

Tagging Social Images by Parallel Tag Graph Partitioning

ZHENG LIU^{1,2}, HUIJIAN HAN² AND HUA YAN^{1,2}

¹*School of Computer Science and Technology
Shandong University of Finance and Economics
Ji'nan, 250014 P.R. China*

²*Shandong Provincial Key Laboratory of Digital Media Technology
Ji'nan, 250014 P.R. China*

In recent years, we have witnessed a great success of social community websites. Large-scale social images with rich metadata are increasingly available on the Web. In this paper, we focus on efficiently tagging social images by partitioning the large-scale tag graph in parallel. Vertices of the tag graph are constructed by the candidate tags which are extended from initial tags. Initial tags are extracted from the rich metadata of social images, including user supplied tags, notes data and group information. Edge weight of the tag graph is calculated by combining two parameters, which are related to image visual features and tag co-occurrence. Both global and local features are considered in parameter 1. For each candidate tag, a neighbor images voting algorithm is performed to calculate parameter 2. As the tag graph may be large-scale, we utilize a parallel graph partitioning algorithm to accelerate the graph partitioning process. After the tag graph is partitioned, we rank all the sub-graphs according to the edge weight within one sub-graph. Afterwards, final tags are selected from the top ranked sub-graphs. Experimental results on Flickr image collection well demonstrate the effectiveness and efficiency of the proposed algorithm. Furthermore, we apply our social image tagging algorithm in tag-based image retrieval to illustrate that our algorithm can really enhance the performance of social image tagging related applications.

Keywords: social image, Flickr, tag, parallel graph partitioning, image retrieval

1. INTRODUCTION

With the popularity of various social media applications, massive social images associated with user supplied tags have been made available in many social media websites in recent years. The popularity of photo-sharing websites like Flickr gives us a chance to observe what ordinary users do in their daily life. Flickr is an image hosting and video hosting website, web services suite, and online community created by Ludicorp and later acquired by Yahoo!. In addition to be a popular website for users to share and embed personal photographs, the service is widely used by bloggers to host images that they embed in blogs and social media. In August 2011, it was reported that Flickr had held more than 6 billion images.

With the rapid development of Web social community, the applications which exploit the social media resources, such as Flickr and Wikipedia, have become popular and attracted more attentions from both academia and industry [1]. In particular, social media community allows the users to provide personalized tags when uploading photos, and then users can tag social images through user-supplied tags and other metadata. The tags which describe the content of images can help users easily manage and access large-scale

Received February 24, 2012; revised May 26 & July 19, 2012; accepted August 29, 2012.
Communicated by Chia-Feng Juang.

image datasets. With these metadata, the manipulations of image data can be easier to be accomplished, such as browsing, indexing and retrieval [2].

Automatic social images tagging plays a critical role in modern tag-based image retrieval systems. Existing tagging methods mostly perform image tagging based on community contributed resources. Unfortunately, such social resources usually contain dirty and incomplete tags, which severely limit the performance of these tagging methods. To tackle these problems, we propose a novel approach to tag social images by partitioning tag graph in parallel, and the major contributions and innovations of our work include:

- The rich metadata of social images are fully utilized in image tagging process. To enrich semantic information of the images to be tagged, not only user-supplied tags but also notes data and group information are adopted in this paper.
- Combining image visual features and tag co-occurrence, two parameters are used to calculate edge weight of the tag graph. Therefore, the tag graph may include more useful information of social image.
- Extended tags are generated from initial tags through Flickr API (flickr.tags.getRelated API)
- We provide an neighbor images voting algorithm to estimate the relationship between candidate tags and the image to be tagged.
- Tag graph is partitioned in parallel, hence, the proposed algorithm can be executed with high efficiency.

The rest of this paper is organized as follows. Section 2 presents the related works about social image tagging. In Section 3, we introduce the framework of our approach and the methods to construct tag graph. Section 4 explains how to obtain social image tags by tag graph partitioning in parallel, particularly, we give a scheme to select final tags from tag graph partitioning results. Experimental results and related analysis are given in section 5. Finally, we draw conclusions and point out future works in Section 6.

2. RELATED WORKS

In this section, we summarize and analyze representative methods which are closely to the techniques used in this paper.

The first category of related techniques refers to social images tagging. Many pioneering works which are related to the problem of social images tagging have been done in recent years. Particularly, there are some existing works focusing on user-supplied tags of social images. Liu *et al.* proposed an approach to rank the tags for each image according to their relevance levels [2]. A new Flickr distance was proposed to measure the visual similarity between concepts according to Flickr [3]. Schmitz proposed a method to build the facted ontology from Flickr's tagging resources [4]. Chen *et al.* also proposed to use the predicted tags to search for groups as recommendation groups for the given image [5]. The learning based tag recommendation approach has been introduced to generate ranking features from multi-modality correlations, and learns an optimal combination of these ranking features by the Rankboost algorithm [6]. Ames *et al.* have explored the motivation of tagging in Flickr website and they claim that most users tag

images to make them better accessible to the general public [7]. Kennedy *et al.* have evaluated the performance of the classifiers trained with Flickr images and associated tags and demonstrate that tags provided by Flickr users actually contain many noises [8].

Particularly, automatic social image tagging attracts the attentions of more and more researchers in multimedia retrieval domain. Si *et al.* provided an effective social image annotation method by cross-domain subspace learning [9]. Zhou *et al.* proposed a method to better align the images with the social tags by clustering images to reduce the uncertainty of the relatedness between images and tags and then re-ranking tags using a cross-modal tag correlation network [10]. Wu *et al.* proposed a machine learning framework to mine social images and investigate its application in automatic image tagging [11]. Tang *et al.* present a semantic-gap-oriented active learning method, which incorporates the semantic gap measure into the information-minimization-based sample selection strategy [12]. Recently, we proposed a social images tagging method by probabilistic topic model and tag association mining, and experimental results show the effectiveness and efficiency of the proposed approach [13].

As a powerful computing tool, graph model has been widely adopted in image tagging research field as follows. Liu *et al.* proposed a graph learning framework for image annotation. The authors proposed a Nearest Spanning Chain method to construct the image-based graph of which edge-weights are derived from the chain-wise statistical information instead of the traditional pairwise similarities [14]. Rui *et al.* proposed a bipartite graph reinforcement model for web image annotation. All candidate annotations are modeled as a bipartite graph [15].

Jin *et al.* investigated to prune irrelevant keywords by the usage of WordNet and re-formulate the removal of erroneous keywords from image annotation problem into graph-partitioning problem, which is weighted MAXCUT problem [16, 17]. Based on Jin's works, we present an algorithm to solve image annotation refinement problem by graph partition and image search engine [18].

As is well known, dimension reduction is quite important for large-scale data processing. Computer vision based dimension reduction methods are listed as follows.

Xie *et al.* present a multi-view stochastic neighbor embedding that systematically integrates heterogeneous features into a unified representation for subsequent processing based on a probabilistic framework [19]. Guan *et al.* proposed a non-negative patch alignment framework to unify popular non-negative matrix factorization related dimension reduction algorithms [20]. Xia *et al.* developed a multi-view spectral embedding algorithm, which can encode different features in different ways to achieve a physically meaningful embedding [21]. Guan *et al.* introduced the manifold regularization and the margin maximization to nonnegative matrix factorization (NMF) and obtained the manifold regularized discriminative NMF to prevent ignoring both the local geometry of data and the discriminative information of different classes [22]. Recently, Guan *et al.* make further research on nonnegative matrix factorization [23, 24].

Ranking is one of the most important problems in information retrieval, and tag ranking approach is also critical for our works. Hence, we present two important ranking methods in the multimedia data ranking field. Tian *et al.* adopted interactive video search re-ranking to bridge the semantic gap by introducing user's labeling effort. They utilized sparse transfer learning to effectively and efficiently encode user's labeling information [25]. In the research field of web image search, Tian *et al.* studied on how to effective

capture the user's intention when the query term is ambiguous in order to promote the performance of re-ranking Web images [26].

Different from the existing related works, this paper presents a novel social image tagging approach by converting the social image tagging problem to parallel tag graph partitioning.

3. CONSTRUCTING TAG GRAPH OF SOCIAL IMAGE

The main idea of our algorithm lies in that we organize the rich metadata of social image to tag graph. Therefore, we should explain how to construct the tag graph in advance.

3.1 Overview of the Proposed Algorithm

Social image community allows users to tag their uploaded media data with descriptive keywords called tags. As an example, Fig. 1 illustrates a social image in Flickr and its associated user-provided tags. As user-provided tags are usually noisy and incomplete, we apply other kind of image metadata in image tagging.

As is shown in Fig. 2, the proposed approach is mainly composed of two steps. In the first step, we construct the tag graph for the image to be tagged. The following is the definition of tag graph.



Fig. 1. An example of social image with user-supplied tags.

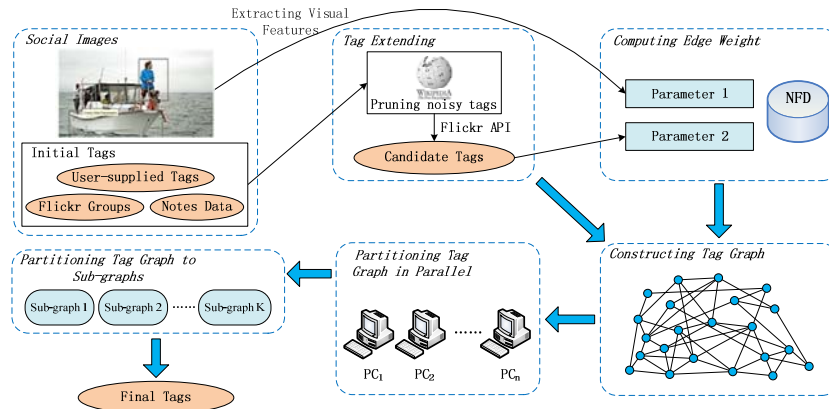


Fig. 2. Framework of the proposed approach.

Definition 1 (Tag Graph): For an image to be tagged, tag graph is constructed by using candidate tags as the vertices and adopting the relationship between a pair of tags as the edge weight.

Vertices of the tag graph are corresponding to candidate tags, which are made up of user-supplied tags, notes data and user group information. Edge weight is calculated using NFD distance (see Section 3.3.1) weighted by two parameters. In the second step, the large-scale tag graph is partitioned in parallel effectively, and then final tags are chosen from sub-graphs.

3.2 Choosing Candidate Tags

Candidate tags are made up of initial tags and extended tags, and in this subsection, we will illustrate how to obtain candidate tags from initial tags and extended tags.

3.2.1 Obtaining initial tags from metadata of social images

The user-supplied tags may contain noisy or uncorrelated tags, such as misspelling, meaningless words and numbers. Therefore, we should perform a pre-processing procedure to prune the un-related tags. We submit each tag as a query to Wikipedia, and only the tags which have a coordinate in Wikipedia are reserved. After the un-related tags pruning, the rest of the reserved user-supplied tags are denoted as Γ^U .

Flickr allows users to organize themselves in self-managed communities, called Flickr Groups [27]. For a Flickr group, the name of it may represent the semantic of photos belonged to this group. Therefore, we add the names of Flickr groups to candidate tags.



Fig. 3. An image¹ with notes data in Flickr.

As is shown in Fig. 3, notes data is another kind of important metadata in Flickr. A note is a specific interesting region (bounding box) defined by users. The metadata of a note includes the note author, the note text, the position and the size of the box [28]. In this work, we add note texts in the candidate tags.

From above, we can see that three kinds of metadata are combined together to construct initial tags (Γ^T), such as User-supplied tags (Γ^U), Flickr groups (Γ^G) and Notes data (Γ^N), and $\Gamma^T = \Gamma^U \cup \Gamma^G \cup \Gamma^N$ is satisfied. Particularly, the words in Flickr group names and notes data also need to perform noisy words pruning process. Apart from noise pruning, the stop words in initial tags should be deleted and word stemming should also be performed. As many users of Flickr do not use English to describe their photos, we adopt Google translate API² to translate non-English to English.

¹ <http://www.flickr.com/photos/epsos/4939044794/>

² <http://code.google.com/apis/language/>

3.2.2 Obtaining extended tags from initial tags

To enrich the semantic information in candidate tags, we apply a Flickr API³ to obtain related tags, and then generate extended tags. Flickr can be considered as a Web-scale image semantic space, according to the Flickr's Related Tag API (flickr.tags.getRelated), each tag has a list of "related" tags. As is shown in Table 1, we give an example to illustrate the candidate extended tags generated by flickr.tags.getRelated API [1].

Table 1. Example of related tags generated by Flickr API.

Tag	Related tags generated by Flickr API
Island	sea beach water sky blue clouds ocean sunset sand sun nature landscape boat summer travel vacation trees italy coast rocks greece green waves mar cloud mare holiday paradise tree canon nikon boats thailand bay orange light tropical italia reflection white ship playa

Supposing tag t_i denotes the i th tag in initial tags, the related tags of which are represented as $R(t_i)$. We merge all the related tags together and eliminate duplicated tags to build up the candidate extended tag set Γ^E .

$$\Gamma^E = \bigcup_{t_i \in \Gamma^T} R(t_i) = \{e_1, e_2, \dots, e_k\} \quad (1)$$

To make the extended tags more relevant to the image to be tagged, two factors are considered in our tag extended policy. Firstly, the influence of higher ranked initial tags is boosted. Secondly, the semantic relevance between tags is taken into account as well. The score of candidate tag e_j is designed as follows.

$$\text{score}(e_j) = \begin{cases} \sum_{t_i \in \Gamma^T} \left\{ \frac{|\Gamma^T| - i + 1}{|\Gamma^T|} \cdot \text{NGD}(e_j, t_i) + \tau^U(e_j) + \tau^G(e_j) + \tau^N(e_j) \right\}, & e_j \in R(t_i) \\ \sum_{t_i \in \Gamma^T} \{ \tau^U(e_j) + \tau^G(e_j) + \tau^N(e_j) \}, & e_j \notin R(t_i) \end{cases} \quad (2)$$

where NGD is a distance function between two words by searching a pair of words using the Google search engine [29]. If initial tags have not been ranked, $(|\Gamma^T| - i + 1)/|\Gamma^T|$ is deleted from Eq. (2). Particularly, we promote the importance of candidate extended tag e_j by three parameter $\tau^U(e_j)$, $\tau^G(e_j)$ and $\tau^N(e_j)$ as follows.

$$\tau^U(e_j) = \begin{cases} \varphi^U, & \text{if } e_j \in \Gamma^U \\ 0, & \text{else} \end{cases}, \tau^G(e_j) = \begin{cases} \varphi^G, & \text{if } e_j \in \Gamma^G \\ 0, & \text{else} \end{cases}, \tau^N(e_j) = \begin{cases} \varphi^N, & \text{if } e_j \in \Gamma^N \\ 0, & \text{else} \end{cases} \quad (3)$$

According to the score of related tags calculated by Eq. (2), the tags with high scores would be reserved as extended tags. In this paper, the parameters φ^U , φ^G and φ^N are set to 0.5, 0.7 and 0.9 respectively. As is shown in Fig. 4, we illustrate how to obtain extended tags from initial tags. For simplicity, we suppose that initial tags of the image in Fig. 4 only consist of user-supplied tags.

³ <http://www.flickr.com/services/api/>

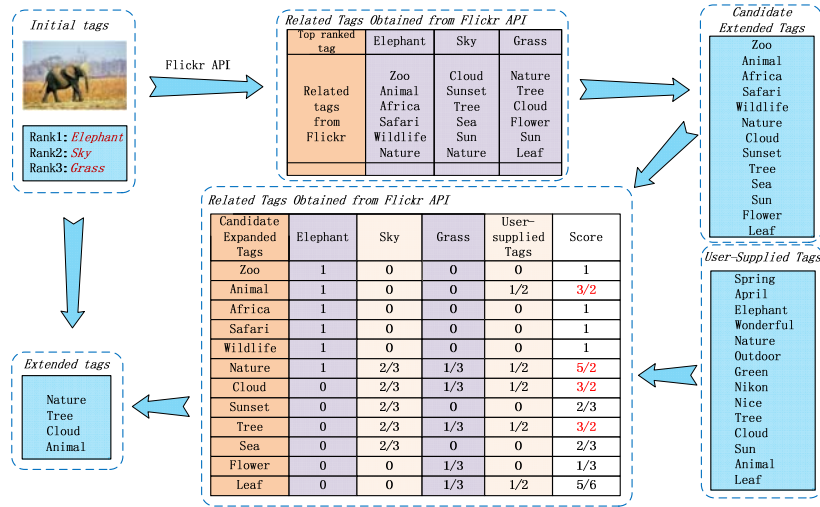


Fig. 4. Illustration of extended tags generating method.

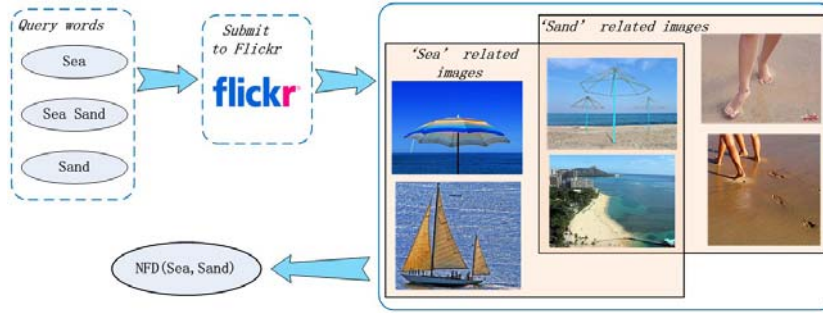


Fig. 5. Illustrations of NFD distance estimating.

Afterwards, we merge all the initial tags and extended tags together to build up the candidate tags after eliminating duplicated tags.

3.3 Calculating Edge Weight

This subsection illustrates the methods to compute edge weight of the tag graph.

3.3.1 Image similarity metric

We define a method named NFD which is analogous to NGD [30] to compute the concurrence similarity between tags based on tag concurrence. As is shown in Fig. 5, NFD between two tags can be estimated based on Flickr as follows,

$$NFD(t_i, t_j) = \frac{\max\{\log f(t_i), \log f(t_j)\} - \log f(t_i, t_j)}{\log G - \min\{\log f(t_i), \log f(t_j)\}} \quad (4)$$

where t_i and t_j represent the two tags in consideration. $f(t_i)$ and $f(t_j)$ are the numbers of images containing tag t_i and tag t_j respectively, which can be obtained by searching Flickr website with the tags as keywords. $f(t_i, t_j)$ is the number of the images returned by Flickr when typing t_i and t_j as the search terms respectively. Moreover, G is the total number of images in Flickr.

3.3.2 Parameter 1

For a given tag, parameter 1 represents the visual similarity between the host image and the images which are tagged by the proposed tag. A search-based method is used in this paper. Given a tag, we submit it to Flickr as the query word firstly, and then compute the visual similarity between image and searching results.

Considering the different application scenarios of global and local features, we use both of them to measure visual similarity. Local features could perform better than global features when the image containing salient objects. Otherwise, global features may play more important roles. Therefore, we introduce both global and local features in our approach, and dynamically tune the weight of them to enhance the image content analysis capability.

Table 2. The low-level features extracted from images.

Feature category	Feature Name	Dimensions
Color	Color Correlogram	44
	Color Texture Moment	14
	Color Moment	6
Texture	Wavelet Features [31]	104

We totally extracted 168-dimension color and texture features (shown in Table 2) as the low-level visual representation of the images. In addition, we employ cosine similarity to estimate the visual similarity between a pair of images based on global features (Shown in Eq. (5)).

$$Sim_G(I_i, I_j) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|} \quad (5)$$

where v_i and v_j are the global feature vectors of I_i and I_j respectively.

Inspired by the recent progress in object recognition, we use the visual word model and SIFT features [32] to measure image similarity. We use SIFT to describe the regions around the keypoints. To construct codebook, we use corel5k dataset [33] as the training data. All SIFT descriptors in the image of corel5k dataset are grouped into clusters, and the centroid of each cluster is acted as visual word. In this paper, visual words are obtained by vector quantization with the Linde-Buzo-Gray (LBG) algorithm [34]. After vector quantization, all images are represented as a D -dimensional vector, and the value of D is equal to the number of visual words. Afterwards, image visual similarity is computed by the distance between feature vectors. In our experiments, 2000 visual words are utilized to compute the visual similarity.

Supposing that visual words vector h_i and h_j are D -dimensional (D equals to the vocabulary size of visual words) vectors of visual word frequencies, which come from image I_i and image I_j respectively. Then, the image similarity based on the visual words model is computed as follows.

$$Sim_L(I_i, I_j) = \frac{h_i \cdot h_j}{\|h_i\| \|h_j\|} \quad (6)$$

Afterwards, the overall image similarity can be obtained by linearly combining both global and local features as follows.

$$Sim(I_i, I_j) = \alpha \cdot Sim_G(I_i, I_j) + (1 - \alpha) \cdot Sim_L(I_i, I_j), 0 < \alpha < 1 \quad (7)$$

As is shown in Eq. (7), the parameter α is used to adjust the influence of global and local features on image similarity measuring.

To choose the most similar images, a predefined threshold (denoted as σ) is set. The similar image set (denoted as $S(I_u)$) of I_u is considered as follows,

$$S(I_u) = \{I_k | Sim(I_k, I_u) > \sigma, k = 1, 2, \dots, n\} \quad (8)$$

where I_u is the image to be tagged, n is the number of images return from Flickr. In the experiment, the parameters σ and n are set to 0.25 and 20 respectively. Then, parameter 1 of tag t is calculated by Eq. (9).

$$\delta_t^1 = \frac{\sum_{r_i \in S(I_u, t)} 1 - Sim(I_u, r_i)}{|S(I_u, t)|} \quad (9)$$

3.3.3 Parameter 2

Following the main idea of paper [35], we modify the definition of neighbor images to make the neighbor voting policy more efficiently for social image tagging. For an image to be tagged, the *neighbor images* are the images tagged by at least one tag which is belonged to user-supplied tags set of the given image. We collect images from social image community to construct the neighbor images set. As is shown in Fig. 6, each user-supplied tag of the image to be tagged is submitted to social image community (e.g. Flickr), and then related images with user-supplied tags are obtained. To make the voting process more objective, the number of neighbor images which are provided by the same user is limited to 2.

Algorithm 1: Neighbor Voting for Candidate Tags

Input: Image I_u with its candidate tags set $C = \{c_1, c_2, \dots, c_n\}$

Output: Normalized voting score of each candidate tag

- 1) The initial voting score of each candidate annotation is set to 0, the voting score of c_i is represented as S_i
-

-
- 2) **For** $i = 1$ to n // n is the number of candidate tags
Submitting c_i to social image community and obtaining related images (N^i).
 - 3) **End for**
 - 4) Constructing neighbor images set (denoted as N)

$$N = \{I_1, I_2, \dots, I_k\} = \bigcup_{i=1}^n N^i$$
 - 5) **For** $i = 1$ to n // each for-loop can calculate the voting score of a candidate tag
 - 6) **For** $x = 1$ to k // k is the number of neighbor images
 - 7) After noise tags pruning, the user-supplied tags set (denoted as T_x) is obtained
 - 8) $T_x = \{t_{x1}, t_{x2}, \dots, t_{xu}\}$
 - 9) **For** $y = 1$ to u // u is the number of user-supplied tags of I_x
 - 10) $S_i = S_i + \text{NFD}(c_i, t_{xy})$
 - 11) **End for**
 - 12) **End for**
 - 13) **Return** $\bar{S}_i = \frac{S_i}{k \times u}$ // normalizing the voting score
 - 14) **End for**
-

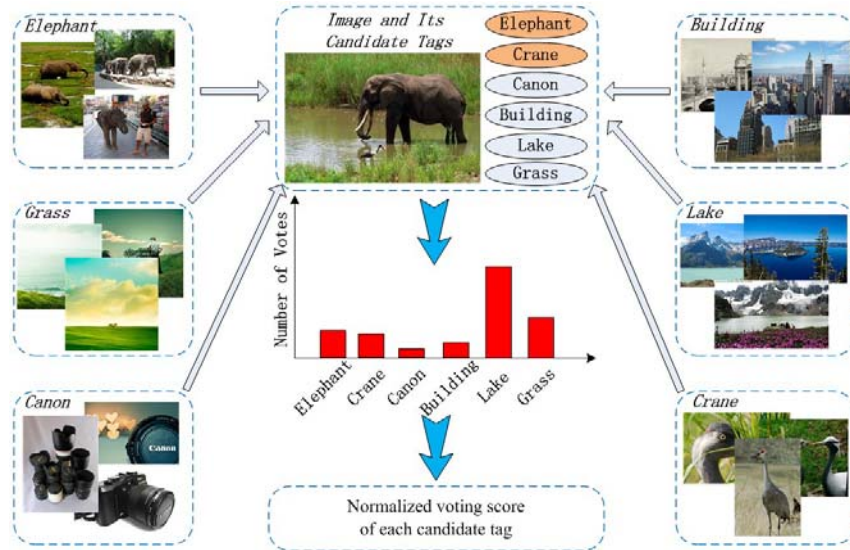


Fig. 6. Illustration of neighbor voting process.

The output of Algorithm 1 ($\bar{S}_i$) is used as parameter 2 for tag $t(\delta_t^2)$.

3.3.4 Calculating edge weight by integrating parameter 1 and parameter 2

Afterwards, we can integrate two parameters together to calculate the weight for tag t as follows.

$$\delta_t = \theta \cdot \delta_t^1 \cdot \delta_t^2 \quad (10)$$

Parameter θ is used to promote the importance of initial tags. If tag t is belonged to initial tags, θ is set to 1.5, otherwise, θ is set to 1.

Definition 2 (Modified edge weight): Let δ_p and δ_q be the weight of candidate tag t_p and t_q respectively, the modified edge weight (denoted as w_{pq}) of E_{pq} is computed as follows.

$$\lambda_{pq} = (\delta_p + \delta_q)/2 \quad (11)$$

$$w_{pq} = \lambda_{pq} \cdot NFD(t_p, t_q) \quad (12)$$

4. OBTAINING SOCIAL IMAGE TAGS BY TAG GRAPH PARTITIONING IN PARALLEL

In section 3, we utilize social image tags and other metadata to establish tag graph. In this section, we will discuss how to obtain final tags from the tag graph.

4.1 Partitioning the Tag Graph in Parallel

In this section, we conduct k -way tag graph partitioning in parallel through three steps, including graph coarsening, initial partitioning, and refinement.

4.1.1 Graph partitioning problem statement

As is illustrated in paper [36], the graph partitioning problem can be formally described as follows.

Let $G = (V, E)$ be an undirected graph of vertices V , with edges E . Both vertices and edges can be weighted, $|v|$ denotes the weight of vertex v and $|e|$ denotes the weight of edge e . A partition of the graph is a mapping of V into K disjoint sub-domains, such that $V_i \cap V_j = \Phi$ and $V_1 \cup V_2 \cup \dots \cup V_k = V$. To balance the partitioning results, the balance condition is defined as the maximum sub-domain weight (denoted as S), $S = \max(|V_k|)$, $k \in [1, K]$.

4.1.2 Partitioning the large-scale tag graph in parallel

We utilize a software package which is named PARMETIS 4.0 to partition the tag graph in parallel. PARMETIS is an MPI-based parallel library which implements a variety of algorithms for partitioning and repartitioning unstructured graphs and for computing fill-reducing orderings of sparse matrices [37]. The parallel graph partitioning algorithm used in ParMETIS_V3_PartKway is based on the serial multilevel k -way partitioning algorithm described in [38] and [39] and parallelized in [40] and [41]. This algorithm is made up of three steps: graph coarsening, initial partitioning, and refinement.

In the coarsening phase, a sequence of smaller graphs $G_i = (V_i, E_i)$ is constructed from the original graph $G_0 = (V_0, E_0)$ such that $|V_i| < |V_{i-1}|$. In most coarsening schemes, a set of vertices of G_i is combined to form a single vertex of the next level coarser graph G_{i+1} . Let V_i^v be the set of vertices of G_i combined to form vertex v of G_{i+1} . In order for a

partitioning of a coarser graph to be good with respect to the original graph, the weight of vertex v is set equal to the sum of the weights of the vertices in V_i^v .

The second phase of a multilevel k -way partitioning algorithm is to calculate a k -way partitioning P_m of the coarse graph $G_m = (V_m, E_m)$ such that each partition contains roughly $|V_0|/k$ vertex weight of the original graph, and G_m contains sufficient information to intelligently enforce the balanced partitioning and the minimum edge-cut requirements.

In the refinement phase, the partitioning P_m of the coarser graph G_m is projected back to the original graph, by going through the graphs $G_{m-1}, G_{m-2}, \dots, G_1$. Since each vertex v of G_{i+1} contains a distinct subset of vertices V_i^v of G_i , P_i is obtained from P_{i+1} by simply allocating the set of vertices V_i^v to the partitioning $P_{i+1}[v]$; i.e., $P_i[u] = P_{i+1}[v]$, $\forall u \in V_i^v$.

4.2 Selecting Final Tags

After the tag graph partitioning process, K sub-graphs are obtained, that is, the candidate tags are divided to K parts. Then, the important task is how to select final tags from these sub-graphs. We design an algorithm to choose final tags by estimating the relationship of all the tags within one sub-graph as follows.

Algorithm 2: Sub-graphs Ranking

Input: K sub-graphs obtained from the tag graph partitioning (denoted as $\{C_1, C_2, \dots, C_K\}$)

Output: Ranking list of the K sub-graphs according to the relevance between the host image and vertices in sub-graphs.

- 1) $AVG(C_i) = 2 \cdot \frac{\sum_{t_g, t_h \in C_i, g > h} W(E_{gh})}{|C_i|(|C_i| - 1)}$, $i \in [1, K]$
 - 2) according to the value of function $AVG(\cdot)$, all sub-graphs are ranked initially
 - 3) **For** two sub-graphs (denoted as C_i and C_j , C_i has higher rank than C_j) which are adjacent to each other
 - 4) **If** $\frac{|AVG(C_i) - AVG(C_j)|}{MAX\{AVG(C_i), AVG(C_j)\}} < \mu$ **then** //parameter μ is set to 0.1.
 - 5) $STDEV(C) = \sqrt{\frac{2 \cdot \sum_{t_g, t_h \in C, g > h} (W(E_{gh}) - AVG(C))^2}{|C|(|C| - 1)}}$
 - 6) **If** $STDEV(C_i) > STDEV(C_j)$
 - 7) swap the position of C_i and C_j in the ranking list
 - 8) **End if**
 - 9) **End if**
 - 10) **End for**
-

Afterwards, the tags in top ranked sub-graphs are reserved as final tags.

5. EXPERIMENTS AND ANALYSIS

All our experiments are conducted on a social image dataset collected from Flickr, and three experiments are designed to make performance evaluating.

5.1 Experiment 1: Performance Evaluating for the Proposed Algorithm

We construct our own dataset by collecting 35 popular photo categories from Flickr by Flickr API, and all the photo categories are listed in Table 3. As Flickr provides a service to give users the relevant photos according to users' query, each photo category is built up by submitting a category name to Flickr. Each photo category consists of 100 photos, and each photo has at least 10 user-supplied tags. We invite four researchers in multimedia research field to provide ground truths for this dataset.

Table 3. Photo categories used in experiment 1.

	Class 1	Class 2	Class 3
Name of photo categories	tiger car harbor Toyota bridge yacht boat shirt cruiser carriage brush aircraft cabinet furniture reef	fall lake winter river mountain brick carpet forest wall	park sunset vegetable wood rainbow machine dish candy street island bread

Furthermore, we manually classify the above photo categories to three classes: (1) **Class 1**-Containing salient objects possibly, (2) **Class 2**-Containing salient objects with little possibility, (3) **Class 3** Uncertain.

The accuracy rate AR is used to estimate the effectiveness of the proposed method for social image tagging as follows.

$$AR(P) = \frac{1}{|P|} \sum_{i=1}^{|P|} \frac{num_true(I_i)}{num_all(I_i)} \quad (13)$$

where P is the image set to be evaluated, and I_i is an image belonged to P . Function $num_true(I_i)$ returns the number of true tags of image I_i , and $num_all(I_i)$ represents the number of final tags of image I_i . Final tags are made up of the tags in top ranked three sub-graphs.

In edge weight computing process, we adopt both global features and local features, and linearly combining them together by the parameter α . To estimate the visual similarity more precise, we design an experiment to seek the optimal value of α for each photo category as shown in Fig. 7.

From Fig. 7, we can see that to enhance accuracy rate when tagging the images in Class 1, 2 and 3, we set α to 0.4, 0.6 and 0.5 respectively.

In this experiment, we compare our approach with other two different scenarios: (1) edge weight computed without parameter 1, that is, only parameter 2 is considered in this case; (2) edge weight calculated without parameter 2. Fig. 8 shows that our approach outperforms the other two schemes, as incorporating visual feature with tag co-occurrence

can obtain more accurate edge weight for social image tagging task. The average values of accuracy rate of the three schemes (“Without parameter 1”, “Without parameter 2” and “Our approach”) are 0.503, 0.552 and 0.652 respectively.

Afterwards, we evaluate the effectiveness of the tag graph partitioning and sub-graphs selecting policy, which are the key part of the proposed approach. The top three sub-graphs are evaluated for each photo category, therefore, we test the tagging accuracy of top three sub-graphs respectively.

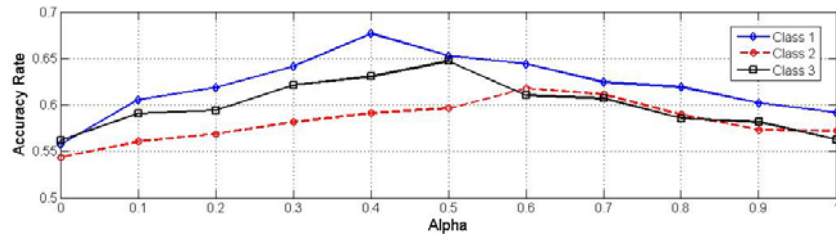


Fig. 7. Accuracy rate for each photo class when alpha changing.

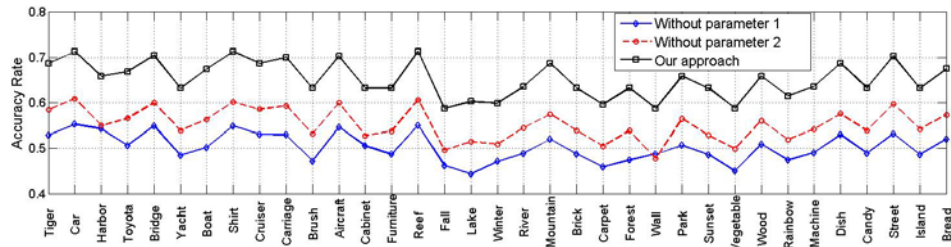


Fig. 8. Tagging accuracy rate of our approach and other two schemes.

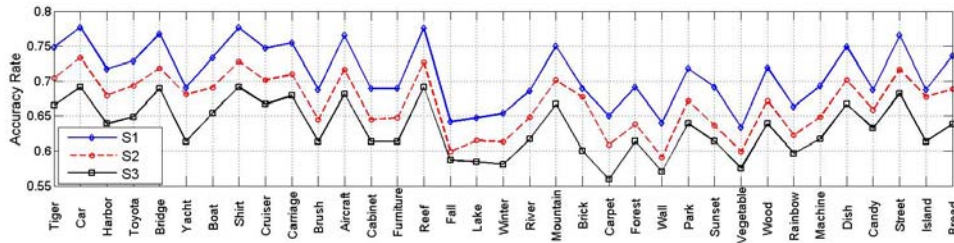


Fig. 9. Tagging accuracy rate of the tags in the top ranked three sub-graphs.

From Fig. 9, we can see that accurate rate of the tags in sub-graph S1 is higher than S2 and S3, and S2 performs better than S3. For all the 35 photo categories, the average accuracy rate of final tags for S1, S2 and S3 are 0.71, 0.669 and 0.633 respectively. This shows that the graph partitioning and sub-graphs ranking algorithm are effective, most of the true tags are positioned in top ranked sub-graphs.

Next, we evaluate the influence of initial tags' accuracy rate to final tags. In this experiment, we test our approach in seven classes. As is shown in Fig. 10, “class 0.1” concludes all the images in the dataset with the value of initial tags accuracy rate be-

tween [0.1, 0.2). Other class is set just like “class 0.1”. The experimental shows that when the accuracy rate of initial tags exceed 0.3, accuracy rate of initial tags has little effect on the growth of accuracy rate of final tags. This shows that when accuracy rate of initial tags are not very low, tag extending process can complement semantic information of initial tags effectively. In Fig. 11, we give a social image tagging example.

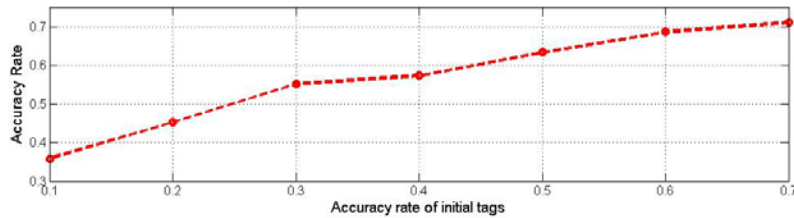


Fig. 10. Influence of initial tags' accuracy rate to final tags.

Image	User-supplied tags	Notes data	Flickr groups
	Boat ship boating nautical sailboat yacht fishing tour cruising crew passengers voyage ocean sea water search sailors mariners marine vessel vacations судно лодка яхта more 游船艇 海 barco yate mar ポート ヨット navire bateau mer navio iate Schiff boot meer سفينة بحر بيخت أبجر زورق بة 보트 요트 바다 gemi tekne yat deniz كشتى nave barca mare kapal perahu pesiar laut barko Bangka dagat statek łódź jacht morze	Cruise Ship Passenger Fishing Boat Watch Yachts	People piers boardwalks thing sea side Ship Sail Boat Vessel Ocean Cruise River Seven Sea Tourist Action Sad ROPES Worldwide Ship Spotters Barche Mad Mariner Yacht AQUELLOS MAGNIFICOS NAVIOS
Extended tags			
anchor australia barca barco bay beach blue boat boot bridge buoy buoyant bw canada canal canon city cloud coast cruise cruiseship dock england eos etsy ferry fish fishing france greece green handmade harbor hdr holiday island italia italy lake landscape light lighthouse mar mare marina maritime mast nature navy night nikon ocean old orange people pier port puerto red reflection retro river rope sail sailboat sailor sand scotland sea seaside ship sky star summer sun sunrise sunset tattoo travel tree uk vacation venice vessel vintage water waves white wood yacht yellow			
Candidate tags		Final tags	
Action anchor australia barca bateau bay beach blue boat boot bridge canada canal canon city cloud coast crew cruise dock england ferry fish fishing france greece green harbor holiday island italia italy lake landscape light lighthouse Mad mar mare marine Mariner maritime mast nature nautical nave navy night nikon ocean orange Passenger People pier port reflection River rope Sad Sail sailboat sailor sand scotland sea seaside Seven ship side sky stAr summer sun sunrise sunset tour Tourist travel tree vacation venice vessel vintage voyage Watch water waves white wood yacht		anchor bay beach blue boat canal cloud coast fish fishing harbor island lake landscape light nature ocean Passenger People Sail sailboat sailor sand scotland sea seaside ship sky sun sunrise sunset tour Tourist travel water waves wood yacht	

Fig. 11. An illustration of a social image tagging example by our approach.

As our method can not only filter noisy tags but also enrich initial tags, in this section we illustrate the results of tag enrichment to verify if our tag extended policy is effective. The performance evaluating before and after enrichment are illustrated in Table 4, particularly, before and after enrichment methods refer to images tagging without and with tagging extended respectively.

Table 4. Performance comparison between two schemes.

Scheme	Precision	Recall	F1
Before Enrichment	0.537	0.436	0.481
After Enrichment	0.652	0.514	0.575

We can know that our tag extended policy could significantly promote quality of social image tagging.

5.2 Experiment 2: Performance Comparing with Other Social Images Tagging Methods

In this experiment, we utilize four computers with Intel Core i7 2.8GHz CPU and 8G RAM to conduct parallel tag graph partitioning process. Moreover, to verify the overall performance of our algorithm, two standard social image datasets are used, such as MIRFlickr-25K and NUS-WIDE-270K.

5.2.1 Evaluation of tagging effectiveness on MIRFlickr-25K

In this subsection, we evaluate the tag effectiveness of our algorithm comparing with existing methods on MIRFlickr-25K dataset [42, 43], which is an image collection consisting of 25000 images. MIRFlickr-25K is collected from Flickr through its public API [42]. To test the performance of our social images tagging method, four other methods are utilized as baselines, such as original user-supplied tags (OUT), RWR [49], TRVSC [44] and LECT [45], furthermore, F1 value is adopted as metric.

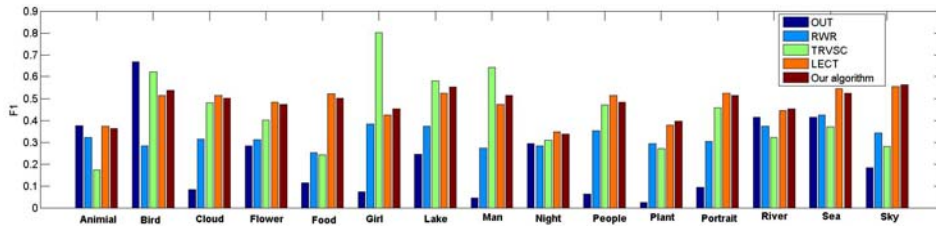


Fig. 12. Performance evaluating on MIRFlickr-25K.

Fig. 12 shows that the performance of our algorithm is much better than OUT, RWR and TRVSC in most cases, and is close to LECT. Tags in OUT are provided by users when uploading photos, and we find the fact that the user-supplied tags in public photo sharing websites are imprecise and incomplete. Therefore, the F1 value of OUT is the worst in five proposed methods. The F1 value of LECT is better than our algorithm, as LECT utilize an efficient iterative approach for image tag refinement through pursuing the low-rank, content consistency, tag correlation and error sparsity. However, the performance of LECT is not satisfied when tagging the social images with large-scale candidate tags.

5.2.2 Evaluation of tagging efficiency on NUS-WIDE-270K

Section 5.2.1 shows tagging effectiveness of our algorithm, nevertheless, we should also verify tagging efficiency on a large-scale dataset. The dataset used in this subsection is NUS-WIDE-270K [47, 48], which is created by Lab for Media Search in National Uni-

versity of Singapore. NUS-WIDE-270K includes 269,648 images and the associated tags from Flickr with a total number of 5,018 unique tags, and there are nearly 25K images in NUS-WIDE-270K without initial tags. As our algorithm can not tag the images without initial tags, in this experiment, images without initial tags are deleted from NUS-WIDE-270K dataset.

As the RWR and TRVSC algorithm are not suitable for large-scale datasets, we only compare the performance of our proposed strategy with OUT and LECT. Tagging results of fifty concept categories in NUS-WIDE-270K are illustrated in Fig. 13.

Fig. 13 shows that our algorithm performs a little better than LECT on NUS-WIDE-270K under F1 metric, and the average F1 value of OUT, LECT and Our algorithm are 0.36, 0.432 and 0.443 respectively. To show the tagging efficiency, we test the time consumption with the increment of initial tags in Fig. 14.

Fig. 14 shows that adopting parallel tag graph partitioning, our algorithm accelerates the process of final tags selecting significantly. From the encouraging results, the conclusion can be drawn that our algorithm can tag large-scale social images effectively.

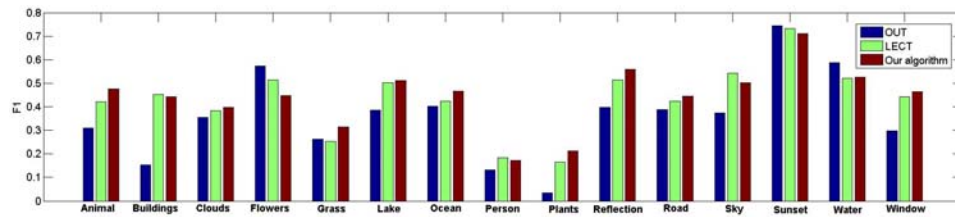


Fig. 13. Performance evaluating on NUS-WIDE-270K.

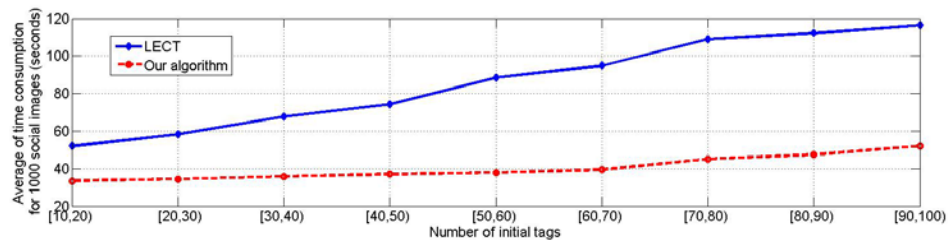


Fig. 14. Time consumption evaluating with number initial tags varying.

5.3 Experiment 3: Performance Evaluating in Tag-Based Image Retrieval

It is widely known that tag-based image retrieval can benefit from image tagging, therefore we conduct experiment 3 to verify the effectiveness of our proposed method in the application of tag-based image retrieval. In this experiment, all tags should be ranked in advance, hence, we use Algorithm 1 of this paper to complete this task (see section 3.3.3). Moreover, experiment 3 uses the same dataset as experiment 1.

5.3.1 Utilizing NDCG as evaluation metric

To rank images, the relevance score of an image is defined in the way of paper [2].

For an input query q , the images which have q as final tags are considered as candidate relevant images. The relevance score of image I_i is defined in the following equation.

$$r(I_i) = -\pi_i + \frac{1}{n_i} \quad (14)$$

where π_i denotes the position of q in the ranking tag list of I_i . If $\pi_i < \pi_j$, $r(I_i) > r(I_j)$ is satisfied. It means that the image which contains the query tag at more advanced positions in its ranked tag list has higher relevance score. On the other hand, if $\pi_i = \pi_j$, the image with fewer tags is set larger relevance score. With the relevance score calculated in Eq. (14), we rank image search results by the scores in descending order. Afterwards, NDCG is utilized as evaluation metric. Four levels are used to make relevance evaluation: Most Relevant (score 4), Relevant (score 3), Weakly Relevant (score 2) and Irrelevant (score 1). As is shown in Fig. 15, the following methods are used to compare with our algorithm: (1) interestingness-based ranking; (2) uploading time-based ranking. The two approaches are the services provided by Flickr, which could provide image ranking lists by interestingness and uploading time records.

5.3.2 Utilizing MAP as evaluation metric

In this subsection, we use another standard metric MAP (Mean Average Precision) [46] to conduct performance evaluating.

$$MAP(\pi^*, \pi) = \frac{1}{rel} \sum_{j: \pi_j^*} P@j \quad (15)$$

where π^* is the ground truth ranking and π is the ranking results computed by Eq. (14). $rel = |\{i: \pi_i^* = 1\}|$ is the number of relevant images and $P@j$ is the percentage relevant images in top j images.

In Figs. 15 and 16, we illustrate the performance of our algorithm on social images retrieval. Experimental results show that compared with two standard image ranking services of Flickr, our algorithm can significantly promote the accuracy of social images ranking, and then remarkably improve tag-based social image retrieval according to relevance.

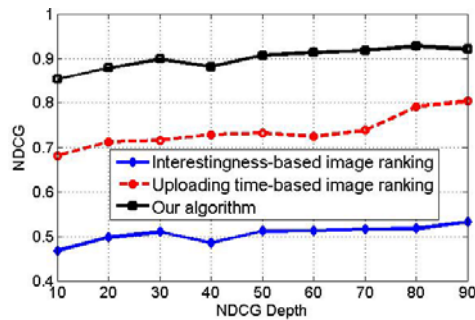


Fig. 15. Performance evaluating on NDCG metric.

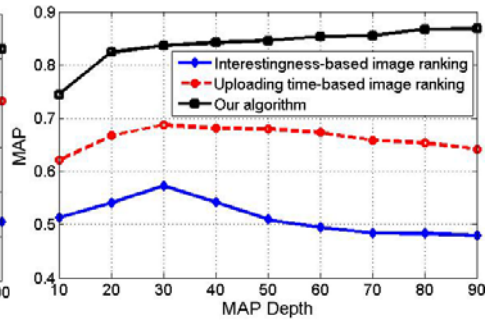


Fig. 16. Performance evaluating on MAP metric.

6. CONCLUSIONS AND FUTURE DIRECTIONS

This paper proposes a novel approach to tag social images by partitioning the large-scale tag graph in parallel. After the tag graph is partitioned, all the sub-graphs are ranked according to the edge weight. Afterwards, final tags are selected from the top ranked sub-graphs. Finally, experiments are conducted on Flickr photos dataset, from the experimental results we can see that our approach is of high effectiveness and efficiency.

However, our approach is not suitable for all social images. For the image with little metadata, the performance our approach can not make us satisfactory. The reason lies in that if the number of initial tags is not large enough, tagging extending process can not add useful semantic information effectively.

In the future, we would extend our works in the following aspects. (1) We will try to test our approach in other social image community. (2) Other edge weight estimating policy will be adopted in tag graph constructing. (3) Although the tag graph partitioning speed is quite fast, we should try to accelerate the tag graph constructing process.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant No. 61303090, No. 61272431, No. 61020106001, No. 61101162, No. 60970048, and No. 60903109), NSFC Joint Fund with Guangdong (U1201258), the Scientific Research Foundation for the Excellent Middle-Aged and Youth Scientists of Shandong Province (Grant No. BS2012DX028, BS2011DX024, and No. BS2013DX013), Shandong Provincial Natural Science Foundation (Grant No. ZR2012FM002, ZR2011FL020, ZR2011FL029, and ZR2013FL010), the Humanities and Social Sciences Project of Education Ministry (Grant No. 13YJC860023), the Doctoral Foundation of Shandong University of Finance and Economics (Grant No. B13034), Project of Shandong Province Higher Educational Science and Technology Program (Grant No. J10LG69), the Science and Technology Project of Ji'nan (Grant No. 201303012) and Program for Scientific Research Innovation Team in Colleges and Universities of Shandong Province. We also appreciate the anonymous reviewers for thoroughly reading the paper and providing thoughtful comments.

REFERENCES

1. H. Xu, X. Zhou, M. Wang, *et al.*, "Exploring Flickr's related tags for semantic annotation of web images," in *Proceeding of the ACM CIVR*, 2009, pp. 352-359.
2. D. Liu, X. S. Hua, L. Yang, *et al.*, "Tag ranking," in *Proceedings of WWW*, 2009, pp. 351-360.
3. L. Wu, X.-S. Hua, N. Yu, *et al.*, "Flickr distance," in *Proceedings of ACM MM*, 2008, pp. 31-40.
4. P. Schmitz, "Inducing ontology from flickr tags," in *Proceedings of WWW*, 2006, pp. 210-214.
5. M. Chen, M. H. Chang, P. C. Chang, *et al.*, "SheepDog-group and tag recommendation for flickr photos by automatic search-based learning," in *Proceedings of ACM MM*, 2008, pp. 737-740.

6. L. Wu, L. J. Yang, N. H. Yu, and X. S. Hua, "Learning to tag," in *Proceedings of WWW*, 2009, pp. 361-370.
7. M. Ames and M. Naaman, "Why we tag: Motivations for annotation in mobile and online media," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2007, pp. 971-980.
8. L. S. Kennedy, S. F. Chang, and I. V. Kozintsev, "To search or to label? Predicting the performance of search-based automatic image classifiers," in *Proceedings of the ACM MIR*, 2006, pp. 249-258.
9. S. Si, D. Tao, M. Wang, and K. P. Chan, "Social image annotation via cross-domain subspace learning," *Multimedia Tools and Applications*, Vol. 56, 2012, pp. 91-108.
10. N. Zhou, J. Peng, X. Feng, and J. Fan, "Towards more precise social image-tag alignment," in *Proceedings of MMM*, 2011, pp. 46-56.
11. P. Wu, S. C.-H. Hoi, P. Zhao, and Y. He, "Mining social images with distance metric learning for automated image tagging," in *Proceedings of ACM WSDM*, 2011, pp. 197-206.
12. J. Tang, Z.-J. Zha, D. Tao, and T.-S. Chua, "Semantic-gap-oriented active learning for multilabel image annotation," *IEEE Transactions on Image Processing*, Vol. 21, 2012, pp. 2354-2360.
13. Z. Liu, W. Zhao, H. Han, and H. Yan, "Tagging social images by probabilistic topic model and tag association mining," *Information-An International Interdisciplinary Journal*, Vol. 15, 2012, pp. 249-258.
14. J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma, "Image annotation via graph learning," *Pattern Recognition*, Vol. 42, 2009, pp. 218-228.
15. X. Rui, M. Li, Z. Li, W.-Y. Ma, and N. Yu, "Bipartite graph reinforcement model for web image annotation," in *Proceedings of the ACM MM*, 2007, pp. 585-594.
16. Y. Jin, K. Jin, L. Khan, and B. Prabhakaran, "The randomized approximating graph algorithm for image annotation refinement problem," in *Proceedings of CVPR*, 2008, pp. 1-8.
17. Y. Jin, L. Khan, and B. Prabhakaran, "Knowledge based image annotation refinement," *Signal Processing Systems*, Vol. 58, 2010, pp. 387-406.
18. Z. Liu and J. Ma, "Refining image annotation by graph partition and image search engine," *Journal of Computer Research and Development*, Vol. 48, 2011, pp. 1246-1254.
19. B. Xie, Y. Mu, D. Tao, and K. Huang, "m-SNE: Multiview stochastic neighbor embedding," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 41, 2011, pp. 1088-1096.
20. N. Guan, D. Tao, Z. Luo, and B. Yuan, "Non-negative patch alignment framework," *IEEE Transactions on Neural Networks*, Vol. 22, 2011, pp. 1218-1230.
21. T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 40, 2010, pp. 1438-1446.
22. N. Guan, D. Tao, Z. Luo, and B. Yuan, "Manifold regularized discriminative non-negative matrix factorization with fast gradient descent," *IEEE Transactions on Image Processing*, Vol. 20, 2011, pp. 2030-2048.
23. N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 23, 2012, pp. 1087-1099.

24. N. Guan, D. Tao, Z. Luo, and B. Yuan, "Nenmf: An optimal gradient method for non-negative matrix factorization," *IEEE Transactions on Signal Processing*, Vol. 60, 2012, pp. 2882-2898.
25. X. Tian, D. Tao, and Y. Rui, "Sparse transfer learning for interactive video search reranking," *ACM Transactions on Multimedia Computing, Communications and Applications*, 2012, arXiv:1103.2756v3, to be published.
26. X. Tian, D. Tao, X. S. Hua, and X. Wu, "Active reranking for web image search," *IEEE Transactions on Image Processing*, Vol. 19, 2010, pp. 805-820.
27. R. A. Negroescu and D. Gatica-Perez, "Analyzing Flickr groups," in *Proceedings of ACM CIVR*, 2008, pp. 417-426.
28. L. Zhang, J. Ma, C. Cui, and P. Li, "Active learning through notes data in Flickr: An effortless training data acquisition approach for object localization," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, 2011, Article No. 46.
29. B. Sigurbjörnsson and R. van Zwol, "Flickr tag recommendation based on collective knowledge," in *Proceedings of WWW*, 2008, pp. 327-336.
30. R. L. Cilibrasi and P. M.-B. Vitányi, "The Google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, 2007, pp. 370-383.
31. T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Transactions on Image Processing*, Vol. 2, 1993, pp. 429-441.
32. D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, 2004, pp. 91-110.
33. P. Duygulu, K. Barnard, J. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Proceedings of ECCV*, 2002, pp. 97-112.
34. Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, Vol. 28, 1980, pp. 84-94.
35. X. Li, C. G. M. Snoek, and M. Worring, "Learning social tag relevance by neighbor voting," *IEEE Transactions on Multimedia*, Vol. 11, 2009, pp. 1310-1322.
36. R. Banos, C. Gil, J. Ortega, and F. G. Montoya, "A parallel multilevel metaheuristic for graph partitioning," *Journal of Heuristics*, Vol. 10, 2004, pp. 315-336.
37. <http://www.glaros.dtc.umn.edu/gkhome/metis/parmetis/download>.
38. G. Karypis and V. Kumar, "Multilevel algorithms for multi-constraint graph partitioning," in *Proceedings of the ACM/IEEE Conference on Supercomputing*, 1998, pp. 1-13.
39. G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme for irregular graphs," *Journal of Parallel and Distributed Computing*, Vol. 48, 1998, pp. 96-129.
40. G. Karypis and V. Kumar, "A coarse-grain parallel multilevel k-way partitioning algorithm," in *Proceedings of the 8th SIAM Conference on Parallel Processing for Scientific Computing*, 1997, pp. 1-12.
41. K. Schloegel, G. Karypis, and V. Kumar, "Parallel multilevel algorithms for multi-constraint graph partitioning," in *Proceedings of EuroPar*, 2000, pp. 296-310.
42. M. Huiskes and M. Lew, "The MIR Flickr retrieval evaluation," in *Proceedings of the ACM MIR*, 2008, pp. 39-43.
43. <http://press.liacs.nl/mirflickr/dlform.php>.
44. D. Liu, X. S. Hua, M. Wang, and H. J. Zhang, "Image retagging," in *Proceedings of*

ACM MM, 2010, pp. 491-500.

45. G. Zhu, S. Yan, and Y. Ma, "Image tag refinement towards low-rank, content-tag prior and error sparsity," in *Proceedings of ACM MM*, 2010, pp. 461-470.
46. J. Zhuang and S. C. H. Hoi, "A two-view learning approach for image tag ranking," in *Proceedings of the ACM WSDM*, 2011, pp. 625-634.
47. 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 *Proceedings of ACM CIVR*, 2009, pp. 368-375.
48. <http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm>.
49. C. Wang, F. Jing, L. Zhang, and H. J. Zhang, "Content-based image annotation refinement," in *Proceedings of IEEE CVPR*, 2007, pp. 1-8.



Zheng Liu (劉曄) born in 1980, earned a B.S. and M.S. degree in Computer Science and Technology from Shandong University, in 2002 and 2005 respectively. After graduate school, he joined school of Computer Science and Technology, Shandong University of Finance and Economics in 2005, and received his Ph.D. in Computer Science from Shandong University in 2011. His main research interests include machine learning, Web mining and information retrieval.



Huijian Han (韓慧健) born in 1971, received his Ph.D. degree in Computer Applied Technique from Shandong University. Han's major field of study is texture mapping of CG. He is a Professor, Master Tutor in School of Computer Science and Technology of Shandong University of Finance and Economics and Shandong Provincial Key Lab of Digital Media Technology.



Hua Yan (閻華) born in 1973, received the Ph.D. degree in Communication and Information System from Shandong University, Jinan, China, in 2007. Now she is an Associate Professor in Shandong University of Finance and Economics. Her current research interests include various aspects of image/video processing and its application.