Multimed Tools Appl (2013) 62:601–631
DOI 10.1007/s11042-011-0863-5

Improving image tags by exploiting web search results

Xiaoming Zhang · Zhoujun Li · Wenhan Chao

Published online: 27 August 2011

Abstract Automatic image tagging automatically assigns image with semantic keywords called tags, which significantly facilitates image search and organization. Most of present image tagging approaches are constrained by the training model learned from the training dataset, and moreover they have no exploitation on other type of web resource (e.g., web text documents). In this paper, we proposed a search based image tagging algorithm (CTSTag), in which the result tags are derived from web search result. Specifically, it assigns the query image with a more comprehensive tag set derived from both web images and web text documents. First, a content-based image search technology is used to retrieve a set of visually similar images which are ranked by the semantic consistency values. Then, a set of relevant tags are derived from these top ranked images as the initial tag set. Second, a text-based search is used to retrieve other relevant web resources by using the initial tag set as the query. After the denoising process, the initial tag set is expanded with other tags mined from the text-based search result. Then, an probability flow measure method is proposed to estimate the probabilities of the expanded tags. Finally, all the tags are refined using the Random Walk with Restart (RWR) method and the top ones are assigned to the query images. Experiments on NUS-WIDE dataset show not only the performance of the proposed algorithm but also the advantage of image retrieval and organization based on the result tags.

X. M. Zhang (✉) · Z. J. Li · W. H. Chao
State Key Laboratory of Software Development Environment,
Beihang University, Beijing 100191, China
e-mail: yolixs@163.com

X. M. Zhang · Z. J. Li · W. H. Chao
School of Computer Science and Engineering, Beihang University,
Beijing 100191, China

X. M. Zhang · Z. J. Li · W. H. Chao
Beijing Key Laboratory of Network Technology, Beihang University,
Beijing 100191, China

Keywords Image tagging · Search based tagging · Tag expansion · Image retrieval

1 Introduction

With the popularity of digital cameras and the Internet, there has been a rapid growth in the number of digital image appearing in the web (e.g., Flickr, Picasa), which requires an effective image search technology. Unlike web document which can be indexed by their keywords, image are 2D media and how to define image "keywords" is still an open question. A possible solution is to index and search images with visual information, i.e., content based image retrieval (CBIR) [24, 35]. Although CBIR has been researched for more than a decade, it still has several bottlenecks. First, due to the semantic gap problem [26], current visual feature extraction techniques are not effective enough in representing the semantic of an image. The second problem is computational expensiveness. Due to the high dimensionality of visual features, the efficiency and scalability of CBIR are usually very low. These problems are further exacerbated in the social media environment because of its huge volumes and the noise generated by users.

To solve these problems, there has been a surge of research in image tagging recently. With the tags, images can be searched like web documents. The systematic experimental results in [29] indicate that social tagged images can aid concept-based video search indeed. There are already some existing web sites which allow users to upload personal images and also tag them manually. However, manually tagging images is intellectually expensive and time consuming. Moreover, individual or community provided tags lack consistency and present numerous irregularities [14] (e.g. abbreviations and mistypes).

Thus, many automatic tagging approaches have been proposed recently [7, 12, 23, 30, 31, 36, 41]. However, most of these tagging approaches have a high dependency on their training dataset, which restrict their scalability and effectiveness on arbitrary dataset. It is also hard for the training-based approaches to update their training dataset, which result in that it is hard for they to scale to online tagging. Furthermore, some fashionable tags which appear in the web recently can't be assigned to the query images because that they don't appear in their tag vocabulary. The second serious problem is that each query image is only assigned with the tags which appear in its visually similar images. Since the tags of most images tend to be sparse and also the semantically similar images may be visually diverse images, some relevant tags which don't appear in the visually similar images can't be assigned to the query image. Moreover, the assigned tag can be combined with a numerical value to indicate its relevance to the query image. For example, tags such as "Steven Paul Jobs" and "Stanford" and others can be assigned to the query image about "apple" computer with small relevant values, though these tags may not appear in its visually similar images.

Compared with the limited number of images and tags that can be used to tag the query images, the potentially unlimited vocabulary of the web scale images and other types of web resource e.g., web documents can be searched to tag images. By using web-scale resources, the assigned tags of the query image can be more comprehensive and also can track the fashionable expression on the Web. Thus, using a search-based approach to utilize the web-scale resource to tag image has attracted great attention. There is already some research on search based image tagging [22, 37–39]. However, these approaches still tag the query image with tags derived from the visually similar images only. Due to the semantic gap, there are many relevant tags which don't appear in the visually similar images. Moreover, other types of web resources such as web document can also be useful for image tagging. However, these approaches have no consideration of these problems.

In this paper, we propose a new image tagging algorithm (CTSTag) based on web search result mining. We assign the query image with a more comprehensive tag set mined from not only web images but also other relevant web text documents. A set of relevant tags initially derived from visually similar images retrieved by CBIR is used as a query to perform text-based search. Thus, other relevant web documents are retrieved by the text-based search, and what is more, other tags which don't appear in the visually similar images can also be retrieved. Then, the initial tag set is improved by mining the text-based search result. We also combine the assigned tag with a probability value which indicate its relevance to the query image. With the search-based framework, our approach is more scalable.

The rest part of this paper is organized as follows. Related works are introduced in Section 2, and the algorithm overview is given in Section 3. Then, the process of deriving initial relevant tags from the content-based search results is presented in Section 4. The improving of initial tag wset based on text-based search results mining is introduced in Section 5. In Section 6, we refine all the tags using a probability propagation method. Extensive experimental results are reported in Section 7 followed by the conclusion in Section 8.

2 Related works

Image tagging has attracted more and more interests in recent years. Some works use a learning or classification method to tag the query images. For example, Barnard et al. [2] develop a number of models for the joint distribution of image regions and words to tag the query images. Lei et al. [18] propose a probabilistic distance metric learning scheme that automatically derives constraints from the uncertain side information. Then, the distance metric is used to retrieve a set of visually similar images from which a set of tags are derived to tag the query image. Bailloeul et al. [1] present a random walk graph-based scheme based on the GCap method to perform automatic image tagging. It uses the canonical correlation analysis technique (CCA) to shorten the semantic gap in the image space and defines a new metric in the text space to correlate tags with content. Geng et al. [10] model the concept affinity as a prior knowledge into the joint learning of multiple concept detectors, and then the concept detectors which are classifiers are used to tag the query images. Li et al. [19] combine statistical modeling and optimization techniques to train hundreds of semantic concepts using example pictures from each concept. For each query image, a list of probabilities for the image being in each concept is estimated, Then, the top ranked concepts are selected. Chen et al. [7] propose a multi-label propagation based annotation approach. Chang et al. [6] propose a method to learn a weighted SVM for binary concept classification on TRECVID images. Li et al. [20] study to what extent social tagging substitutes expert labeling for creating negative examples which are used in automatic visual categorization. Tsikrika et al. [33] build concept classifiers that use automatically acquired labelled samples i.e., the clickthrough data logged by retrieval systems as training data. Wang et al. [40] propose a neighborhood similarity measure for video annotation, which explores the local sample and label distributions. Hong et al. [13] propose a dynamic captioning method to enhance the accessibility of videos for hearing impairment, which is able to help hearing impaired users match the scripts with the corresponding characters. The LabelMe [28] provides a database and an online annotation tool that allows the sharing of images and annotations. It annotate image based on object class recognition as opposed to instance recognition. However, most of these methods try to learning a mapping between low-level visual features and high-level semantic concepts, which are not scalable to cover the potentially unlimited array of concepts existing in social tagging. Moreover, uncontrolled visual content generated by users creates a broad domain environment which has a significant diversity in visual appearance, even for the same concept.

Some other works are mainly based on example-based approach which assumes that visually similar images are also annotated by similar tags. For example, Wang et al. [36] use RWR to automatically refine the original tags of images, and only the top ones are reserved as the final tags. Siersdorfer et al. [31] propose an automatic video tagging method based on video duplicate and overlap detection. The assigned tags are derived from the videos which overlap with the query video. Zhou et al. [42] propose a iterative annotation algorithm which incorporate the keyword correlations and the region matching. The approach in [25] estimates initial relevance scores for the tags of each image based on probability density estimation, and then RWR is applied to refine the relevance scores. Lei et al. [17] propose a multi-modality recommendation model based on both tag and visual content correlation. Rankboost algorithm is then applied to learn an optimal combination of those ranked features of different modalities. Li et al. [21] propose a neighbour voting tagging method, in which the tags which appear in neighbours frequently are selected to tag the query image.

A main problem of these approaches is that they have a high dependency on a small-scale high quality training set with a limit tag vocabulary, which means that they can only tag the query images which have visually similar neighbors in the training set. Recently, there has been some research which tries to leverage web-scale data for image annotation. These approaches use a web search method to annotate images [37–39]. For example, Wang el at. [39] use text-based search to retrieve a set of semantically similar web images, and then the Search Result Clustering (SRC) [15] algorithm is used to cluster these images. Finally, the name of each cluster is used to annotate the query image. However, this approach needs initial keywords for each query image. Wang et al. [38] use the CBIR technology to retrieve a set of visually similar from the large-scale Web image dataset, then annotations of each web image are ranked. Finally, the candidate annotations are re-ranked using Random Walk with Restarts (RWR) and only the top ones are reserved as the final annotations. However, these approaches just use a search based method to retrieve visually similar images from a large-scale web database instead of training dataset, and then derive tags from these retrieved images. They have no consideration of other relevant tags which don't appear in the visually similar images or appear in web documents.

3 Algorithm overview

Unlike other algorithms which only use the visual information extracted from the query image to find similar images and associated tags, we propose to also mining other web resources in connection with tags initially derived from visually similar images. Particularly, we try to overcome the problem that only the training images are exploited to tag image by using other sources of knowledge with the search-based framework, notably both tags from visually similar image and new tag candidates mined from textual searches in text documents. The process of our algorithm is described in Fig. 1, and the pseudo code is detailed in Fig. 2. It is mainly composed of 3 steps.

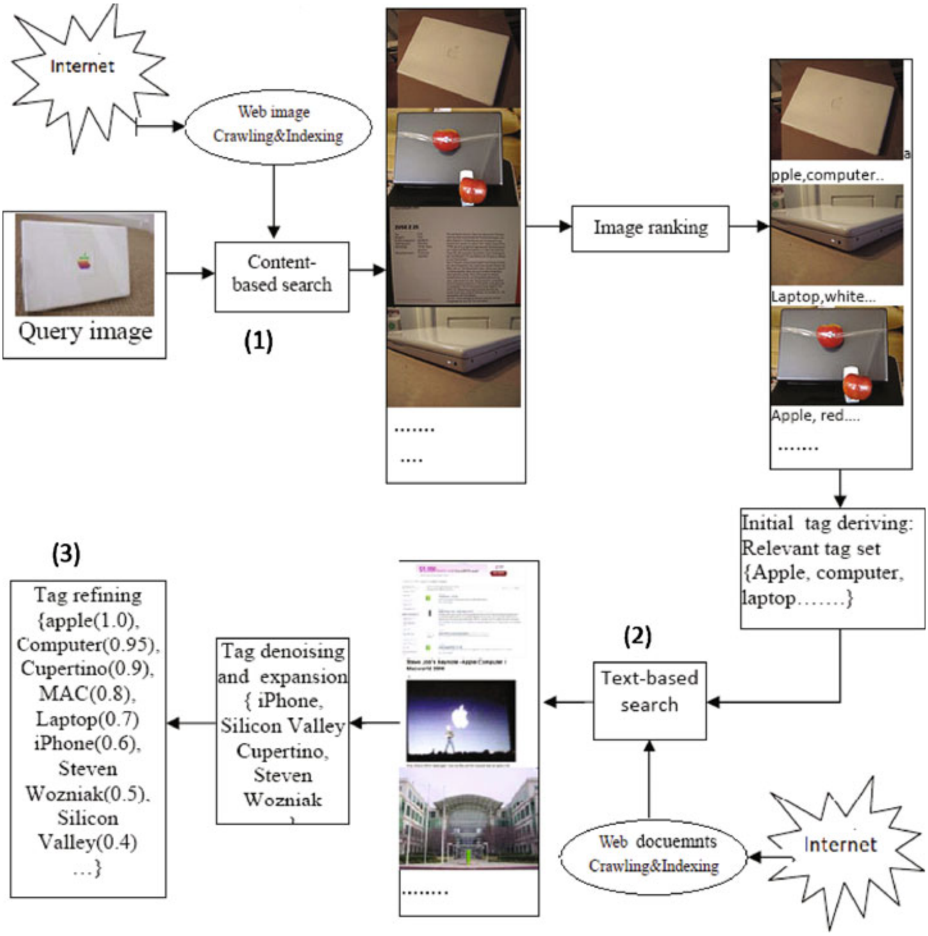

Internet
Web image Crawling&Indexing
Query image
Content-based search
(1)
....
....
Image ranking
apple,computer..
Laptop,white...
Apple, red....
....
Initial tag deriving: Relevant tag set {Apple, computer, laptop.......}
(2)
Text-based search
Web docuemnts Crawling&Indexing
Internet
....
Tag denoising and expansion { iPhone, Silicon Valley Cupertino, Steven Wozniak }
(3)
Tag refining {apple(1.0), Computer(0.95), Cupertino(0.9), MAC(0.8), Laptop(0.7) iPhone(0.6), Steven Wozniak(0.5), Silicon Valley(0.4) ...}

Fig. 1 The flowchart of probabilistic image tagging based on web search. It contains 3 steps: (*1*) initial relevant tags derivation from content-based search results, (*2*) Tags improving based on text-based search results, (*3*) tags refinement

Fig. 2 Pseudo code of the proposed algorithm

```
Input: query image $Iq$.
Output: a set of tags with their probabilities.
Procedure: Image tagging
1.  Image set $S = Search_{CBIR}(Iq)$;
2.  for each image $I_i \in S$
3.      Computing $Consistsem(I_i)$;
4.  Ranking ($S$) by $Consistsem(I_i)$;
5.  $S'$ = top-$K$ images of $S$;
6.  $T\_list$ = tags associated with $S'$;
7.  for each tag $t_i \in T\_list$
8.      Computing $P(t_i|t_q)$
9.  Ranking ($T\_list$) by $P(t_i|Iq)$;
10. $Q$ = top ranked tags of $T\_list$;
11. $F = Search_{text}(Q)$
12. $Q = Denoising(Q, F)$;
13. $Te = Tag\_exp(Q, F)$;
14. $Prob\_flow(Q, Te)$;
15. $R$ = Refine ($Q \cup Te$);
16. Return top tags of $R$.
```

1. Given a query image I_q, we retrieve a set of visually similar web images S using CBIR from a large-scale Web image set (line 1). All the retrieved images are ranked based on their semantic consistency scores, and only the top-K images are selected (lines 2–5, Section 4.1). Then, the most probabilistic tags are derived from the tags associated with the top-K images. These derived tags constitute the initial tag set Q (lines 6–10, Section 4.2). For example, for the query image I_q of apple computer, a set of initial tags Q={"apple", "computer", " laptop",...} can be derived from the content-based search results.
2. Text-based search is used to retrieve other web images and text documents with the initial relevant tag set Q as the query (line 11). As a result, other potentially relevant tags which don't appear in the visually similar images can be retrieved. Based on the occurrence distribution among the search results, the initial tag set is first denoised (line 12, Section 5.1). Then it is expanded with the terms mined from the search result(line 13, Section 5.2). The probability of expanded tag is estimated by measuring the probability flow which flow into the expanded tag from the initial tag set (line 14, Section 5.3). By mining expanded tags from the text-based search result, the assigned tags are more comprehensive and can also absorb the new tags from other resources. For example, using the text-based search, other web images such as apple headquarter "Silicon Valley" and document about "Steve Wozniak" the co-founder of apple computer can also be retrieved. Thus the initial tag set is expand with {"Silicon Valley", "Steve Wozniak",...}.
3. Finally, a measure of transition strength by mining tag correlation is proposed to construct the tag transition graph. Then, all the tags are refined using the RWR method based on this tag transition graph (line 15, Section 6.2), and the top tags are reserved (line 16).

The improvement of our algorithm over other baseline algorithms named NVTag [21] and SBIA [38] introduced in Section 2 are described in the following. Though SBIA use a search-based method to overcome the problem of scalability, both of the baseline algorithms only assign the query image with the common tags derived from visually similar images. However, we try to overcome the problem of small training datasets by using other sources of knowledge with the search-based framework. We assign the query tags with tags from visually similar image documents, and, what is the main novelty, new tag candidates from textual searches in text documents.

4 Initial tags

One main idea of example-based tagging approaches is that visually similar images contain similar tags. Thus, we also derive the initial tag set Q for the query image I_q from its visually similar images. We use CBIR to retrieve a set of visually similar images S from the large-scale web database, and the top images with the greatest semantic consistency values are selected from S. Then, the initial tags which are relevant to the query image are selected from the tags associated with these selected images.

4.1 Semantic consistency of image

Due to the semantic gap problem [26], usually there are some visually similar images which have large semantic divergence with the query image I_q in the search result S of CBIR. As shown in Fig. 3, the second retrieved image which are visually similar but are semantically different with the query image. We expect to select the visually similar images that are also semantically similar with the query image. In reality, web images always have rich textual information, such as tags, comments and photographer's description. This textual information reflects the semantic mean of the corresponding image to some extent. In this paper, we introduce "semantic consistency" which is used to measure in what degree the retrieved image is semantically similar with the query image. It is used to select the retrieved images which are more semantically similar with the query image.

We estimate the semantic consistency value of a retrieved image based on both of its semantic and visual similarity with other images in the search result. The intuition is that the retrieved images which are semantically similar with the query image should be semantically similar with each other, e.g., the first and third retrieved images in Fig. 3, while a noisy retrieved image will be semantically diverse with most of other retrieved images, e.g., the second retrieved image in Fig. 3. For each image I_i in S we calculate its visual similarity $Sim_{\text{visual}}(I_i, I_q)$ to the query image. Meanwhile, the textual similarity $Sim_{\text{text}}(I_i, I_j)$ between each two images I_i and I_j in S is measured by the the cosine similarity between the vectors of their tags and terms extracted from surrounding text. We use $\vec{v}$ to represent textual features and $\vec{x}$ to represent visual features. The semantic consistency value of image I_i with respect

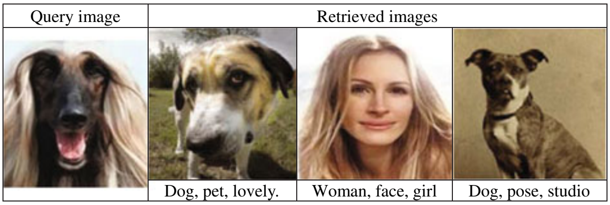


Fig. 3 Example of visually similar images

to I_q is calculated as following:

$$Consist_{\text{sem}}(I_i) = \frac{1}{|S|}\sum_{I_j \in S} Sim_{\text{text}}(I_i, I_j) Sim_{\text{visual}}(I_j, I_q)$$

$$Sim_{\text{text}}(I_i, I_j) = \frac{\overrightarrow{v_i} * \overrightarrow{v_j}}{|\overrightarrow{v_i}| \times |\overrightarrow{v_j}|}$$

$$Sim_{\text{visual}}(I_i, I_j) = exp\left(-\frac{||\overrightarrow{x_i} - \overrightarrow{x_j}||^2}{\sigma^2}\right) \tag{1}$$

Then, we select the top-K images with the greatest $Consist_{\text{sem}}(.)$ value, and the initial tags are derived from the tags associated with these images. Unlike our approach, SBIA and NVTag derive tags from the most visually similar images directly. This calculation of semantic consistency value is similar with that proposed in [38]. However, The consistency value described in [38] is independent of the query image and its number of visually similar images.

4.2 Probabilities of initial tags

For each query image, it is desired to assign it with the most relevant tags which can describe its semantic content. We use the probability that the query image generating the tag to represent the relevance. The greater the probability is the more relevant it is.

From the generation point of view, the conditional probability of tag t_i generated from the query image I_q can be approximate by the joint generating probability of every similar image of I_q. Then, the probability $P(t_i|I_q)$ which interprets the relevance of t_i to the query image is estimated using the following formula:

$$P(t_i|I_q) \approx \frac{N_k(t_i)}{K} P_G(t_i|I_q) + \frac{K - N_k(t_i)}{K} P_L(t_i|I_q) \tag{2}$$

where $N_k(t_i)$ represents the number of images which contain the tag t_i among the top-K images. This estimation includes two parts which are balanced by the parameter. The intuition of the first part $P_G(t_i|I_q)$ is that the more similar images contain the tag and the more frequently the tag appears, then the more probable is for the query image to be tagged with this tag. The intuition behind the second part is that the more similar the images which contain the tag are, the more probable is for the tag to be assigned. We use the following formula to estimate $P_G(t_i|I_q)$:

$$P_G(t_i|I_q) = \sum_{I_j \in \text{top}-K} P(t_i|I_j) P(I_j|I_q)$$

$$P(t_i|I_j) = \begin{cases} 1 \ t_i \in I_j \\ 0 \ \text{else} \end{cases}$$

$$P(I_j|I_q) = \frac{Sim_{\text{visual}}(I_j, I_q)}{K} \tag{3}$$

where $Sim_{\text{visual}}(I_j, I_q)$ represents the visual similarity between I_j and I_q. However, this formula will prefer the tags which appear frequently among the top-K images. Thus, we use the following formula to alleviate this problem:

$$P_L(t_i|I_q) = \frac{\sum\limits_{I_x \in I_m(t_i)} Sim_{\text{visual}}(I_x, I_q)}{|I_m(t_i)|} \tag{4}$$

where $I_m(t_i)$ denotes the set of images which contain tag t_i within the database. In realization, we use a subset of images which are the nearest neighbors to the central of $I_m(t_i)$ to approximate $I_m(t_i)$. Since $0 \leq P_G(t_i|I_q) \leq 1$ and $0 \leq P_L(t_i|I_q) \leq 1$, it is also true that $0 \leq P(t_i|I_q) \leq 1$. We can see that formula (2) can boost the tags which appear less frequently but are relevant to the query image. Thus, it make a balance between the tags which appear frequently and the tags which appear less frequently. Then, all the tags associated with the top-K images are ranked according to their probability values, and only the top tags are selected to compose the initial tag set Q.

5 Tags improving

Since there may be many relevant tags which don't appear in the visually similar images, and also there may be many new keywords which haven't been used to tag images. We perform a text-based search to retrieve other relevant web resources by using the initial tag set as the query. Then the initial tag set can be denoised and expanded by mining the search result denoted by F. Among varied retrieval models, the Vector Space Model is used [34]. With the text-based search, not only the web images but also the web text documents can be retrieved to tag the query image. Thus, the result tag set can be more comprehensive.

5.1 Tags denoising

Because of the semantic gap problem, the content based image search usually retrieves many noisy images. Thus it is inevitable to include many noisy tags in the initial tag set. A direct expansion of the initial tag set [4] may include many other noisy tags. Thus, it needs to further denoise the initial tags set before it is expanded. However, it is difficult to distinguish the noisy tags only using visual content due to the semantic gap problem. In this subsection, we use the tag occurrence distribution among the text-based search results to further denoise the initial tag set.

By regarding each initial tag as a document "containing" the associated web pages, then the denoising of initial tag set can be turn to removing the noisy "documents". To represent a tag t_i, a vector $\overrightarrow{t_i} = < tf(t_i, D_1), tf(t_i, D_2), ..., tf(t_i, D_{|F|}) >$ is constructed, where $tf(t_i, D_j)$ is the term frequency of tag t_i appearing in the retrieved web page D_j. As it is shown, the vector interprets the tag occurrence distribution among the retrieved web pages. For the noisy tags, its occurrence distribution among the retrieved web pages is different with the relevant tags and other noisy tags, which means its vector may be different with that of most of other tags. While, a relevant tag co-occur with other relevant tags frequently, which means its vector representation is similar with that of other relevant tags. Thus, the more similar a vector is with other vectors, then the more confident is for the corresponding tag to be a relevant tag.

We use the following formula to estimate the confidence score that a tag t_i is a relevant tag:

$$Con(t_i) = \frac{1}{|Q|} \sum_{t_i \in Q} Sim_{\text{text}}(\overrightarrow{t_i}, \overrightarrow{t_j}) \tag{5}$$

where $Sim_{\text{text}}(\overrightarrow{t_i}, \overrightarrow{t_j})$ is the cosine similarity of t_i and t_j. To denoise the initial tag set, we use a simple filtering method which only remove the tags whose confidence score is below a threshold value θ from the initial tags set Q. After the denoising process, the initial tags set Q is then expanded. According to the observation, the confidence socres of noisy tags are evidently small comparing to that of relevant tags in most case. Then, the average value of initial tags's confidence scores can separate most of the noisy tags and relevant tags. Thus, we set the threshold value θ with the average value for simplification:

$$\theta = \frac{1}{|Q|} \sum_{t_i \in Q} Con(t_i) \tag{6}$$

5.2 Tag expanding

As the text-based search result contains many web pages and terms which are keywords or tags, a direct tag expansion from the text search result will be very complex. Thus, a ranking method is used to select a subset of important web pages from which the important terms are extracted. Then the potentially relevant tags are selected from these terms to expand the initial tag set. The conditional probability $P(Q|D)$, i.e., the probability of generating the query Q given the observation of a web page D [9] is used to rank the retrieved web pages in a descending order. This Q is the initial tag set after the denoising process. We use the general unigram model to formulate $P(Q|D)$ as following:

$$\begin{aligned} P(Q|D) &\approx \sum_{t_i \in Q} P(Q|t_i) P(t_i|D) \\ &= \sum_{t_i \in Q} \frac{P(t_i|Q) P(Q)}{P(t_i)} P(t_i|D) \end{aligned} \tag{7}$$

where the prior probability $P(Q)$ is constant, and $P(t_i|Q)$ is calculated using formula (2) by replacing I_q with Q. As we only use $P(Q|D)$ to approximately rank web pages, we omit $P(Q)$ from the formula for simplification:

$$\begin{aligned} P(Q|D) &\approx \sum_{t_i \in Q} \frac{P(t_i|Q) P(t_i|D)}{P(t_i)} \\ P(t_i|D) &= \frac{|D|}{|D|+\mu} P_{LM}(t_i|D) + \frac{\mu}{|D|+\mu} P_{LM}(t_i|\mathcal{C}) \\ P_{LM}(t_i|D) &= \frac{tf(t_i, D)}{|D|}, \; P_{LM}(t_i|\mathcal{C}) = \frac{tf(t_i, \mathcal{C})}{|\mathcal{C}|} \end{aligned} \tag{8}$$

where $P_{LM}(t_i|D)$ is the maximum likelihood estimation of t_i in D, and $\mathcal{C}$ is the collection which is approximated by the search result in this paper, and μ is the smoothing parameter and we set it with the average length of web page. $tf(t_i, D)$ and $tf(t_i, \mathcal{C})$ are the frequency of t_i in D and $\mathcal{C}$ respectively.

After ranking we extract terms from the top pages. If the page contains tags, we only extract the tags. Otherwise, we extract the keywords from the textual content of the page. Each term is represented as a keyword after the stemming process. The following method is used to assess the weight of term f in a page D:

$$w_D(f) = \log\left(\frac{tf(f, D)}{|D|} + 1\right) * \log\left(\frac{|Q|}{n_t(f, Q)} + 1\right) \tag{9}$$

where $tf(f, D)$ is the term frequency of f in D, and $n_t(f, Q)$ is the number of initial tags that f has co-occurred with. The addition of 1 is used to avoid zero or negative weights. It is similar with TF-IDF. But it instead the inverse document frequency with the inverse frequency of co-occurred tag. The intuition is that the more tags a term co-occurs with, then the less specific and important the term is. For example, the common word "photo" may co-occurs with many tags, and thus its weight should be small. For each top page, we extract the terms whose weight are greater than the average term weight of this page.

Then, we expand the initial tag set with a sub set of terms selected from the extracted terms. Usually, the common and noisy terms have similar "relations" with most of initial tags. This means that each irrelevant term have an even distribution of relation among the initial tags. However, the relevant terms usually have strong relations with some of the initial tags but have weaker relation to other initial tags, and these terms are preferable to be selected as the expanded tags. A strong relation between a term and a initial tag is reflected in that the term co-occur frequently with the initial tag, and vice versa. We use the posterior probability to represent the relation between tag t_i and term f:

$$P(t_i|f) = \frac{C(t_i \cap f)}{\sum\limits_{t_j \in Q} C(t_j \cap f)} \tag{10}$$

where $C(t_i \cap f)$ denotes the number of web pages which contain both tag t_i and term f. Then each term has a list of posterior probabilities for the initial tags. Terms that have a strong relation with tag t_i will have a high value for $P(t_i|f)$, and low values for $P(t_j|f), \forall_j \neq i$. The irrelevant terms will have more evenly distributed values among the posteriors. To measure the degree of the confidence that term f is related to the initial tag set, we compute its entropy E as following:

$$E(f) = -\sum_{t_i \in Q} P(t_i|f) \log_2 P(t_i|f) \tag{11}$$

The lower the entropy, the higher the confidence that the term is relative to one or several initial tags is. Similarly, higher entropy term has less confidence that it is relative to the initial tags. Thus the top terms with the smallest entropy are selected as the expanded tags.

5.3 Probability flow

Like the initial tags, the expanded tags also need a probability value to indicate their relevance to the query image. Since the expanded tag have no direct relation with the query image, we propose a method to measure probability flow based on the expanded tag's relation with initial tags. Probability flow reflects how strongly the probability values of the initial tags are inherited by the expanded tag. Then, the probability value of the expanded tag can be estimated by measuring the probability flow. We use the formula $PF(Q \lhd t_e)$ to denote the probability value which flows from the initial tag set Q to the expanded tag t_e, and then the probability of the expanded tag t_e is approximated by the formula:

$$P(t_e|I_q) \approx PF(Q \lhd t_e) = PF\left(\bigoplus_{t_i \in Q} t_i^e \lhd t_e\right) \tag{12}$$

where $\bigoplus_{t_i \in Q} t_i^e$ denotes the concept combination of the initial tag set when the expanded tag is t_e. Concept combination is important in IR, as the combination of words within a query topic represents a single underlying concept [3]. An important intuition in our concept combination is that the initial tag which is more relevant with the expanded tag can dominate other initial tags. For example, in the concept combination for initial tag set {"computer", "white" }, tag "computer" can be considered to dominate tag "white" if the expanded tag is related to computer e.g., "cpu". The probability value of the expanded tag is a part of probability value that flows into the expanded tag from the concept combination of initial tag set, and it is mainly determined by the initial tags which have strong relations with it. Thus, the probability of tag "cpu" is mainly determined by the probability of tag "computer". Given a expanded tag t_e, we use the following heuristic method to construct the concept combination of Q:

Step 1 The vector of each tag $t_i \in Q$ is represented by $\overrightarrow{t_i^e} =< w_{t_i^e}^1, w_{t_i^e}^2, ..., w_{t_i^e}^{|F|} >$, where $|F|$ is the number of pages within the text-based search result, and $w_{t_i}^k$ is estimated as following:

$$\begin{aligned} w_{t_i^e}^k &= \begin{cases} P(t_e|t_i)P(t_i|I_q) & t_i \in D_k \\ 0 & \text{else} \end{cases} \\ P(t_e|t_i) &= \varepsilon * \frac{C(t_e \cap t_i)}{C(t_i)} + (1-\varepsilon)\frac{C(\overline{t_e} \cap \overline{t_i})}{C(t_i)} \\ \varepsilon &= \frac{C(t_e \cap t_i)}{C(t_e \cap t_i) + C(\overline{t_e} \cap \overline{t_i})} \end{aligned} \tag{13}$$

where ε is a balance parameter, $C(t_e \cap t_i)$ is the co-occurrence which is the number of pages that contain both t_i and t_e, and $C(\overline{t_e} \cap \overline{t_i})$ is the number of web pages which contain neither t_i nor t_e. The probability $P(t_i|I_q)$ indicates the relevance of t_i to the query image I_q, and it is estimated using formula (2). The weight is a generating probability which combine both the co-occurrence and absence of the two tags with a balance parameter.

Step 2 The vector of the expanded tag t_e is represented by $\overrightarrow{t_e} =< w_{t_e}^1, w_{t_e}^2, ..., w_{t_e}^{|F|} >$, and the weight is estimated as following:

$$w_{t_e}^k = \begin{cases} 1, & t_e \in D_k \\ 0, & \text{else} \end{cases}$$

Step 3 The vector of concept combination is then represented by:

$$\overrightarrow{\underset{t_i \in Q}{\oplus} t_i^e} = \left\langle \underset{t_i \in Q}{\oplus} w_{t_i^e}^1, \underset{t_i \in Q}{\oplus} w_{t_i^e}^2, ..., \underset{t_i \in Q}{\oplus} w_{t_i^e}^{|F|} \right\rangle$$

where $\underset{t_i \in Q}{\oplus} w_{t_i^e}^k$ is estimated as following:

$$\begin{aligned} \underset{t_i \in Q}{\oplus} w_{t_i^e}^k &= P\left(t_e | \hat{t^k}\right) P\left(\hat{t^k} | I_q\right) \\ \hat{t^k} &= \arg\max_{t_i \in Q} P(t_e | t_i) \end{aligned} \tag{14}$$

In this vector, each element is dominated by the initial tag which has the greatest generating probability. Then the probability value which flows from the initial tag set Q to the expanded tag t_e is defined as the following formula:

$$PF\Big(\underset{t_i \in Q}{\oplus} t_i^e \lhd t_e\Big) = \frac{\sum_{1 \le k \le |F|} \underset{t_i \in Q}{\oplus} w_{t_i^e}^k * w_{t_e}^k}{\sum_{1 \le k \le |F|} w_{t_e}^k} \tag{15}$$

The probability flow has following properties:

1. $0 \le PF\Big(\underset{t_i \in Q}{\oplus} t_i^e \lhd t_e\Big) \le \max_{t_i \in Q} P(t_i | I_q)$.
2. If $t_j \in Q$, then $PF\Big(\underset{t_i \in Q}{\oplus} t_i^j \lhd t_j\Big) = P(t_j | I_q)$.
3. The initial tag which has a greater generating probability i.e., a stronger relation with the expanded tag has a stronger influence on the probability flow, and the more frequently the tag co-occur with the initial tags the more probability value it inherit.

Thus, it is reasonable to estimate the probability value of expanded tag by measuring the probability flow. This probability value is used as the initially relevant value of expanded tag, and then the RWR method is used to refine all the tags based on the tag graph.

6 Tag refinement

In the above sections, we estimate the probability values for initial tags and expanded tags independently, and it is assumed that the probability of expanded tag isn't greater than that of initial tag. In this section, we refine all the tags based on their correlation with other tags. In order to fully utilize the probability values estimated in the former stages, we use the RWR algorithm to refine the candidate tag set based on the tag graph. Then, the tags which are more relevant to the query image no matter they are initial tags or expanded tags are boosted.

6.1 Tag correlation and transition

To construct a directed tag graph, each candidate tag is regarded as a vertex and each two vertexes are connected with two directed edges. The directed edge is combined with a transition strength which indicates the probability of transition from the tail vertex to the head vertex. To estimate the transition strength, we first combine the co-occurrence and absence to estimate the correlation of two tags, and then the correlations are used to estimate the transition strength.

In the previous works, the co-occurrence is often used to estimate the correlation between two tags. However, this approach doesn't consider the absence of current tags and their correlation with other tags. We propose a balanced correlation measure method which considers both the co-occurrence and absence to estimate the correlation between two tags.

$$\begin{aligned} Cor(t_i, t_j) = {} & Pr(t_i, t_j) \log\left(\frac{C(t_i \cap t_j)}{C(t_i) + C(t_j) - C(t_i \cap t_j)} + 1\right) \\ & + Pr(\overline{t_i}, \overline{t_j}) \log\left(\frac{C(\overline{t_i} \cap \overline{t_j})}{C(\overline{t_i}) + C(\overline{t_j}) - C(\overline{t_i} \cap \overline{t_j})} + 1\right) \\ Pr(t_i, t_j) = {} & \frac{C(t_i \cap t_j)}{|F|},\ Pr(\overline{t_i}, \overline{t_j}) = \frac{C(\overline{t_i} \cap \overline{t_j})}{|F|} \end{aligned} \tag{16}$$

where $\overline{t_i}$ denotes the absence of tag t_i, and $Pr(t_i, t_j)$ is the joint probability that t_i and t_j appear together in a web page and $Pr(\overline{t_i}, \overline{t_j})$ is the joint probability that neither t_i no t_j appear in a web. A base value 1 is added to the log item to assure that the log value is greater than 0. With this formula, any two tags which not only often appear together but also are absent together frequently have a great correlation.

However, the correlation measured by the above-mentioned formula only considers the relation between two tags. There may be many hidden correlations which can't be discovered. On the other side, this estimated correlation is also symmetric. But the transition strength between two tags is asymmetric in some time. For example, the probability of an image with a tag "car" to be assigned with the tag "BMW" is smaller than the probability of an image with a tag "BMW" to be assigned with the tag "car". Thus, it isn't reasonable to represent the transition strength by the symmetric correlation directly. To alleviate these problems, we use the following formula to compute the transition strength from tag t_i to tag t_j:

$$Tran(t_i \rightarrow t_j) = \frac{\sum_{k=1}^{|\mathcal{R}|} Cor(t_i, t_k) * Cor(t_k, t_j)}{\sum_{k=1}^{\mathcal{R}} Cor(t_i, t_k)} \tag{17}$$

where $\mathcal{R}$ is the candidate tag set which is composed of both of the initial tags and expanded tags. As we can see, it is derived from the common neighbors. With this formula, the hidden correlation between two tags can also be discovered. It is obvious that this measure is asymmetric, which reflects the fact that the probability of transition from tag t_i to t_j is not always equal to the probability of transition from tag t_j to t_i.

6.2 Tag refinement with RWR

Random walk with Restart (RWR) method have been widely applied in machine learning and information retrieval fields [27, 32]. To boost the tags which are more relevant to the query image, we also use RWR to propagate the probability over the tag graph. To build tag graph, the transition matrix C is constructed by setting C_{ij} with the transition strength $Tran(t_i \to t_j)$. Then, each column of the matrix is normalized to be one.

$$Tran(t_i \to t_j) = \frac{Tran(t_i \to t_j)}{\sum_{i=1}^{|\mathcal{R}|} Tran(t_i \to t_j)} \tag{18}$$

We use $P_k(t_i|I_q)$ to denote the probability of tag t_i at the kth iterations. Then the probabilities of all tags at the kth iteration is denoted by $\overrightarrow{P_k} = [P_k(t_i|I_q)]_{|\mathcal{R}|\times 1}$. Thus, we can formulate the refining process using the following formula:

$$P_k(t_i|I_q) = (1-\alpha)\sum_{j=1}^{|\mathcal{R}|} C_{ij} P_{k-1}(t_j|I_q) + \alpha P'(t_i|I_q) \tag{19}$$

where $P'(t_i|I_q)$ is the normalized value of the initial probability which has been estimated in former sections, and $\alpha(0 < \alpha < 1)$ is a weight parameter. The above process will promote the tags which have strong relations with other relevant tags and weaken the tags which have less relation with other relevant tags. The RWR can also be re-written as a matrix format:

$$\overrightarrow{P_k} = (1-\alpha)C\overrightarrow{P_{k-1}} + \alpha\overrightarrow{P'} \tag{20}$$

The probability of each tag tends to be a fix value after a number of iterations. Finally, the top tags with their probability values are assigned to the query image, and the higher the probability is the more relevant the tag is. The tag probability is also helpful to tag-based image retrieval, clustering and classification.

7 Experiments

A series of experiments are conducted on web images and text documents database to evaluate the proposed algorithm CTSTag. First, we use the web images and text documents to test the precision and recall of different image tagging algorithms. Then, to show the effectiveness of the result tags on image retrieval, the retrieval performances of query by keyword based on the result tags of different image tagging algorithms are compared. Finally, to compare the effectiveness of the result tags on image classification and clustering, we take the image as a document and its tags as the terms contained by the document.

7.1 Dataset

Web database We downloaded one million tagged web images from Flickr using its API service. Each web image contains many surrounding text such as tags and description and so on. These downloaded images cover the topics of the evaluation dataset, and it is also ensured that they are evenly distributed over the different topics. The number of manually labeled tags per image varies from 2 to more than 100, and more than 10 million unique tags in total. By taking the tags of the query images in the evaluation dataset as queries, we search web text documents from the Internet using search engines such as Google. After filtering the duplicate text documents, we download about 300 thousands of web text documents from the Internet. Thus, these documents also cover the topics of the evaluation dataset. For each web document, we extract its text part to represent it. A WordNet stemmer is used to do stemming, and a snowball stop word list is used to eliminate stop words. Then, we index each document by the extracted keywords associated with their term frequencies.

Evaluation dataset We use the NUS-WIDE dataset [8] as the evaluation set. The NUS-WIDE dataset is a web image dataset created by NUS's Lab for media search. It contains 269,648 images downloaded from Flickr with a total of 425,059 unique tags. The number of manually labeled tags per image varies from 1 to more than 100, with an average value of about 30. The WordNet stemmer is also used to do stemming, and then we removing the tags which are used less than 10 times in the entire collection. Finally, the average number of manually labeled tags per images is about 15. This dataset also manually annotate the ground-truth for the 81 concepts which belong to different categories. The 81 concepts mostly correspond to the frequent tags in Flickr, and they contain both general concepts and specific concepts. In the classification and clustering experiments, we will use these concepts to label classes.

We randomly select 100,000 images from the evaluation dataset as the query images denoted as M_q to test the tagging algorithms. Then, other experiments, i.e., image retrieval, and image classification and clustering are performed to evaluate the result tags of M_q. For each query image, the CBIR retrieves visually similar images from the remaining images of the evaluation dataset and the Web database, and the text-based search retrieves other relevant images and text documents from the remaining images of the evaluation dataset and the Web database. Among varied visual feature supplied by NUS-WIDE, we use the 64-D color histogram and 73-D edge direction histogram to represent an image. Hence, our experimental setting is much closer to a real scenario.

7.2 Image tagging

In this section, a set of experiments are designed to test the effectiveness of our algorithm CTSTag. It is also compared with neighbor voting based tagging algorithm

(NVTag) [21] and the other search based image tagging algorithm (SBIA) [38]. For CTSTag, thirty initial tags which are more than the average number of tags per image are derived before the denoising precess. Then we expand the initial tag set with 50 other tags. For NVTag and SBIA, the number of result tags varies from several to the maximum number of 60 based on the number of manually labeled tags. To efficiently search millions of images by content, we divide the whole dataset into smaller subsets by the K-means clustering based on visual similarity. Each subset is indexed by a cluster centre. Then for a query image, we find neighbours within fewer subsets whose centers are the closest to the query.

We employ several standard criteria to evaluate the image tagging performance, i.e., average precision (Av_P) and average recall (Av_R). With m result tags, the average precision and recall are denoted by $Av\_P@m$ and $Av\_R@m$.

$$Av\_P = \frac{1}{N_{\text{query}}} \sum_{i=1}^{N_{\text{query}}} P(I_i) \tag{21}$$

$$Av\_R = \frac{1}{N_{\text{query}}} \sum_{i=1}^{N_{\text{query}}} R(I_i) \tag{22}$$

$$P(I_i) = \frac{|A_i \cap B_i|}{|A_i|},\ R(I_i) = \frac{|A_i \cap B_i|}{|B_i|}$$

where N_{query} is the total number of query images, and A_i is the set of result tags assigned to image I_i by the tagging algorithm and B_i is the set of human-produced tags for image I_i.

As most tagging algorithms tag the query images with the most common tags, many tags that appear less frequently may never be assigned to query images. However, the tags which appear less frequently may also be very relevant to the query image. Thus we also evaluate the performance with another criterion, i.e., tag coverage rate (Cov_rate). The tag coverage rate indicates how many tags in the vocabulary are used to tag query images by the tagging algorithm. With m result tags, the average coverage rate is denoted by $Cov\_rate@m$.

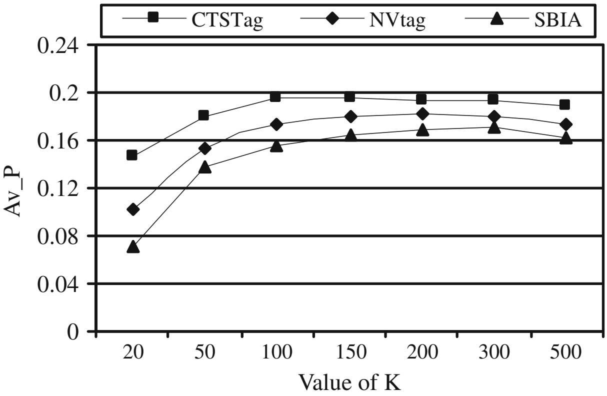
$$Cov\_rate = \frac{\left| \bigcup_{i=1}^{N_{\text{query}}} A_i \right|}{|V|} \tag{23}$$

where V is the tag vocabulary of the evaluation dataset.

In this set of experiments, two parameters are evaluated first. For CTStag, one parameter is the size K which indicates how many visually similar images with the greatest semantic consistency values are selected from the search result of CBIR, and the other parameter is the restart parameter α. After the parameters are fixed, we compare the $Av\_P@m$ and $Av\_R@m$ of different tagging algorithms.

In the algorithms SBIA and NVTag, K is the size of visually similar images. Thus, K is a common and crucial parameter of all the three algorithms. To facilitate the further evaluations, K is first decided. To compare the tagging performance with different K, the final tags of all the three algorithm are fixed to top 15 tags. Then,

Fig. 4 Average precision of different value of K



the average precision and recall are shown in Figs. 4 and 5 with *K* varied from 50 to 500. It shows that the performance curves of different algorithms are similar though their peaks are different. The performance of NVTag is the best when *K* is 200, and *K* is 300 for SBIA to achieve its best performance. As CTSTag select the images with greatest semantic consistency values, its curves are more sooth after that they achieve their peaks. The performance of CTSTag is similar when *K* is equal or larger than 150. Thus, we in the following experiments, *K* is set to be 300, 200 and 150 for SBIA, NVTag and CTSTag respectively.

The other parameter to be evaluated is the restart parameter α. As the SBIA also use RWR to refine candidate tags, we will compare the performance of SBIA and CTSTag with α varied from 0.1 to 0.9. The number of result tags *m* is also set to be 15. Figures 6 and 7 show the average precision and recall of both algorithms. It indicates that the curves of both figures are similar. Both precision and recall rates reach to the lowest value when α is 0.9 with *m* fixed. Both of the two algorithms achieve their best performances when α is set to be 0.3, Therefore, in all of the following experiments, the parameter α is set to be 0.3 for SBIA and CTSTag.

Fig. 5 Average recall of different value of K

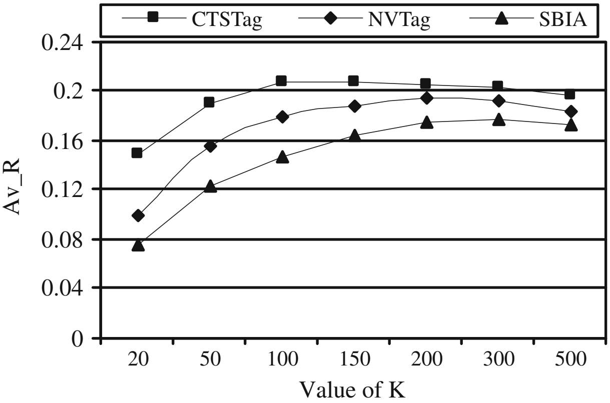
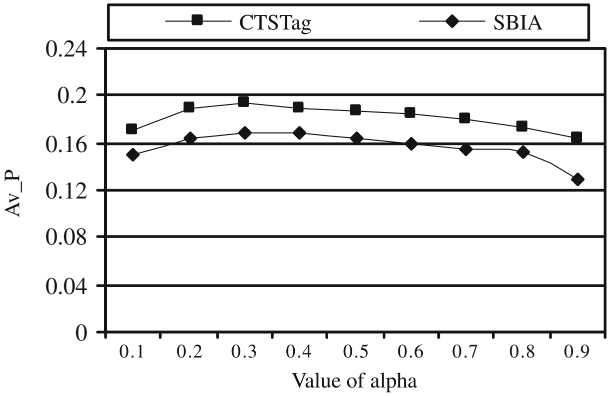


Fig. 6 Average precision of different value of α



Based on the aforementioned parameters evaluation, we then compare $Av\_P@m$ and $Av\_R@m$ of different algorithms in this set of experiments. Figures 8 and 9 show the $Av\_P@m$ and $Av\_R@m$ with m varied from 1 to 15. According to these figures, CTSTag consistently outperforms both NVTag and SBIA. There are several reasons. The first one is that we reduce the effect of noisy images by selecting the visually similar images which have great semantic consistency values. Second, the SBIA only use the TF-IDF to re-weight the tags associated with the retrieved images, which rarely consider the visual similarity between images and information of other relevant tags. The NVTag use the frequency of a tag minus its prior frequency to train the tags, which also give less consideration on visual similarity and information of other relevant tags. Our algorithm combines both the frequency and image visual similarity to derive the initial tags. Furthermore, we denoise the initial tags based on their correlation among the text-based search result, and we also boost the tags which are more related to the query image use the RWR method. Another observation is that all of these precisions decline with the number of result increasing. This is because that more noisy tags can be included when more tags are assigned to the query image. Figure 11 shows some examples of top-10 result tags returned by different algorithms.

Fig. 7 Average recall of different value of α

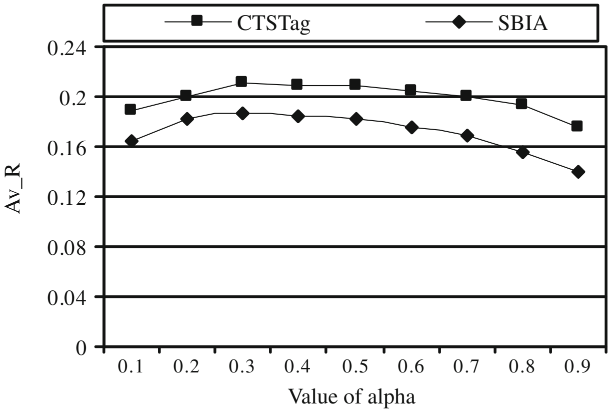
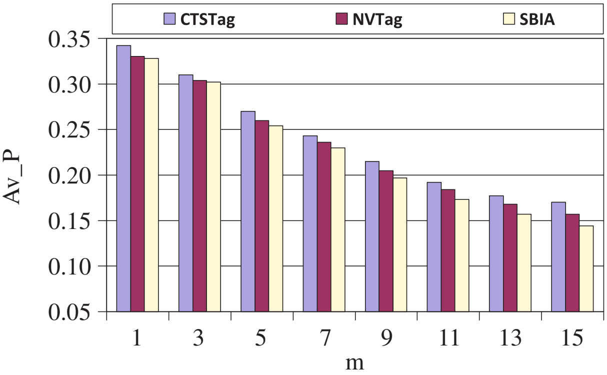
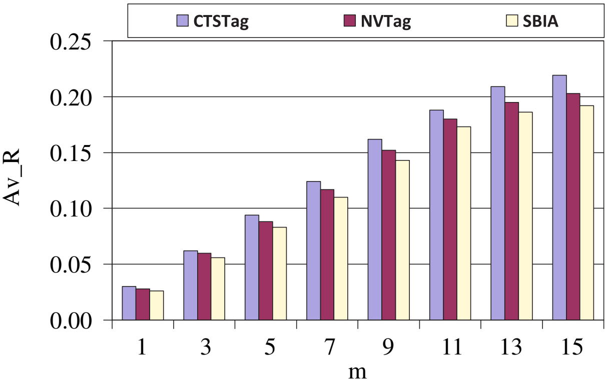


Fig. 8 Average top-m precision of different algorithms



In the previous evaluations, we strictly use the manually labeled tags of NUS-WIDE as the ground truth tags. Although this evaluation method can provide a fair comparison between different tagging algorithms, it may shrink the tagging performance. As the NVTag and SBIA prefer to tag the query image with common tags, then only a small subset of the training tags can be used to tag image. However, one of the advantages of our algorithm is to tag the query images with the relevant tags mined from an unlimited tag vocabulary. Thus we also compare their tag coverage rate. Figure 10 shows the *Cov_rate@m* of different algorithms with *m* varied from 1 to 15. It indicates that the NVTag tag the query images with a very small subset of the tag vocabulary and many tags are never selected. This is because that most of the assigned tags by NVTag are the ones which appear frequently in neighbor images. SBIA use a search-based method to retrieve the visually similar images. Then, all the tags associated with the retrieved image are ranked using a text-based search method. Its result tags have a larger coverage. However, SBIA also derives candidate tags from the visually similar images only. As CTSTag derives candidate tags not only from visually similar images but from other related images and text documents, and all the candidate tags are refined based on their correlation on the search result equally. Thus, the result tags of CTSTag have the largest coverage rate (Fig. 11).

Fig. 9 Average top-m recall of different algorithms



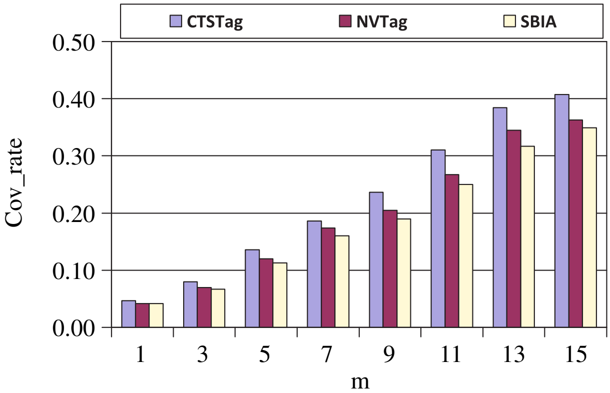


Fig. 10 Top-m coverage rate of different algorithms

The computation complexity of our approach for image tagging is mainly composed of three parts, i.e., initial tags deriving, and tag expansion and tag refinement. The computation complexity of initial tag deriving is about $O(m^2 + n^2)$, where m is the number of visually similar image retrieved by CBIR and n is the number of tags associated with the K images which have the greatest semantic consistency values. The complexity of tag expansion is about $O(l^2 + l' * T)$, where l is the number of initial tags, and l' is the number of initial tags after denoising and T is there number of terms extracted from the text-based search result. The computation complexity of tag refinement is $O(v^3)$, where v is the total number of initial tags and expanded tags. For the NVtag, its computation complexity is about $O(N * K' + N^2)$, where N is the number of tags associated with the K' neighbours of the query image. The computation complexity of SBIA is about $O(N' * K'' + M^3)$ where N'

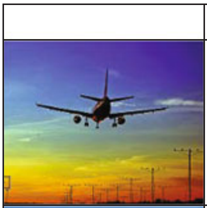
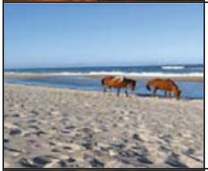
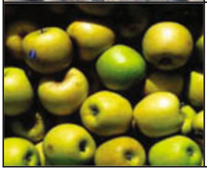
	Ground truth	CTSTag	SBIA	NVTag
	airplane, sunset, landing, aircraft, sky, sun, lights, aeroplane, land, wheels.	airplane, sunset, rain, lights, sky, transportation, sun, airport, airbus, Boeing.	airplane, sun, jet, tree, airport, sky, clounds, desert, model, aeroplane.	airplane, airport, sun, clounds, photo, sunset, aircraft, airbus, sky, aeroplane.
	White, ocean, water, horse, sand, beach, sky, animal, island.	horse, cloud, sea, ocean, dog, animal, water, ship, boat, bank.	horse, ocean, boy, road, sun, cloud, girl, jumping, sand.	beach, ocean, sea, sky, sun, clound, girl, runing, car.
	Green, yellow, Apples, Night, Canon, Digital, ixus50, food, nature	Apple, fruit, tree, green, store, retail, food, nature, juice, life.	Apple, green, leaf, tree, toy, desk, dish, photo, picture, love	Tree, green, apple, pear, picture, season, nature, food, photo, boy.

Fig. 11 Examples of image tagging result

is the number of tags associated with the K'' images retrieved by SBIA and M is the number of tags derived from those associated with the retrieved images. Since our algorithm needs some additional processes of tag mining from the text-based search result, its computational complexity is greater than that of NVTag and SBIA. Though our approach is some more complicated, it shows much better performance in the evaluation. We think this trade-off is worthwhile because the time efficiency can be potentially increased by adopting parallel algorithm and distributed architectures.

7.3 Image retrieval

In this section, we employ a general tag-based image retrieval used in existing systems such as Flickr to evaluate the effectiveness of the result tags of different tagging algorithms. The retrieval system indexes the result tags of the query images M_q, and a well-founded ranking function Okapi BM25 is used to ranking the retrieved images [16]. Give a query q containing keywords $\{t_1, t_2, ..., t_n\}$, the relevance score $F(q, I_i)$ of an image I_i is estimated by the following formula:

$$\begin{aligned} F(q, I_i) &= \sum_{j=1}^{n} qtf(t_j)idf(t_j)\frac{tf(t_j) * (k+1)}{tf(t_j) + k * \left(1 - b + b * \frac{L_{I_i}}{L_{\text{ave}}}\right)} \\ &= \sum_{j=1}^{n} qtf(t_j)idf(t_j)\frac{tf(t_j) * (k+1)}{tf(t_j) + k} \\ idf(t_j) &= \log\left(\frac{N}{|L_{t_j}|} + 1\right) \end{aligned} \tag{24}$$

where, $qtf(t_j)$ is the frequency of tag t_j in query q, and $tf(t_j)$ is the frequency of t_j in image I_i. L_{I_i} is the number of result tags in image I_i, and L_{ave} is the average number of image tags. N is the total number of images used to evaluate image retrieval, and $|L_{t_j}|$ is the number of images tagged with t_j. The other parameter b $(0 \leq b \leq 1)$ determines the scaling by L_{I_i}. Since individual tags are typically used once per image, $tf(t_j)$ is set to be 1 for SBIA and the relevance value for NVTag. As CTSTag assigns each result tag with a probability value, we substitute $tf(t_j)$ with its probability value. It is in this manner that we embed the tag probability into the image retrieval framework. In realization, we set the parameters k and b to be 1 and 0.5 for simplicity.

To evaluate the retrieval result, we use two evaluation criteria i.e. the average precision $Av\_P@m$ of top m retrieved images and the average recall $Av\_R@m$ of top m retrieved images.

$$Av\_P@m = \frac{1}{n}\sum_{i=1}^{n} P(q_i)@m \tag{25}$$

$$Av\_R@m = \frac{1}{n}\sum_{i=1}^{n} R(q_i)@m \tag{26}$$

$$P(q_i)@m = \frac{RI_i(m)}{m}, \quad R(q_i)@m = \frac{RI_i(m)}{groundtruth(q_i)}$$

where $RI_i(m)$ is the number of relevant images in the top m result images of query q_i, and n is the total number of queries and $goundtruth(q_i)$ is the total number of relevant images of query q_i.

Since the NVTag and SBIA only derived candidate tags from visually similar images, the tags that appear less frequently in the visually similar images are less likely to be selected. Our tagging algorithm can assign the query images with the tags that are relevant but appear less frequently or never appear in the visually similar images. To compare the effectiveness of result tags on image retrieval, we design two experiments, i.e., one using single-word queries, one using double-word queries.

In the single-word queries experiment, we use the common words as the queries, e.g., {{airport}, {boat}, {beach}, {bridge}, {car}, {computer}, {dog}, {fish}, {sun}, {tree}}. In the double-word queries experiment, we combine the common word with a relatively uncommon word to set a query, e.g., {{airport girl}, {boat jumping}, {beach swimsuits}, {bridge cloud}, {car accessories}, {computer office}, {dog baseball}, {fish vocation}, {sun sports}, {tree countryside}}.

Figures 12 and 13 show the average retrieval performances of single-word queries experiment. Figures 14 and 15 show the the average retrieval performances of double-word queries experiment. Two conclusions can be drawn from these figures. First, when we set a common word as a query, the performances of image retrieval based on different result tags are similar. This is because that common tags are preferred to be selected to tag the query images by all the three algorithms. Second, when we add the single-word query with an uncommon word, the performance of image retrieval based on result tags of CTSTag are improved obviously. This because that both NVTag and SBIA assign the query image with the tags that are common in the visually similar images, and the uncommon tags are less likely to be selected. Since the double-word queries contain the uncommon words, many images whose result tags don't contain the uncommon tags can't be retrieved. However, the CTSTag expand the initial tag set with other relevant tags that may be uncommon or never appear in the visually similar images, and then the correlation between all the tags are used to boost the tags which may be neglected by other algorithms.

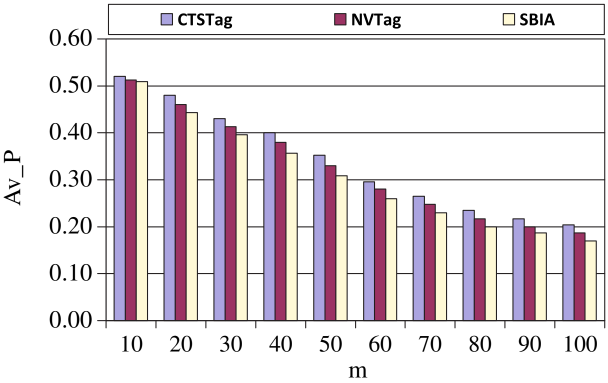


Fig. 12 Average top-m precision of different algorithms

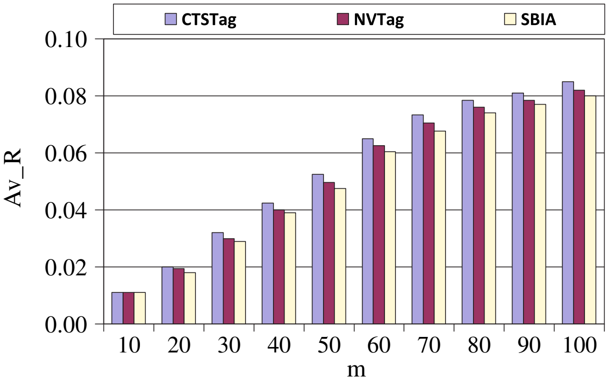


Fig. 13 Average top-m recall of different algorithms

7.4 Image classification and clustering

In this section, we will show the results for classification as well as clustering of query images M_q using feature vectors obtained by different algorithm. We will perform the experiments on the feature vectors constructed by the following methods using result tags:

1. NVTag: Vectors constructed based on the result tags produced by the NVTag algorithm. The value of each dimension it set with the relevance value estimated by NVTag if the current image is assigned with the corresponding tag and 0 otherwise.
2. SBIA: Vectors constructed based on the result tags produced by the SBIA algorithm. The value of each dimension is set with 1 if the current image is assigned with the corresponding tag and 0 otherwise.
3. CTSTag: Vectors constructed based on the result tags produced by the CTSTag algorithm. The value of each dimension is set with the probability value if the current image is assigned with the corresponding tag and 0 otherwise.

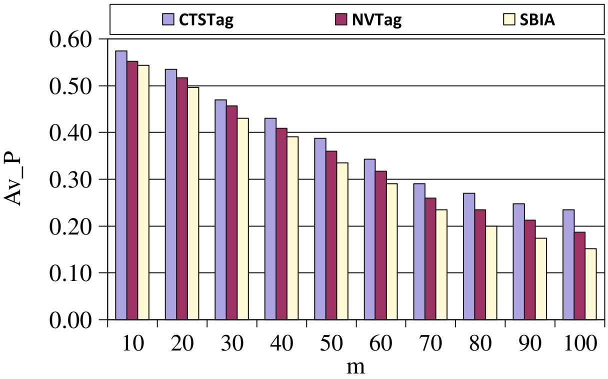


Fig. 14 Average top-m precision of different algorithms

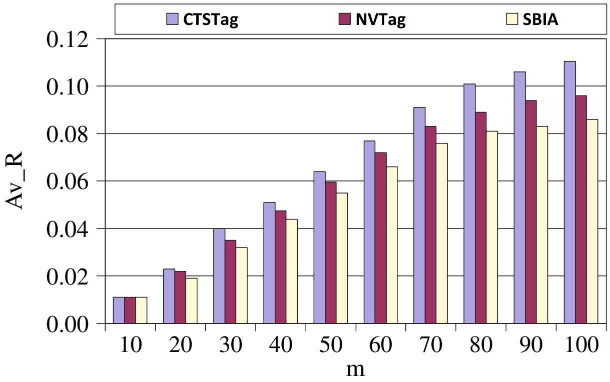


Fig. 15 Average top-m recall of different algorithms

7.4.1 Classification experiments

For the classes used in our classification experiments, we use the ground truth of 81 concepts in the NUS-WIDE dataset to label 81 classes. For each query image in M_q, it is assigned to the classes if its annotated by the corresponding concepts. Then we omit the images that are contained in more than one class, and we choose 50 categories each of which contains more than 1,000 images. For each class, we randomly choose 500 images for training the classification and a disjoint set of 500 images for test. The intuition to do this is that we obtain equal numbers of training/test images per class. Then KNN is used to classify the images, as it has been shown perform very well for text-based classification tasks.

We train different classifiers based on different numbers of training images $N =$ 100, 200, 300, 400, 500 per class respectively. The criteria to evaluate the performance of image classification are F1-measure and ROC curve. The results of the comparisons for F1 value is shown in Fig. 16, and the comparison for ROC

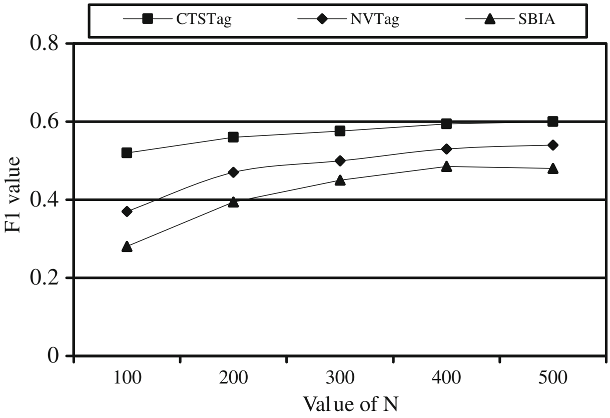


Fig. 16 F1 value of different tag representations for image classification

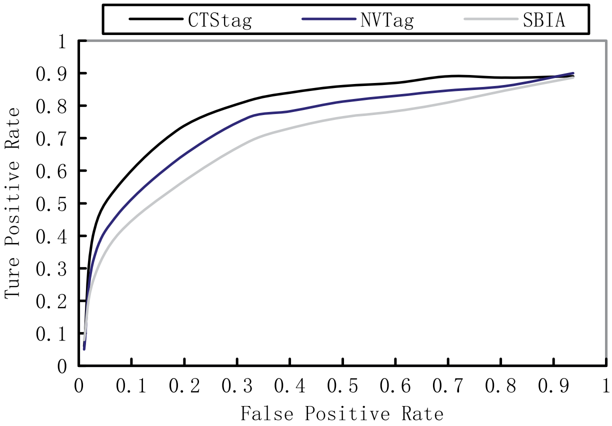


Fig. 17 ROC curves for performance comparison of classification

curve is shown in Fig. 17. Several conclusions can be drawn from these comparisons. First, CTSTag always provides a better performance than SBIA and NVTag given different number of classes. Second, with the number of training images varying, the performance of CTSTag varied in a smaller scale than that of SBIA of NVTag. This is because that CTSTag has a better performance of tagging and also a tag probability is helpful to weight the tag for image classification. Thus, for the CTSTag, a smaller number of images can achieve the same performance.

7.4.2 Clustering experiments

Unlike other works which employs visual similarity to clustering image [5], this set of experiments aims to analyze the clustering performance of tag representation for image. We use the *k-means* [11] to partition the images set into a groups called clusters. Unlike classification results, the clusters do not have topic labels. Let k be the number of classes or clusters, N_i the total number of clustered images in the ith

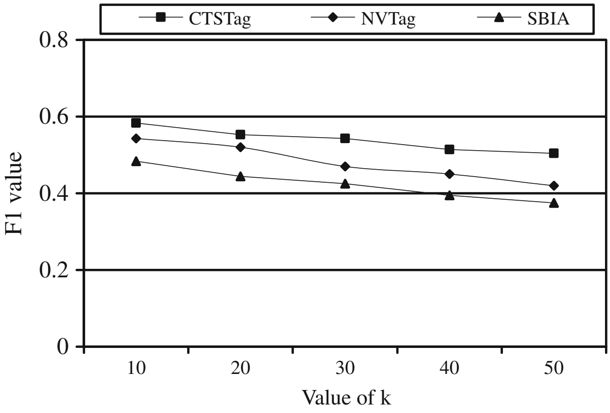


Fig. 18 F1 value of different tag representations for image clustering

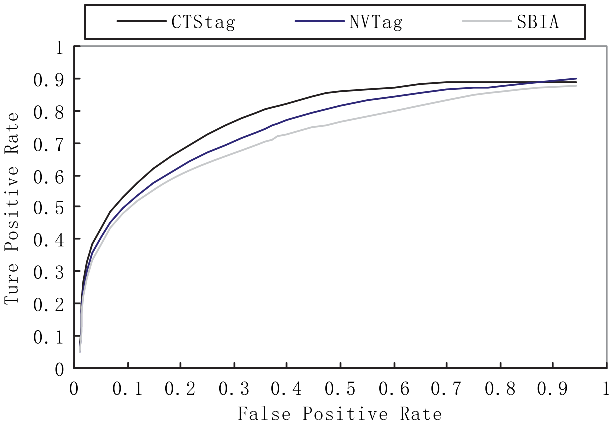


Fig. 19 ROC curves for performance comparison of clustering

class, C_i the size of the ith class, $N_{i,j}$ the number of images contained in the ith class and having cluster label j. We define the F1 value as following:

$$clu\_precision = \frac{\max\limits_{j_1,\ldots,j_k \in \mathrm{map}(1,\ldots,k)} \sum_{i=1}^{k} \frac{N_{i,j_i}}{N_i}}{k}$$

$$clu\_recall = \frac{\max\limits_{j_1,\ldots,j_k \in \mathrm{map}(1,\ldots,k)} \sum_{i=1}^{k} \frac{N_{i,j_i}}{C_i}}{k}$$

$$F1 = \frac{2 * clu\_precision * clu\_recall}{clu\_precision + clu\_recall} \tag{27}$$

where $(j_1, ..., j_k) \in map(1, ..., k)$ is a one-to-one mapping from the cluster labels to the classes. The F1 value interprets how good the partitioning produced by the clustering method reflects the actual class structure. We randomly select a number of class $k = 10, 20, 30, 40, 50$ from the classes built in the former section, and we also randomly select 1,000 images per class. Then we perform several times of *k-means* clustering for each selection of k and report the best precision for each time of clustering.

The comparison of F1 value is shown in Fig. 18, and the comparison of ROC curves is given in Fig. 19. The main observations are very similar to the image classification mentioned in the former section. It indicates that clustering with tags generated by CTSTag outperform than clustering with tags generated by NVTag and SBIA. Meanwhile, all of their performances drop down with the number of clusters varied from 10 to 50. This is because that the number of noisy tags increases with the increasing number of classes.

8 Conclusions

In this paper, we formulate the image tagging as a search problem, and a novel probabilistic image tagging algorithm based on mining Web search result is proposed, in which the result tags are improved by exploiting text-based search result. First, CBIR is used to retrieve visually similar images, and then the initial tags with their probability value are derived from these retrieved images which are ranked by their semantic consistency values. Second, with the initial tags as a query the text-based search is performed to retrieve other relevant web resource. Then, the initial tag set is expanded with terms mined from the text-based search result. Finally, RWR is used to refine all of the tags based on the tag transition matrix. The search based framework guarantees that our algorithm isn't limited to the training dataset. Furthermore, this algorithm can use other source of knowledge to improve image tagging. Experimental results on NUS-WIDE show not only the effectiveness of our tagging algorithm but also the effectiveness of other applications, i.e., image retrieval by using the result tags as the indexes, image classification and clustering using tag representation for image.

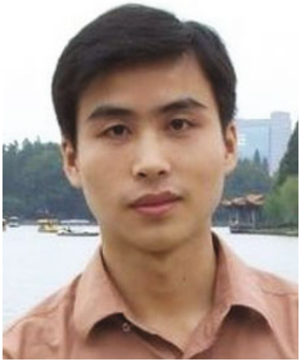
Acknowledgements This work was supported by the National Natural Science Foundations of China (60973105 and 61003111), and the fund of the State Key Laboratory of Software Development Environment (SKLSDE-2011ZX-03). The authors would like to thank the Editors and the anonymous reviewers 739 for their valuable comments and remarks on the previous versions of this paper.

References

1. Bailloeul T, Zhu CZ, Xu YH (2008) Automatic image tagging as a random walk with priors on the canonical correlation subspace. In: Proceeding of 9th ACM international conference on multimedia information retrieval, pp 75–82
2. Barnard K, Duygulu P, Forsyth D, de Freitas N, Blei DM, Jordan MI (2003) Matching words and pictures. J Mach Learn Res 3(6):1107–1135
3. Bruza PD, Song D (2002) Inferring query models by computing information flow. In: Proceedings of CIKM 2002, pp 260–269
4. Cao G, Nie J, Gao J, Robertson S (2008) Selecting good expansion terms for pseudo-relevance feedback. In: Proceedings of the 31st ACM SIGIR conference on research and development in information retrieval. Singapore, pp 243–250
5. Cao L, Pozo AD, Jin X, Luo J, Han J (2010) RankCompete: simultaneous ranking and clustering of web photos. In: Proceedings of the 19th international conference on World Wide Web
6. Chang SF, He J, Jiang YG, El Khoury E, Ngo CW, Yanagawa A, Zavesky E (2008) Columbia University/VIREO-CityU/IRIT TRECVID2008 high-level feature extraction and interactive video search. In: Proceedings of TRECVID 2008
7. Chen XY, Mu YD, Yan SC, Chua TS (2010) Efficient large-scale image annotation by probabilistic collaborative multi-label propagation. In: Proceedings of 18th annual ACM international conference on multimedia, pp 35–44
8. Chua T-S, Tang J, Hong R, Li H, Luo Z, Zheng Y-T (2009) NUS-WIDE: a real-world web image database from national University of Singapore. In: ACM international conference on image and video retrieval. Greece, 8–10 Jul 2009
9. Croft WB, Lafferty J (2002) Language models for information retrieval. Kluwer int. series on information retrieval, vol 13. Kluwer Academic Publishers
10. Geng B, Yang L, Xu C, Hua X (2008) Collaborative learning for image and video annotation. In: Proceeding of the 1st ACM international conference on multimedia information retrieval, pp 443–450
11. Han J, Kamber M (2001) Data mining: concepts and techniques. Morgan Kaufmann
12. Heesch D, Yavlinsky A, Ruger S (2006) Nnk: networks and automated annotation for browsing large image collections from the World Wide Web. In: Proceedings of the 14th ACM International Conference on Multimedia, pp 493–494
13. Hong R, Wang M, Xu M, Yan S, Chua T-S (2010) Dynamic caption: video accessibility enhancement for hearing impairment. In: ACM international conference on multimedia (ACM MM)

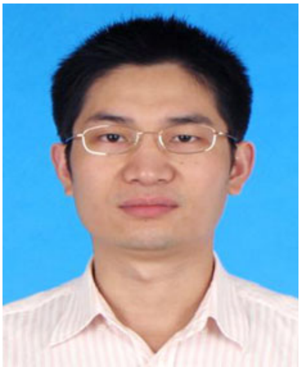
14. Naphade M, Smith JR, Tesic J, Chang S-F, Hsu W, Kennedy L, Hauptmann A, Curtis J (2006) Large-scale concept ontology for multimedia. IEEE Multimed 13(3):86–91
15. Jing F, Wang C, Yao Y, Deng K, Zhang L, Ma W (2006) IGroup: web image search results clustering. In: Proceedings of the 14th annual ACM international conference on multimedia, pp 377–384
16. Jones KS, Walker S, Robertson SE (2000) A probabilistic model of information retrieval: development and comparative experiments—part 2. Journal of Information Processing and Management 36(6):809–840
17. Lei W, Linjun Y, Nenghai Y, Hua XS (2009) Learning to tag. In: Proceedings of the 18th ACM international conference on World Wide Web, pp 20–24
18. Lei W, Steven CH, Rong Jin H, Jianke Z, Nenghai Y (2009) Distance Metric Learning from Uncertain Side Information with Application to Automated Photo Tagging. In: Proceeding of 17th ACM international conference on multimedia, pp 15–24
19. Li J, Wang JZ (2006) Real-time computerized annotation of pictures. In: Proceedings of the 14th annual ACM international conference on multimedia, pp 911–920
20. Li X, Snoek CGM (2009) Visual categorization with negative examples for free. In: Proceedings of the 17th international conference on multimedia, pp 661–664
21. Li X, Snoek CG, Worring M (2009) Learning social tag relevance by neighbor voting. IEEE Trans Multimed 11(7):1310–1322
22. Li X-R, Snoek CG, Worring M (2009) Annotating images by harnessing worldwide user-tagged photos. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, pp 3717–3720
23. Liu J, Wang B, Li M, Li Z, Ma W, Lu H, Ma S (2007) Dual cross-media relevance model for image annotation. In: Proceedings of the 15th international conference on multimedia, pp 605–614
24. Liu Y, Zhang D, Lu G, Ma WY (2007) A survey of content-based image retrieval with high-level semantics. Pattern Recogn 40(1):262–282
25. Liu D, Wang M, Hua XS, Zhang HJ (2009) Tag ranking. In: Proceeding of the 18th ACM international conference on World Wide Web, pp 351–340
26. Lu Y, Zhang L, Tian Q, Ma W-Y (2008) What are the high-level concepts with small semantic gaps? In: Proceeding of IEEE 21th conference on computer vision and pattern recognition, pp 1–8
27. Page L, Brin S, Motwani R, Winograd T (1998) The pagerank citationranking: bringing order to theWeb, technical report. Stanford University, Stanford
28. Russell BC, Torralba A, Murphy KP, Freeman WT (2008) LabelMe: a database and web-based tool for image annotation. Int J Comput Vis 77(1):157–173
29. Setz AT, Snoek CGM (2009) Can social tagged images aid concept-based video search? In: Proceedings of ICME, pp 1460–1463
30. Shen Y, Fan JP (2010) Leveraging loosely-tagged images and inter-object correlations for tag recommendation. In: Proceedings of 18th annual ACM international conference on multimedia, pp 5–14
31. Siersdorfer S, San Pedro J, Sanderson M (2009) Automatic video tagging using content redundancy. In: Proceeding of the 32nd ACM international conference on research and development in information retrieval, pp 16–23
32. Tong H, Faloutsos C, Pan J (2006) Fast random walk with restart and its applications. In: Proceedings of the IEEE 6th international conference on data mining, pp 613–622
33. Tsikrika T, Diou C, de Vries AP, Delopoulos A (2010) Reliability and effectiveness of clickthrough data for automatic image annotation. Multimed Tools Appl 55(1):27–52
34. Turtle HR, Croft WB (1992) A comparison of text retrieval models. Comput J 35(3):279–298
35. Vassilieva NS (2009) Content-based image retrieval methods. Program Comput Softw 35(3): 158–180
36. Wang C, Jing F, Zhang L, Zhang H-J (2006) Image annotation refinement using random walk with restarts. In: Proceedings of 14th ACM international conference on multimedia, pp 647–650
37. Wang X, Zhang L, Jing F, Ma W (2006) AnnoSearch: image auto-annotation by search. In: Proceedings of the 19th IEEE computer society conference on computer vision and pattern recognition, vol 2, pp 1483–1490
38. Wang C, Jing F, Zhang L, Zhang HJ (2008) Scalable search-based image annotation. Multimedia Syst 14(4):205–220
39. Wang XJ, Zhang L, Li XR, Ma W-Y (2008) Annotating images by mining image search results. IEEE Trans Pattern Anal Mach Intell 30(11):1919–1932

40. Wang M, Hua X-S, Tang J, Hong R (2009) Beyond distance measurement: constructing neighborhood similarity for video annotation. IEEE Trans Multimedia 11(3):465–476
41. Yang K, Wang M, Zhang H (2009) Active tagging for image indexing. In: Proceedings of the IEEE international conference on multimedia and expo, pp 1620–1623
42. Zhou X, Wang M, Zhang Q, Zhang J, Shi B (2007) Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. In: Proceedings of the 6th ACM international conference on image and video retrieval, pp 25–32



Xiaoming Zhang was born in Hunan, China, on December 7, 1980. He received the B.Sc. degree, and the M.Sc. degrees in computer science and technology from the National University of Defence Technology, China, in 2003, 2007 respectively. Now he is the doctoral student at the Beihang University. His major interests are multimedia retrieval, image tagging and data mining.

Zhoujun Li received his M.Sc and Ph.D degrees in computer science from the National University of Defence Technology, China, in 1984 and 1999, respectively. He is currently working at the school of computer, Beihang University, and he has been the professor since 2001. His research interests include the data mining, information retrieval, database.



Wenhan Chao received his Ph.D degrees in computer science from the National University of Defence Technology 2007. He is currently working at the school of computer, Beihang University, and he has been the lecturer since 2007. His research interests include the data mining, information retrieval, database.