation and solution to various tag analysis tasks, upon which, tag refinement, tag-to-region assignment as well as automatic tagging can be implemented in a coherent way.
- We propose to use the multi-edge graph to model the parallel semantic relationships between the images. We also discover a core equation to connect the tags at the image level with the tags at the region level, which naturally realizes the cross-level tag propagation.
- We propose an iterative and convergence provable procedure with the cutting plane method for efficient optimization.

2. GENERAL MULTI-EDGE GRAPH

An object set endowed with pairwise relationships can be naturally illustrated as a graph, in which the vertices represent the objects, and any two vertices that have some kind of relationship are joined together by an edge. However, in many real-world problems, representing the relationship between two complex relational objects with a single edge may cause considerable information loss, especially when the individual objects convey multiple information channels simultaneously. A natural way to remedy the information loss issue in the traditional graph is to link the objects with multiple edges, each of which characterizes the information in certain aspect of the objects, and then use a multi-edge graph to model the complex relationships amongst the objects. Take the friendship mining task on Flickr website as example, the individual users often provide a collection of information channels including personal profiles, uploaded images, posted comments, tags, etc. Typically, the friendship relations do not reside only within a single information channel, but rather are spread across multiple information channels. Therefore, we represent each shared information channel between any two users with an edge, which naturally forms a multi-edge graph structure. Note that the available information channels for the individual users might be different[1], thus the edge number between different user pairs may be different. With this new graph structure, we can see a more comprehensive picture of the relations between the users, and thus better recognize the friendship relations.

Now we give the definition of multi-edge graph. Formally, we have a multi-edge graph $\mathcal{G} = (V, E)$, where $V = \{1, 2, ..., N\}$ denotes the vertex set, E is a family of multi-edge set $\mathcal{E}^{ij}$ of V, where $\mathcal{E}^{ij} = \{e_1^{ij}, ..., e_{|\mathcal{E}^{ij}|}^{ij}\}$ denotes the multiple edges between vertex i and vertex j. Since each edge in the graph represents one shared information channel between two vertices, we may place a feature representation for the edge to indicate the shared information within the specific information channel conveyed by this edge. We assume that the multi-edge graph $\mathcal{G} = (V, E)$ is represented by a symmetric edge weighting matrix $\mathbf{W}$, where each W_{ij} measures the weighting similarity between two edge feature representations.

Obviously both graph vertices and edges may convey the class/tag information for the multi-edge graph, and then the general target of learning over the multi-edge graph becomes to infer the tags of the edges supported by the individual information channels and/or the tags of the unlabeled vertices based on the tags of those labeled vertices. To realize the cross-level tag inference, a core equation unique for multi-edge graph is available for bridging the tags at the vertex level and the tags at the edge level. Given a tag c, assume a scalar $\mathbf{y}_i^c \in \{0, 1\}$[2] is used to illustrate the labeling information of the vertex i, where $\mathbf{y}_i^c = 1$ means that vertex i is labeled as positive with respect to c and $\mathbf{y}_i^c = 0$ otherwise. We also associate a real-value label $\mathbf{f}_t^c \in [0, 1]$ with the edge e_t to capture the probability that the edge e_t is labeled as positive with respect to c under the specific information channel residing on e_t. Then for any two vertices i and j, a core equation can be formulated as follows:

$$\mathbf{y}_i^c \mathbf{y}_j^c = \max_{e_t \in \mathcal{E}^{ij}} \mathbf{f}_t^c. \tag{1}$$

Discussion. Compared with the traditional single edge graph, the proposed multi-edge graph has the following characteristics: (1) each edge is linked at a specific information channel, which naturally describes the vertex relation from a specific aspect; (2) the graph adjacent structure better captures the complex relations between the vertices owing to the multi-edge linkage representation; and (3) the graph weighting is performed over the individual edge representations, which may describe the correlation of the heterogeneous information channels.

There are also some works [18, 19] related to the term of "multi-graph" in literature, but the focus of these works is to combine multiple graphs obtained from different modalities

[1] For example, some users only upload photos onto Flickr but never provide other information, while some users only post comments to others' photos without sharing his/her own photos.

[2] $\mathbf{y}_i^c$ denotes the entry corresponding to tag c in the tag vector $\mathbf{y}_i$, and thus is a *scalar*. Similar statement holds for vector $\mathbf{f}^c$ and its scalar entry $\mathbf{f}_t^c$.

of the same observations to obtain a more comprehensive single edge graph, which, however, is essentially different from the multi-edge graph introduced in this work, where multiple edges are constructed to characterize the parallel relationships from different information channels between the vertices.

The recently proposed hypergraph [20] also attempts to model the complex relationships among the objects. However, the modeling is performed on a single information channel of the objects, which also causes considerable information loss. Instead of modeling the relationship of objects with a single channel, multi-edge graph differs from hypergraph by explicitly constructing the linkages with multiple information channels of the vertices.

3. CONSTRUCT MULTI-EDGE GRAPH FOR UNIFIED TAG ANALYSIS

In this section, we first describe the bag-of-regions representation for the images, and then introduce the multi-edge graph construction procedure for image-based tag analysis.

3.1 Bag-of-Regions Representation

An image usually contains several regions with different semantic meanings. A straightforward solution for image content analysis is to divide an image into several semantic regions and extract features from each region for content analysis. In this work, we use image segmentation to determine the semantic regions within the individual images. If the segmentation is ideal, regions shall correspond to the semantic objects. However, in general, perfect image segmentation, which builds a one-to-one mapping between the generated regions and objects in the image, is still beyond the capabilities of those existing algorithms [21, 22]. Therefore, it is impossible to expect a segmentation algorithm to partition an image into its constituent objects. Inspired by the work in [23], we propose to use multiple segmentations to relieve the limitations of image segmentation. For each image, we compute multiple segmentations by employing different segmentation algorithms. Each of the resulting segmentations is still assumed to be imperfect, and the weakened assumption is that some of the segments from all segmentations are correct. Specifically, we choose to use two popular image segmentation algorithms in this work. One is the graph-based segmentation algorithm in [21], which incrementally merges smaller-sized patches with similar appearances and with small minimum spanning tree weights. The other one is the Normalized Cut algorithm proposed by Shi et al. [22], which performs the segmentation via a spectral graph partitioning formulation. After obtaining the multiple segmentation results, we apply the "bag-of-regions" representation to describe each image, and each segment constitutes one atomic component of one bag. Figure 2 illustrates the bag-of-regions representation of an example input image.

Furthermore, we represent each segmented region with Bag-of-Words (BoW) features. We first apply the Harris-Laplace detector [24] to determine a set of the salient points. Then the Scale Invariant Feature Transform (SIFT) features [25] are computed over the local areas defined by the detected salient points. We perform K-means clustering method to construct the visual vocabulary with 500 visual words. Then each SIFT feature is mapped into an integer (visual word index) between 1 and 500. Finally, each region is represented as a normalized histogram over the visual vocabulary.

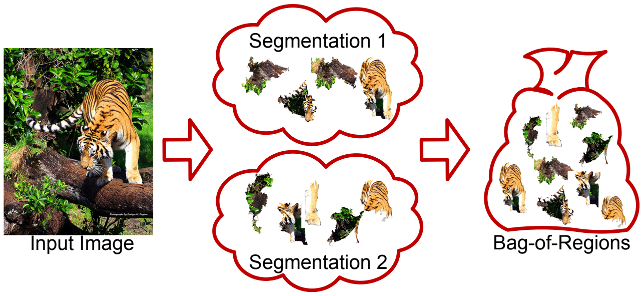

Figure 2: The "bag-of-regions" representation of an input image. We use two segmentation algorithms to segment the input image simultaneously and then collect all derived segments to obtain the bag-of-regions representation.

3.2 Multi-edge Graph Construction

For tag analysis, each vertex of the multi-edge graph characterizes one unique image from the entire image collection. The multi-edge graph construction can be decomposed into two steps: graph adjacency construction and graph weight calculation. For graph adjacency construction, we aim to build up the edge linkages between the regions of different image vertices (each region is considered as one information channel). Given two image vertices i and j, and their corresponding bag-of-regions representations, $\mathbf{x}_i = \{\mathbf{x}_i^1, \mathbf{x}_i^2, ..., \mathbf{x}_i^{|\mathbf{x}_i|}\}$ and $\mathbf{x}_j = \{\mathbf{x}_j^1, \mathbf{x}_j^2, ..., \mathbf{x}_j^{|\mathbf{x}_j|}\}$, our approach is to put one edge if and only if the two regions in the two images are "close enough" (consistent in information channels). In our setting, we use the mutual k-nearest neighbors strategy to discover the closeness relations between the regions. Specifically, two regions $\mathbf{x}_i^m$ and $\mathbf{x}_j^n$ are linked with an edge if $\mathbf{x}_i^m$ is among the k_1 nearest neighbors of $\mathbf{x}_j^n$ and vice versa. Here the k_1 nearest neighbors are measured by the usual Euclidean distance and the parameter k_1 is fixed to be 5. Once the linkages between the regions are created, the adjacency relations between the image vertices can be naturally constructed where any two vertices with at least one edge connection are considered as adjacent in the graph structure. It is worth noting that the adjacent relations defined in this work are symmetric and consequently the constructed adjacency graph is undirected. Then, for each edge e_t linking two regions $\mathbf{x}_i^m$ and $\mathbf{x}_j^n$, we define its feature representation $\mathbf{v}(e_t)$ as the average of $\mathbf{x}_i^m$ and $\mathbf{x}_j^n$, namely, $\mathbf{v}(e_t) = (\mathbf{x}_i^m + \mathbf{x}_j^n)/2$, aiming to convey unbiased information from two regions. For graph weight calculation, based on the feature representations of the edges, we define the similarity between two edges e_t and e_h as follows

$$W_{th} = \begin{cases} e^{-\frac{\|\mathbf{v}(e_t)-\mathbf{v}(e_h)\|^2}{2\sigma^2}}, & \text{if } h \in N_{k_2}(t) \text{ or } t \in N_{k_2}(h), \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $N_{k_2}(\cdot)$ denotes the edge index set for the k_2-nearest neighbors of an edge measured by Euclidean distance and k_2 is set as 100 for all the experiments. σ is the radius parameter of the Gaussian function, and is set as the median value of all pairwise Euclidean distances between the edges.

4. UNIFIED TAG ANALYSIS OVER IMAGE-BASED MULTI-EDGE GRAPH

In this section, we present the unified tag analysis framework over the image-based multi-edge graph. We first describe the problem formulation and then introduce an efficient iterative procedure for the optimization.

4.1 Basic Framework

We restrict our discussion in a transductive setting, where l labeled images $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_l, \mathbf{y}_l)\}$ and u unlabeled images $\{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$ are given[3]. We represent each image $\mathbf{x}_i$ $(i = 1, 2, \ldots, l+u)$ with the "bag-of-regions" representation as described in Section 3.1. All unique tags appeared in the labeled image collection can be collected into a set $\mathcal{Y} = \{y_i, y_2, \ldots, y_m\}$, where m denotes the total number of tags. For any labeled image $(\mathbf{x}_i, \mathbf{y}_i)$ $(i = 1, 2, \ldots, l)$, the tag assignment vector $\mathbf{y}_i$ is a binary vector $\mathbf{y}_i \in \{0, 1\}^m$, where the k-th entry $\mathbf{y}_i^k = 1$ indicates that $\mathbf{x}_i$ is assigned with tag $y_k \in \mathcal{Y}$ and $\mathbf{y}_i^k = 0$ otherwise. Besides, a multi-edge graph $\mathcal{G} = (V, E)$ with vertices V corresponding to the $l + u$ images, where vertices $L = \{1, \ldots, l\}$ correspond to the labeled images, and vertices $U = \{l+1, \ldots, l+u\}$ correspond to the unlabeled images, is also provided. Furthermore, we collect all edges in the graph and re-index the edges in E as $E = \{e_1, e_2, \ldots, e_T\}$, where T is the total number of edges in the graph.

The general goal of tag analysis is to infer globally consistent edge-specific tag confidence vector $\mathbf{f}_t$ for each edge e_t $(t = 1, \ldots, T)$, wherein the c-th entry $\mathbf{f}_t^c$ denotes the probability of assigning tag y_c to the edge e_t. This is essentially to infer the *fine* tag information at the region level from the *coarse* tag information at the image level. Intuitively, the side information conveyed by a set of images annotated with the same tag could be sufficient for disclosing which regions should be simultaneously labeled with the tag under consideration. This is due to the fact that any two images with the same tag will be linked at least by one edge that connects two regions corresponding to the concerned tag in the multi-edge graph. The repetition of such pairwise connections in a fraction of the labeled images will be an important signal that can be used to infer a common "visual prototype" throughout the image collection. Finding multiple visual prototypes and their correspondences with respect to the individual tags is the foundation of the unified tag analysis framework. More importantly, the core equation in Eq. (1) plays an important role in linking the tags of the edges and the tags of the images, and offers the capability for propagating tags between them.

Furthermore, by assuming that similar edges share similar regional features, the distribution of edge-specific tag confidence vector should be *smooth* over the graph. Given a weighting matrix $\mathbf{W}$ which describes the pairwise edge relationship based on Eq. (2), we can propagate the edge-specific tag confidence vectors from the labeled image pairs to the unlabeled ones by enforcing the smoothness of these edge-specific tag confidence vectors over the graph during the propagation. Here, we refer to this approach as *Cross-level Tag Propagation* due to the extra constraints from the core vertex-vs-edge tag equations for all vertex pairs.

We formulate the cross-level tag propagation within a regularized framework which conducts the propagation implicitly by optimizing the regularized objective function,

$$\begin{aligned} \min_{\mathbf{F}} \quad & \Omega(\mathbf{F}, \mathbf{W}) + \lambda \sum_{c=1}^{m} \sum_{i,j \in L^c} \ell(\mathbf{y}_i^c \mathbf{y}_j^c, \max_{e \in \mathcal{E}^{ij}} \mathbf{I}_c^\top \mathbf{f}_{k(e)}) \\ s.t. \quad & \mathbf{f}_t \geq \mathbf{0}, \quad \sum_{c=1}^{m} \mathbf{f}_t^c = 1, \quad t = 1, 2, \ldots, T, \end{aligned} \tag{3}$$

where $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_T]^\top$ consists of the edge-specific tag confidence vectors to be learned for T edges in the graph, L^c denotes the collection of labeled images annotated with a specific tag y_c, $k(e)$ denotes the index of edge e in E. $\ell(\cdot, \cdot)$ is a convex loss function such as hinge loss, and $\mathbf{I}_c$ is an m-dimensional binary vector with the c-th entry being 1 and other entries all being 0's. Ω is a regularization term responsible for the implicit tag propagation, and λ is a regularization parameter. Since the numerical values residing in $\mathbf{F}$ are edge-specific tag confidence scores over the tags, we enforce $\mathbf{f}_t \geq \mathbf{0}$ to ensure that the scores are non-negative. Furthermore, the $\sum_{c=1}^{m} \mathbf{f}_t^c = 1$ constraint is imposed to ensure that the sum of the probabilities to be 1, which essentially couples different tags together and makes them interact with each other in a collaborative way.

We enforce the tag smoothness over the multi-edge graph by minimizing the regularization term Ω:

$$\Omega(\mathbf{F}, \mathbf{W}) = \sum_{t,h=1}^{T} \mathbf{S}_{th} \|\mathbf{f}_t - \mathbf{f}_h\|^2 = 2tr(\mathbf{F}^\top \mathbf{L} \mathbf{F}). \tag{4}$$

Here $\mathbf{S} = \mathbf{Q}^{-\frac{1}{2}} \mathbf{W} \mathbf{Q}^{-\frac{1}{2}}$ is a normalized weight matrix of $\mathbf{W}$. $\mathbf{Q}$ is a diagonal matrix whose (i,i)-entry is the i-th row sum of $\mathbf{W}$. $\mathbf{L} = (\mathbf{I} - \mathbf{S})$ is the graph Laplacian. Obviously, the minimization of Eq. (4) yields a smooth propagation of the edge-specific tag confidence scores over the graph, and nearby edges shall have similar tag confidence vectors.

Based on the pairwise side information provided by images annotated with the same tag, we instantiate the loss function $\ell(\cdot, \cdot)$ in Eq. (3) with ℓ_1-loss:

$$\ell(\mathbf{y}_i^c \mathbf{y}_j^c, \max_{e \in \mathcal{E}^{ij}} \mathbf{I}_c^\top \mathbf{f}_{k(e)}) = |\mathbf{y}_i^c \mathbf{y}_j^c - \max_{e \in \mathcal{E}^{ij}} \mathbf{I}_c^\top \mathbf{f}_{k(e)}|. \tag{5}$$

By introducing slack variables and plugging Eq. (4) and Eq. (5) into Eq. (3), we obtain the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{F}, \xi_{cij}} \quad & 2tr(\mathbf{F}^\top \mathbf{L} \mathbf{F}) + \lambda \sum_{c=1}^{m} \sum_{i,j \in L^c} \xi_{cij} \\ s.t. \quad & -\xi_{cij} \leq \mathbf{y}_i^c \mathbf{y}_j^c - \max_{e \in \mathcal{E}^{ij}} \mathbf{I}_c^\top \mathbf{f}_{k(e)} \leq \xi_{cij}, \quad \xi_{cij} \geq 0, \\ & \mathbf{f}_t \geq \mathbf{0}, \quad \mathbf{1}^\top \mathbf{f}_t = 1, \quad t = 1, 2, \ldots, T. \end{aligned} \tag{6}$$

[3] Note that the proposed framework is general and may not need the support of unlabeled data. For example, in tag refinement and tag-to-region assignment tasks, we can only use the labeled images to obtain the desired results.

This is also equivalent to

$$\begin{aligned} \min_{\mathbf{F},\xi_{cij}} \quad & 2tr(\mathbf{F}^\top\mathbf{L}\mathbf{F}) + \lambda\sum_{c=1}^{m}\sum_{i,j\in L^c}\xi_{cij} \\ s.t. \quad & \mathbf{I}_c^\top \mathbf{f}_{k(e)} - \xi_{cij} \le \mathbf{y}_i^c\mathbf{y}_j^c, \quad e\in\mathcal{E}^{ij} \\ & \mathbf{y}_i^c\mathbf{y}_j^c - \max_{e\in\mathcal{E}^{ij}} \mathbf{I}_c^\top\mathbf{f}_{k(e)} \le \xi_{cij}, \\ & \xi_{cij}\ge 0, \quad c=1,2,\ldots,m, \quad i,j\in L^c, \\ & \mathbf{f}_t \ge \mathbf{0}, \quad \mathbf{1}^\top\mathbf{f}_t = 1, \quad t=1,2,\ldots,T. \end{aligned} \tag{7}$$

4.2 Optimization with Cutting Plane

The optimization problem in Eq. (7) is convex, resulting in a globally optimal solution. However, optimizing Eq. (7) with respect to all $\mathbf{f}_t$'s is of great computational challenge. Instead of solving it directly, we employ the alternating optimization method. The main idea is to sequentially solve a batch of edge-specific tag confidence vectors residing on the edges between two given vertices at each time by fixing other vectors on the remaining edges. We repeat this procedure until Eq. (7) converges or a maximum number of iterations is reached.

In each iteration, solving a subset of edge-specific tag confidence vectors $\mathbf{F}^{ij} = [\mathbf{f}_{k(e_1^{ij})}, \mathbf{f}_{k(e_2^{ij})}, \ldots, \mathbf{f}_{k(e^{ij}_{|\mathcal{E}^{ij}|})}]$ that correspond to the edges between two given vertices i and j yields a standard Quadratic Programming (QP) problem:

$$\begin{aligned} \min_{\mathbf{F}^{ij},\xi_c} \quad & \Omega(\mathbf{F}^{ij},\mathbf{W}) + \lambda\sum_{c=1}^{m}\xi_c \\ s.t. \quad & \mathbf{I}_c^\top\mathbf{f}_{k(e)} - \xi_c \le \mathbf{y}_i^c\mathbf{y}_j^c, \quad e\in\mathcal{E}^{ij}, \\ & \mathbf{y}_i^c\mathbf{y}_j^c - \max_{e\in\mathcal{E}^{ij}}\mathbf{I}_c^\top\mathbf{f}_{k(e)} \le \xi_c, \\ & \xi_c \ge 0, \quad c=1,2,\ldots,m, \\ & \mathbf{f}_{k(e)} \ge \mathbf{0}, \quad \mathbf{1}^\top\mathbf{f}_{k(e)} = 1, \quad e\in\mathcal{E}^{ij}. \end{aligned} \tag{8}$$

Since we only solve the $\mathbf{f}_d$'s residing on edges between vertex i and vertex j and treat the others as constant vectors during the optimization, the smooth term that relates to the concerned optimization variable can be written as

$$\begin{aligned} \Omega(\mathbf{F}^{ij},\mathbf{W}) = 2\sum_{e_t\in\mathcal{E}^{ij}}\sum_{e_h\notin\mathcal{E}^{ij}} S_{t^\star h^\star}(\mathbf{f}_{t^\star}^\top\mathbf{f}_{t^\star} - 2\mathbf{f}_{t^\star}^\top\mathbf{f}_{h^\star}) \\ +2\sum_{e_t\in\mathcal{E}^{ij}}\sum_{e_h\in\mathcal{E}^{ij}\setminus\{e_t\}} S_{t^\star h^\star}(\mathbf{f}_{t^\star}^\top\mathbf{f}_{t^\star} - 2\mathbf{f}_{t^\star}^\top\mathbf{f}_{h^\star} + \mathbf{f}_{h^\star}^\top\mathbf{f}_{h^\star}), \end{aligned} \tag{9}$$

where $t^\star = k(e_t)$ and $h^\star = k(e_h)$ denote the indices of edge e_t and edge e_h in the edge set E, respectively.

We vectorize the matrix $\mathbf{F}^{ij}$ by stacking their columns into a vector. Denote by $\mathbf{r} = vec(\mathbf{F}^{ij})$, then $\mathbf{f}_{k(e_t)}$ can be calculated as $\mathbf{f}_{k(e_t)} = \mathbf{A}_t\mathbf{r}$, where $\mathbf{A}_t = [\mathbf{I}_{(t-1)*m+1}, \ldots, \mathbf{I}_{t*m}]^\top$ with $\mathbf{I}_g$ an $(m\times|\mathcal{E}^{ij}|)$-dimensional column vector in which the g-th entry is 1 while the other entries are all 0's. Thus the objective function can be rewritten as

$$\begin{aligned} \min_{\mathbf{r},\xi_c} \quad & \Omega(\mathbf{r},\mathbf{W}) + \lambda\sum_{c=1}^{m}\xi_c \\ s.t. \quad & \mathbf{I}_c^\top\mathbf{A}_t\mathbf{r} - \xi_c \le \mathbf{y}_i^c\mathbf{y}_j^c, \quad e_t\in\mathcal{E}^{ij}, \\ & \mathbf{y}_i^c\mathbf{y}_j^c - \max_{e_t\in\mathcal{E}^{ij}}\mathbf{I}_c^\top\mathbf{A}_t\mathbf{r} \le \xi_c, \\ & \xi_c \ge 0, \quad c=1,2,\ldots,m, \\ & \mathbf{r}\ge\mathbf{0}, \quad \mathbf{1}^\top\mathbf{A}_t\mathbf{r} = 1, \quad e_t\in\mathcal{E}^{ij}. \end{aligned} \tag{10}$$

Although the objective function in Eq. (10) is quadratic and the constraints are all convex, the second constraint is not smooth. Hence, we cannot directly use the off-the-shelf convex optimization toolbox to solve the problem. Fortunately, the *cutting plane method* [26] is able to well solve the optimization problem with non-smooth constraints. In Eq. (10), we have m slack variables ξ_c $(c = 1, 2, ..., m)$. To solve it efficiently, we first drive the 1-slack form of Eq. (10). More specifically, we introduce one single slack variable $\xi = \sum_{c=1}^m \xi_c$. Finally, we can obtain the objective function as

$$\begin{aligned} \min_{\mathbf{r},\xi} \quad & \Omega(\mathbf{r},\mathbf{W}) + \lambda\xi \\ s.t. \quad & \xi \ge \sum_{c=1}^{m}\max\left(0, \mathbf{I}_c^\top\mathbf{A}_t\mathbf{r} - \mathbf{y}_i^c\mathbf{y}_j^c, \mathbf{y}_i^c\mathbf{y}_j^c - \max_{e_t\in\mathcal{E}^{ij}}\mathbf{I}_c^\top\mathbf{A}_t\mathbf{r}\right) \\ & \mathbf{r}\ge\mathbf{0}, \quad \mathbf{1}^\top\mathbf{A}_t\mathbf{r} = 1, \quad e_t\in\mathcal{E}^{ij}. \end{aligned} \tag{11}$$

Now the problem becomes how to solve Eq. (11) efficiently, which is convex and has non-smooth constraints. In cutting plane terminology, the original problem in Eq. (11) is usually called the *master problem*. The variable ξ is regarded as the nonlinear function of $\mathbf{r}$, i.e., $\xi(\mathbf{r}) = \sum_{c=1}^m \max(0, \mathbf{I}_c^\top\mathbf{A}_t\mathbf{r} - \mathbf{y}_i^c\mathbf{y}_j^c, \mathbf{y}_i^c\mathbf{y}_j^c - \max_{e_t\in\mathcal{E}^{ij}}\mathbf{I}_c^\top\mathbf{A}_t\mathbf{r})$. The basic idea of cutting plane method is to substitute the nonlinear constraints with a collection of linear constraints. The algorithm starts with the following optimization problem:

$$\begin{aligned} \min_{\mathbf{r}} \quad & \Omega(\mathbf{r},\mathbf{W}) \\ s.t. \quad & \mathbf{r}\ge\mathbf{0}, \quad \mathbf{1}^\top\mathbf{A}_t\mathbf{r} = 1, \quad e_t\in\mathcal{E}^{ij}. \end{aligned} \tag{12}$$

After obtaining the solution $\mathbf{r}^s$ $(s = 0)$ of the above problem, we approximate the nonlinear function $\xi(\mathbf{r})$ around $\mathbf{r}^s$ by a linear inequality:

$$\xi(\mathbf{r}) \ge \xi(\mathbf{r}^s) + \frac{\partial\xi(\mathbf{r})}{\partial\mathbf{r}}|_{\mathbf{r}=\mathbf{r}^s}(\mathbf{r}-\mathbf{r}^s), \tag{13}$$

Denote by $g_1(\mathbf{r}) = 0$, $g_2(\mathbf{r}) = \mathbf{I}_c^\top\mathbf{A}_t\mathbf{r} - \mathbf{y}_i^c\mathbf{y}_j^c$, $g_3(\mathbf{r}) = \mathbf{y}_i^c\mathbf{y}_j^c - \max_{e_t\in\mathcal{E}^{ij}}\mathbf{I}_c^\top\mathbf{A}_t\mathbf{r}$, the subgradient of $\xi(\mathbf{r})$ in Eq. (13) can be calculated as follows:

$$\frac{\partial\xi(\mathbf{r})}{\partial\mathbf{r}}|_{\mathbf{r}=\mathbf{r}^s} = \sum_{c=1}^{m}\left(z_p(\mathbf{r})\times\frac{\partial g_p(\mathbf{r})}{\partial\mathbf{r}}\right)|_{\mathbf{r}=\mathbf{r}^s}. \tag{14}$$

Here,

$$z_p(\mathbf{r}) = \begin{cases} 1, & \text{if} \quad p = \arg\max_{p=1,2,3} g_p(\mathbf{r}), \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

and $\partial g_1(\mathbf{r})/\partial\mathbf{r} = 0$, $\partial g_2(\mathbf{r})/\partial\mathbf{r} = \mathbf{I}_c^\top\mathbf{A}_t$ and $\partial g_3(\mathbf{r})/\partial\mathbf{r} = -\sum_{e_t\in\mathcal{E}^{ij}}\gamma_{e_t}\mathbf{I}_c^\top\mathbf{A}_t$, where γ_{e_t} is defined as follows:

$$\gamma_{e_t} = \begin{cases} 1, & \text{if} \quad e_t = \arg\max_{e_t\in\mathcal{E}^{ij}}\mathbf{I}_c^\top\mathbf{A}_t\mathbf{r}, \\ 0, & \text{otherwise}. \end{cases} \tag{16}$$

The linear constraint in Eq. (13) is called a *cutting plane* in the cutting plane terminology. We add the obtained constraint into the constraint set and then use efficient QP package to solve the reduced cutting plane problem. The optimization procedure is repeated until no salient gain when adding more cutting plane constraints. In practice, the cutting plane procedure halts when satisfying the ε-optimality condition, i.e., the difference between the objective value of

Algorithm 1 Unified Tag Analysis Over Multi-edge Graph

```
1: Input: A multi-edge graph $\mathcal{G} = (V, E)$ of an image collection $\mathcal{X}$, with the labeled subset $\mathcal{X}_l$ and the unlabeled subset $\mathcal{X}_u$, edge affinity matrix $\mathbf{W} = \{w_{ij}\}$ $(i, j = 1, 2, .., T)$ of the edge set $E$, and regularization constant $\lambda$.
2: Output: The edge-specific tag confidence vectors $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_T]^\top$.
3: Initialize $\mathbf{F} = \mathbf{F}^0, t = 0, \Delta J = 10^{-3}, J^{-1} = 10^{-3}$.
4: while $\Delta J / J^{t-1} > 0.01$ do
5:     for any pairwise labeled vertices $i$ and $j$ do
6:         Derive problem in Eq. (11), set the constraints set $\Omega = \phi$ and $s = -1$;
7:         Cutting plane iterations:
8:         repeat
9:             s=s+1;
10:            Get $(\mathbf{r}^s, \xi^s)$ by solving Eq. (11) under $\Omega$;
11:            Compute the linear constraint by Eq. (13) and update the constraint set $\Omega$ by adding the obtained linear constraint into it.
12:        until Convergence
13:    end for
14:    for any pairwise vertices $i$ and $j$ in which at least one is unlabeled do
15:        Solve the QP problem whose form is similar to Eq. (12).
16:    end for
17:    $t = t + 1$; $\mathbf{F}^t = \mathbf{F}^{t-1}$; $\Delta J = J^{t-1} - J^t$;
18: end while
```

the master problem and the objective value of the reduced cutting plane problem is smaller than a threshold ε. In this work, we set ε to be 0.1. We use J^t to denote the value of the objective function in Eq. (7), then the whole optimization procedure can be summarized as in Algorithm 1.

4.3 Unified Tag Analysis with the Derived $\mathbf{F}$

Once the tag confidence vectors $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_T]^\top$ are obtained, the predicted tag vector $\mathbf{y}^*$ for a given semantic region $\mathbf{x}$ can be obtained in a majority voting strategy. Assume there are totally p edges connected with the region $\mathbf{x}$ in the multi-edge graph, which form an edge subset $\mathcal{E}(\mathbf{x}) = \{e_1^{\mathbf{x}}, e_2^{\mathbf{x}}, \ldots, e_p^{\mathbf{x}}\}$. We collect the corresponding confidence vectors into a subset $\mathbf{F}(\mathbf{x}) = [\mathbf{f}_{k(e_1^{\mathbf{x}})}, \mathbf{f}_{k(e_2^{\mathbf{x}})}, \ldots, \mathbf{f}_{k(e_p^{\mathbf{x}})}]$, where $k(e_i^{\mathbf{x}})$ denotes the index of edge $e_i^{\mathbf{x}}$ in the whole edge set of the multi-edge graph. In essence, each confidence vector in $\mathbf{F}(\mathbf{x})$ represents the tag prediction result of region $\mathbf{x}$, where the c-th entry corresponds to the confidence score of assigning tag y_c to region $\mathbf{x}$. Since each region usually only contains one semantic concept, we select the tag with the maximum confidence score in the confidence vector as the predicted tag of the given region. Furthermore, since there are p edges simultaneously connected with region $\mathbf{x}$, we predict the tag with each confidence vector and select the tag with the most prediction number as the final predicted tag of region $\mathbf{x}$. By doing so, we can obtain a binary tag assignment vector for each region, and then the three tag analysis tasks can be implemented in the following manner:

Tag Refinement. For each image with user-provided tags, we predict the binary tag assignment vector for each of its composite regions in the bag-of-regions representation and then fuse all binary vectors through the logic OR operator. The obtained vector then indicates the refined tags for the given image.

Tag-to-Region Assignment. The tag-to-region assignment is implemented in a pixel-wise manner. Since our approach relies on two segmentation algorithms, one pixel will appear twice in two regions. For each region, we predict its tag and count the corresponding predicted number with the tag prediction method discussed above. If the tag predictions for the two regions are inconsistent, we select the tag with the most prediction number as the predicted tag of the concerned pixel.

Automatic Tagging. We predict the binary tag assignment vector for each region of an unlabeled image, and then fuse the obtained binary vectors via OR operator.

4.4 Algorithmic Analysis

We first analyze the time complexity of Algorithm 1. In the optimization, each sub-problem is solved with the cutting plane method. As for the time complexity of the cutting plane iteration, we have the following two theorems:

THEOREM 1. *Each iteration in steps* 9-11 *of Algorithm 1 takes time $O(dn)$ for a constant constraint set Ω, where d is the number of edges between vertex i and vertex j.*

THEOREM 2. *The cutting plane iteration in steps* 8-12 *of Algorithm 1 terminates after at most* $\max\{\frac{2}{\varepsilon}, \frac{8\lambda R^2}{\varepsilon^2}\}$ *steps, where*

$$R^2 = \sum_{c=1}^{m} \max_{e_t \in \mathcal{E}^{ij}} (0, \|\boldsymbol{I}_c^\top \boldsymbol{A}_t\|, \|\sum_{e_t \in \mathcal{E}^{ij}} \gamma_{e_t} \boldsymbol{I}_c^\top \boldsymbol{A}_t\|),$$

and ε is the threshold for terminating the cutting plane iteration (see Section 4.2). Here $\|\cdot\|$ denotes ℓ_2-norm.

The proofs of the above two theorems are direct by following the proofs in [27], and are omitted here due to space limitations. In this work, we implement Algorithm 1 on the MATLAB platform of an Intel XeonX5450 workstation with 3.0 GHz CPU and 16 GB memory, and observe that the cutting plane iteration converges fast. For example, in the tag-to-region assignment experiment on the MSRC dataset (see Section 5.2), one optimization sub-problem between two vertices can be processed within only 0.1 seconds, which demonstrates that the cutting plane iteration of Algorithm 1 is efficient. Furthermore, as each optimization sub-problem in Eq. (11) is convex, its solution is globally optimal. Therefore, each optimization between two vertices will monotonically decrease the objective function, and hence the algorithm will converge. Figure 3 shows the convergence process of the iterative optimization, which is captured during the tag-to-region assignment experiment on the MSRC dataset. Here one iteration refers to one round of alternating optimizations for all vertex pairs, which corresponds to steps 5-16 of Algorithm 1. As can be seen, the objective function converges to the minimum after about 5 iterations.

5. EXPERIMENTS

In this section, we systematically evaluate the effectiveness of the proposed unified formulation and solution to various tag analysis tasks over three widely used datasets and report comparative results on two tag analysis tasks: (1) tag-to-region assignment, and (2) multi-label automatic tagging. We have also conducted extensive experiments on

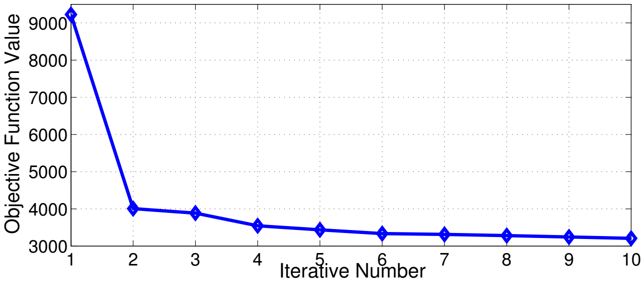

Figure 3: The convergence curve of Algorithm 1 on the MSRC dataset in the tag-to-region assignment experiment (See Section 5.2 for the details).

the tag refinement task over real-world social images and obtained better results than the state-of-the-art tag refinement methods in [7] and [8]. This result is not shown here due to space limitations.

5.1 Image Datasets

We perform the experiments on three publicly available benchmark image datasets: MSRC [11], COREL-5k [28] and NUS-WIDE [4], which are progressively more difficult. The MSRC dataset contains 591 images from 23 object and scene categories with region-level ground truths. There are about 3 tags on average for each image. The second dataset, COREL-5k, is a wildly adopted benchmark for keywords based image retrieval and image annotation. It contains $5,000$ images manually annotated with 4 to 5 tags, and the whole vocabulary contains 374 tags. A fixed set of $4,500$ images are used as the training set, and the remaining 500 images are used for testing. The NUS-WIDE dataset, recently collected by the National University of Singapore, is a more challenging collection of real-world social images from Flickr. It contains $269,648$ images in 81 categories and has about 2 tags per image. We select a subset of this dataset, focusing on images containing at least 5 tags, and obtain a collection of $18,325$ images with $41,989$ unique tags. We refer to this social image dataset as NUS-WIDE-SUB and the average number of tags for each image is 7.8.

5.2 Exp-I: Tag-to-Region Assignment

Our proposed unified solution is able to automatically assign tags to the corresponding regions within the individual images, which naturally accomplishes the task of tag-to-region assignment. In this subsection, we compare the quantitative results from our proposed unified solution with those from existing tag-to-region assignment algorithms.

5.2.1 *Experiment setup*

To evaluate the effectiveness of our proposed unified solution in the tag-to-region assignment task, two algorithms are implemented as baselines for comparison. (1) The kNN algorithm. For each segmented region of an image, we first select its k nearest neighboring regions from the whole semantic region collection, and collect the images containing these regions into an image subset. Then we count the occurrence number of each tag in the obtained image subset and choose the most frequent tag as the tag of the given semantic region. We apply this baseline with different parameter setting, i.e., k=49 and 99, and thus can obtain two results from this baseline. (2) The bi-layer sparse coding (bi-layer) algorithm in [12]. This algorithm uses a bi-layer sparse coding formulation to uncover how an image or semantic region can be robustly reconstructed from the over-segmented image patches of an image set, and is the state-of-the-art for tag-to-region assignment task.

The evaluation of the tag-to-region assignment is based on the region-level ground truth of each image, which is not available for the COREL-5k and NUS-WIDE-SUB datasets. To allow for direct comparison, here we report the performance comparisons on the MSRC-350 and COREL-100 datasets, both of which have been utilized to evaluate the tag-to-region performance in [12]. More specifically, the MSRC-350 dataset focuses on relatively well labeled classes in MSRC dataset, which contains 350 images annotated with 18 different tags. The second dataset, COREL-100, is the subset of COREL image set, which contains 100 images with 7 tags, and each image is annotated with the region-level ground truth. Besides, we fix the parameter λ to be 1 in all the experiments. The tag-to-region assignment performance is measured by pixel-level accuracy which is the percentage of pixels with agreement between the assigned tags and the ground truth.

Table 1: Tag-to-region assignment accuracy comparison on MSRC-350 and COREL-100 dataset. The kNN-based algorithm is implemented with different values for the parameter k, namely, kNN-1: k=49, kNN-2: k=99.

Dataset	kNN-1	kNN-2	bi-layer [12]	M-E Graph
MSRC-350	0.45	0.37	0.63	**0.73**
COREL-100	0.52	0.44	0.61	**0.67**

5.2.2 *Results and Analysis*

Table 1 shows the pixel-level accuracy comparison between the baseline algorithms and our proposed unified solution on the MSRC-350 and COREL-100 datasets. From the results, we have the following observations. (1) Our proposed unified solution achieves much better performance compared to the other three baseline algorithms. This clearly demonstrates the effectiveness of our proposed unified solution in the tag-to-region assignment task. (2) The contextual tag-to-region assignment algorithms including the bi-layer and our proposed multi-edge graph based unified solution outperform the kNN based counterparts. This is because the former two harness the contextual information among the semantic regions in the image collection. (3) The multi-edge graph based tag-to-region assignment algorithm clearly beats the bi-layer sparse coding algorithm, which owes to the fact that the structure of the multi-edge graph avoids the ambiguities among the smaller size patches in the bi-layer sparse coding algorithm. Figure 4 illustrates the tag-to-region assignment results for some exemplary images from MSRC-350 and COREL-100 datasets produced by our proposed unified solution. By comparing the results in [12], the results in Figure 4 are more precise, especially for the boundaries of the objects.

5.3 Exp-II: Multi-label Automatic Tagging

In this subsection, we evaluate the performance of our proposed unified formulation and solution in the task of automatic image tagging. Since our proposed unified solution is founded on the multi-edge graph, the learning procedure is essentially in a transductive setting.

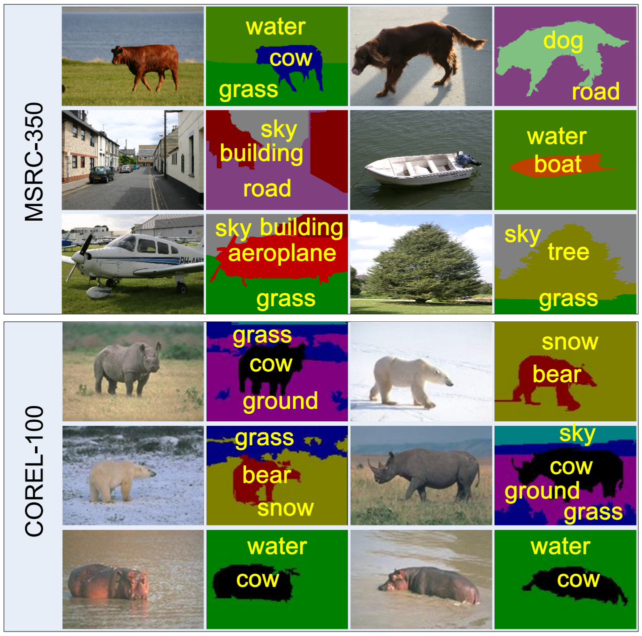

Figure 4: Example results of tag-to-region assignment on MSRC-350 and COREL-100 dataset. For better viewing, please see original color pdf file.

5.3.1 *Experiment Setup*

We compare our proposed multi-edge graph based semi-supervised learning algorithm against the following established semi-supervised multi-label learning algorithms for automatic image tagging. (1) The Multi-Label Local and Global Consistency (ML-LGC) algorithm proposed by Zhou et al. [17]. It decomposes the multi-label automatic image tagging problem into a set of independent binary classification problems, and solves each of them with a graph-based label propagation procedure. (2) The Constrained Non-negative Matrix Factorization (CNMF) algorithm proposed by Liu et al. [29]. It infers the multiple tags associated with an unlabeled image through optimizing the consistency between image visual similarity and tag semantic correlation. (3) The Semi-supervised Multi-label learning algorithm by solving a Sylvester Equation (SMSE) proposed by Chen et al. [30], which is similar to the algorithm in [31]. The algorithm constructs two graphs on the sample level and tag level, then defines a quadratic energy function on each graph, and finally obtains the tags of the unlabeled data by minimizing the combination of the two energy functions. Note that these three algorithms are actually the state-of-the-art algorithms for semi-supervised multi-label learning in literature, and the automatic image tagging based on these algorithms are implemented at the image-level without image segmentation requirement. We use the BoW model to represent each image as a 500 dimensional feature vector. (4) The Multiple-Instance Semi-Supervised Learning algorithm (MISSL) proposed by Rahmani et al. [32]. It transforms the automatic image tagging problem into a graph-based semi-supervised learning problem that works at both the image and the semantic region levels. We use the bag-of-regions representation in Section 3.1 as the multiple-instance representation of the images. Since MISSL can only handle binary classification, we decompose the multi-label learning task into a set of binary classification problems and solve each of them independently.

Table 2: Performance comparison of different automatic tagging algorithms on different datasets.

Dataset	method	Precision	Recall
MSRC (22 tags)	ML-LGC	0.62	0.55
	CNMF	0.65	0.61
	SMSE	0.67	0.62
	MISSL	0.63	0.60
	M-E Graph	**0.72**	**0.67**
COREL-5k (260 tags)	ML-LGC	0.22	0.24
	CNMF	0.24	0.27
	SMSE	0.23	0.28
	MISSL	0.22	0.29
	M-E Graph	**0.25**	**0.31**
NUS-WIDE-SUB (81 tags)	ML-LGC	0.28	0.29
	CNMF	0.29	0.31
	SMSE	0.32	0.32
	MISSL	0.27	0.33
	M-E Graph	**0.35**	**0.37**

Regarding the parameter setting for each algorithm, we consider the suggested parameter setting strategies in the original work and present the best result as the final output for each algorithm. Besides, we also fix the parameter λ in our algorithm to be 1 throughout the experiments in this section. In the experiments, we assign each image the top m tags with the highest confidence scores obtained from the automatic tagging algorithm, where m is the number of tags generated by our algorithm.

We evaluate and compare among the five multi-label automatic tagging algorithms over the three datasets. For the MSRC dataset, there are only 3 images annotated with tag "horse", so we remove this tag and evaluate different algorithms on the remaining 22 tags. We randomly and evenly split it into the labeled and unlabeled subsets. Since the training/testing data split in COREL-5k dataset has been provided by this dataset, we accordingly use the 4,500 training images as the labeled data and use the remaining 500 images as the unlabeled data. There are totally 260 distinct tags appearing in the testing set. For the NUS-WIDE-SUB dataset, we use the 81 semantic concepts which have been provided with ground truth annotations as the tagging vocabulary of this dataset[4]. We also randomly and evenly split the dataset into labeled and unlabeled subsets. For the performance evaluation metric we adopt the precision and recall, which are popularly applied in the multi-label image tagging tasks. We calculate precision and recall on each tag and average each of them from multiple tags to measure the final performance.

5.3.2 *Results and Analysis*

Table 2 shows the performance comparison of the above five algorithms over the three datasets. From the results, we

[4] It is worth noting that the multi-label image tagging task is greatly different from the tag refinement task in that the former aims at utilizing precisely labeled images to predict potential tags of the unlabeled images, while the latter works towards refining the imprecise tags provided by the users. Therefore, we cannot directly use the user-provided tags as the labels of the training images for the NUS-WIDE-SUB dataset.

have the following observations: (1) The proposed unified solution based on multi-edge graph outperforms the other four multi-label image tagging algorithms over all datasets. It indicates the effectiveness of the proposed unified solution in the automatic image tagging task. (2) Our proposed unified solution outperforms the MISSL which is also based on the semantic regions, as our algorithm explicitly explores the region-to-region relations in the multi-edge graph structure.

6. CONCLUSIONS AND FUTURE WORK

We have introduced a new concept of multi-edge graph into image tag analysis, upon which a unified framework is derived for different yet related analysis tasks. The unified tag analysis framework is characterized by performing cross-level tag propagation between a vertex and its edges as well as between all edges in the graph. A core equation is developed to unify the vertex tags and the edge tags, based on which, the unified tag analysis is formulated as a constrained optimization problem and the cutting plane method is used for efficient optimization. Extensive experiments on two example tag analysis tasks over three widely used benchmark datasets have demonstrated the effectiveness of the proposed unified solution. For future work, we will pursue other applications of the multi-edge graph, such as analyzing user behavior in social network or mining knowledge from the rich information channels of multimedia documents.

7. ACKNOWLEDGMENT

This work is supported by the CSIDM Project No.CSIDM-200803 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore.

8. REFERENCES

[1] Flickr. http://www.flickr.com/.

[2] Picasa. http://picasa.google.com/.

[3] L. Kennedy, S.-F. Chang, and I. Kozintsev. To search or to label?: Predicting the performance of search-based automatic image classifiers. In *MIR*, 2006.

[4] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *CIVR*, 2009.

[5] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *WWW*, 2009.

[6] M. Ames and M. Naaman. Why we tag: Motivations for annotation in mobile and online media. In *CHI*, 2007.

[7] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Content-based image annotation refinement. In *CVPR*, 2007.

[8] D. Liu, X.-S. Hua, M. Wang, and H.-J. Zhang. Image retagging. In *MM*, 2010.

[9] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *ICCV*, 2007.

[10] Y. Chen, L. Zhu, A. Yuille, and H.-J. Zhang. Unsupervised learning of probabilistic object models (poms) for object classification, segmentation, and recognition using knowledge propagation. *TPAMI*, 2009.

[11] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006.

[12] X. Liu, B. Cheng, S. Yan, J. Tang, T. Chua, and H. Jin. Label to region by bi-layer sparsity priors. In *MM*, 2009.

[13] J. Li and J. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *TPAMI*, 2003.

[14] E. Chang, K. Goh, G. Sychay, and G. Wu. Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines. *TCSVT*, 2003.

[15] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR*, 2003.

[16] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[17] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003.

[18] D. Zhou and C. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, 2007.

[19] D. Zhou, S. Zhu, K. Yu, X. Song, B. Tseng, H. Zha, and C. Giles. Learning multiple graphs for document recommendations. In *WWW*, 2008.

[20] D. Zhou, J. Huang, and B. Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *NIPS*, 2006.

[21] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[22] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 2000.

[23] B. Russell, W. Freeman, A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[24] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 2004.

[25] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[26] J. Kelley. The cutting-plane method for solving convex programs. *JSIAM*, 1960.

[27] T. Joachims. Training linear svms in linear time. In *KDD*, 2006.

[28] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about corel - evaluation in image retrieval. In *CIVR*, 2002.

[29] Y. Liu, R. Jin, and L. Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *AAAI*, 2006.

[30] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *SDM*, 2008.

[31] Z.-J. Zha, T. Mei, J. Wang, Z. Wang, and X.-S. Hua. Graph-based semi-supervised learning with multiple labels. *JVCI*, 2009.

[32] R. Rahmani and S. Goldman. MISSL: Multiple instance semi-supervised learning. In *ICML*, 2006.