3-2017

# Social tag relevance learning via ranking-oriented neighbor voting

Chaoran CUI
*Shandong University of Finance and Economics*

Jialie SHEN
*Singapore Management University*, jlshen@smu.edu.sg

Jun MA
*Shandong University*

Tao LIAN
*Shandong University*



## Citation

CUI, Chaoran; SHEN, Jialie; MA, Jun; and LIAN, Tao. Social tag relevance learning via ranking-oriented neighbor voting. (2017). *Multimedia Tools and Applications*. 76, (6), 8831-8857.
**Available at:** https://ink.library.smu.edu.sg/sis_research/3542

Published in Multimedia Tools and Applications, 2016 April, Advanced Online.
http://doi.org/10.1007/s11042-016-3512-1

# Social tag relevance learning via ranking-oriented neighbor voting

**Chaoran Cui[1] · Jialie Shen[2] · Jun Ma[3] · Tao Lian[3]**

**Abstract** High quality tags play a critical role in applications involving online multimedia search, such as social image annotation, sharing and browsing. However, user-generated tags in real world are often imprecise and incomplete to describe the image contents, which severely degrades the performance of current search systems. To improve the descriptive powers of social tags, a fundamental issue is tag relevance learning, which concerns how to interpret the relevance of a tag with respect to the contents of an image effectively. In this paper, we investigate the problem from a new perspective of learning to rank, and develop a novel approach to facilitate tag relevance learning to directly optimize the ranking performance of tag-based image search. Specifically, a supervision step is introduced into the neighbor voting scheme, in which the tag relevance is estimated by accumulating votes from visual neighbors. Through explicitly modeling the neighbor weights and tag correlations, the risk of making heuristic assumptions is effectively avoided. Besides, our approach does not suffer from the scalability problem since a generic model is learned that can be applied to all tags. Extensive experiments on two benchmark datasets in comparison with the state-of-the-art methods demonstrate the promise of our approach.

✉ Chaoran Cui
bruincui@gmail.com

Jialie Shen
jlshen@smu.edu.sg

Jun Ma
majun@sdu.edu.cn

Tao Lian
liantao1988@gmail.com

[1] School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China

[2] School of Information Systems, Singapore Management University, Singapore, Singapore

[3] School of Computer Science and Technology, Shandong University, Jinan, China

**Keywords** Tag-based image search · Tag relevance learning · Neighbor voting · Learning to rank

## 1 Introduction

With the rapid development of multimedia and social network technologies, we have witnessed an explosive growth of social images in recent years. This also raises an urgent demand for smart search technologies to explore large-scale social image collections, such as Flickr[1] and Pinterest[2]. Distinguished from general Web images with rich contextual information such as titles and surrounding text on Web pages, social images are frequently associated with user-generated tags that describe the image contents. Naturally, these tags can be used to index social images to facilitate the search process. Therefore, tag-based image search has become the *de facto* choice to access and browse resources in current social image repositories [5, 29].

In spite of the increasing popularity of tag-based image search, its performance is still far from satisfactory. One of the major reasons for this stagnation is due to the inferior quality of user-generated tags. As reported in [14], only about 50 % of the tags in Flickr truly reflect the contents of the images. Moreover, due to the limitations of both time and domain knowledge about the labeling process, it is impractical for an ordinary user to annotate an image comprehensively. In consideration of their characteristics of *imprecise* and *incomplete*, the user-generated tags are incapable of being the qualified indexing keywords for tag-based image search. Therefore, to enhance tag-based image search, a fundamental challenge is how to estimate the relevance of a tag with respect to the visual contents of an image, which is referred to as the problem of *tag relevance learning* [22].

Considerable research efforts have been invested to address the problem of tag relevance learning. Many methods rely on supervised machine learning algorithms to build the connection between visual features and semantic concepts [4, 48]. In general, a classifier is first learned for each tag over the training set, and then the relevance of a particular tag regarding the image contents is estimated by the classifier prediction score. However, this sort of tag-specific modeling has been widely challenged [19] with the concerns about its inefficiency when applying models for the huge number of tags in real world. Besides, how to select high quality training examples at large scale is still an open research problem [21].

Confronted with the huge amount of emergent social media and tags, many unsupervised data-driven approaches have been recently developed [1]. Among these approaches, the neighbor voting [19, 20] scheme has attracted increasing attention and proven to be one of the most promising solution for tag relevance learning [37]. It is based on the intuition that if visually similar images share the same tags, these tags are likely to reflect the actual visual contents. The scheme first finds visually similar neighbors of an image, and then estimates the relevance of each tag by accumulating votes from visual neighbors. Compared with the supervised methods aforementioned, the neighbor voting scheme exhibits the advantage in scalability, since it does not require offline model learning. However, the unsupervised

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

[2] http://www.pinterest.com/

nature also makes it difficult to handle some of the key problems in modeling. Existing methods usually heuristically determine the weights of neighbors by the visual similarity between images [15], or in a soft manner that performs a random walk over the $k$-nearest neighbor graph [50]. These assumptions may not be valid to the same degree in different situations, and the performance gain obtained by these heuristic weighting strategies also appears to be limited [50]. In addition, many contextual information such as tag correlations, which is highly beneficial as shown previously in tag-based applications, has not yet been fully exploited in current neighbor voting methods [36].

Apart from the above deficiencies of their own, there also remains a common limitation in both existing supervised and unsupervised solutions for tag relevance learning. In essence, the aim of tag relevance learning is to find out the tags that can be applied as content descriptors for the images, so that the accuracy of tag-based image search can be further improved by indexing the images with these reliable tags. However, existing studies generally perform tag relevance learning without the explicit intention of improving the performance of tag-based image search. In most cases, the unsupervised techniques rely on heuristic rules to estimate tag relevance [20], while those supervised counterparts conduct the learning process by optimizing the classification accuracy for specific tags [48], or maximizing the likelihood of the annotations of training images [38]. We argue that these objectives are not directly related to the search performance, and optimizing them does not necessarily yield good search results.

Motivated by the earlier discussions, we aim to propose a novel approach for tag relevance learning to mitigate the limitations of current methods. Towards this end, a supervision step is introduced into the neighbor voting scheme. Through this step, the possibility is offered that utilizing the information from the data collection to reduce the need for making heuristic assumptions. We explicitly model the individual weight of each visual neighbor, and the pairwise correlations between tags are also captured through a low-rank approximation. More importantly, we seek to investigate the problem of tag relevance learning from a new perspective of learning to rank. Tag relevance is regarded as a ranking criterion, and the learning process is consequently conducted by directly optimizing the ranking performance of tag-based image search. Besides, our approach still maintains the good scalability although the supervision step is introduced. This is because the ground truth used in training is only for a small number of query tags, but from which a tag-independent generic model can be learned and applied to predict relevance for all tags.

In literature, social tags denote keywords generated by ordinary users (rather than experts) to describe the media contents in online social platforms [19, 22]. In this paper, we target at the problem of tag relevance learning for improving the descriptive power of social tags with respect to the image contents. Although the proposed methodology could be adapted to other applications for general data (e.g., image annotation), our main purpose is to demonstrate its effectiveness for real-world user-tagged social images. Extensive experiments have been conducted on two benchmark datasets by applying our approach to both applications of tag-based image search and automatic tag recommendation in social media. The results show that our approach achieves a remarkable improvement over the state-of-the-art methods from different perspectives.

The remainder of this paper is organized as follows. Section 2 gives a review of related work. Section 3 presents the proposed ranking-oriented neighbor voting framework for tag relevance learning. Section 4 describes the experimental setup. Section 5 reports the experimental results and analysis. Section 6 concludes this work and points out some directions for future research.

## 2 Related work

In this section, we review the existing literature on tag relevance learning, and categorize the related research into three classes [22]: model-based methods, instance-based methods, and transduction-based methods.

### 2.1 Model-based methods

This class of methods rely on training examples to build the parameterized models for tag relevance learning. Among these model-based methods, two main groups can be identified: tag-specific modeling and tag-generic modeling. For tag-specific modeling, a typical paradigm is to formulate tag relevance learning as a classification problem, in which each tag is treated as a class label and the relevance value is estimated by the classifier prediction score. In [4], linear SVM classifiers were trained with features augmented by pre-trained classifiers of popular tags for social image search. Zhou et al. [48] employed social tagged images to learn visual concept detectors that can be applied to recognize concepts at both image-level and image region-level. Shen et al. [30] partitioned each tagged image into a set of image instances, and developed a multiple instance learning algorithm for instance label identification by automatically identifying the correspondences between multiple tags and image instances. A multi-task structured SVM was further developed for exploiting the inter-tag correlations to achieve more effective learning of inter-related object classifiers.

Besides the paradigm of building classifiers, Feng et al. [8] aggregated the prediction models for different tags into a matrix. Instead of learning each prediction model independently, they proposed to learn all the prediction models simultaneously by exploring the theory of matrix recovery, and introduced a trace norm regularization to capture the dependence among different tags and to control the model complexity. In [38], logistic regression models were built per tag to boost the recall of the weighted nearest neighbor model for rare tags. In [42], Weston et al. proposed to learn a low-dimensional joint embedding space for both images and tags through optimizing the precision at the top of the ranked list of annotations for given images. Frome et al. [9] applied the deep learning technique to learn a visual-semantic embedding model using both labeled image data as well as semantic information gleaned from unannotated texts. Gong et al. [10] presented a multi-view embedding approach for images, tags, and their semantics. To keep the learning process scalable, explicit nonlinear kernel mappings were used to efficiently approximate kernel CCA (Canonical Correlation Analysis).

For tag-generic modeling, it maintains a uniform modeling configuration for all tags and consequently is more flexible to adapt to new tags. Wu et al. [43] proposed a multi-modal tag recommendation method based on both tag and visual correlations. Each modality was used to generate a ranking feature, and the tag relevance function was an optimal combination of these ranking features determined with the RankBoost algorithm. Similarly in [41], the authors proposed a semi-supervised learning framework for tag ranking, which established a ranking projection from the visual word distribution to the tag distribution. In this paper, our approach is also an example of tag-generic modeling, in which a supervised neighbor voting model is learned and applied to predict relevance for all tags.

### 2.2 Instance-based methods

This class of methods does not build an explicit model, but directly constructs the hypothesis by comparing test images with training instances. Representative members of instance-based methods are the neighbor voting algorithm [19] and its variants [2, 15, 16, 20, 36, 50], where the tag relevance function is estimated by counting the tag frequency in the visual neighbors of the image. The set of visual neighbors was typically created with the early fusion of global features [19], the late fusion of multiple single-feature learners [16, 20], the distance metric learning to combine different visual features [38], and a cross-modal representation mapping between visual and tag features [2]. Once the visual neighbors are determined, the standard neighbor voting algorithm [19] simply let all neighbors vote equally, while many endeavors have been invested to weight neighbors in terms of their importance. In [15], visual similarity was directly used as the voting weight. Zhu et al. [50] modeled the relationships among the neighbors by constructing a $k$-nearest neighbor graph, and an adaptive teleportation random walk was subsequently conducted over the graph to estimate the tag relevance. However, empirical results [50] showed that the performance gain obtained by these heuristic weighting strategies appears to be limited. In [36], the neighbor voting scheme was realized with the use of the contextual information of tag co-occurrence to boost the accuracy of tag-based image search. Nevertheless, there was no obvious improvement attained by directly incorporating the tag co-occurrence in the experiments [36]. In this paper, our approach explicitly models the neighbor weights and tag correlations, and thereby avoids the risk of making heuristic assumptions.

As an alternative to the neighbor voting algorithm, Liu et al. [24] first estimated the initial tag relevance based on the probability density estimation, and then performed a random walk over a tag similarity graph to refine the relevance score. In [45], Yang et al. studied how to establish the mapping between tags and image regions, i.e., to assign tags to image regions. They extended the group sparse coding technique with region spatial correlations to reconstruct each test region from the set of training regions. The tag localization task was then conducted by propagating tags from sparsely selected groups of training regions to the test region according to the reconstruction coefficients. In [23], image reconstruction and tag reconstruction were considered in parallel, and the resultant tag relevance scores produced by the two modalities were linearly combined.

### 2.3 Transduction-based methods

Different from model-based and instance-based methods producing rules or models that are directly applicable to a novel instance, this class of methods only performs reasoning from a given training set to a specific test set. There is no distinction between the training and test phase, and the output of transduction learning is used as the tag relevance score for a given image-tag pair. The majority of transduction-based methods are based on matrix factorization. Given the observed image-tag association matrix as input, the output is a reconstructed association matrix, the entries of which are the tag relevance scores. As an early effort, Liu et al. [25] refined the association matrix with an optimization framework based on the consistency between the visual similarity and the semantic similarity of social images. Similarly in [49], tag refinement was performed by optimizing the association matrix with the constraints of low-rank, error sparsity, content consistency and tag correlation. In [44], the

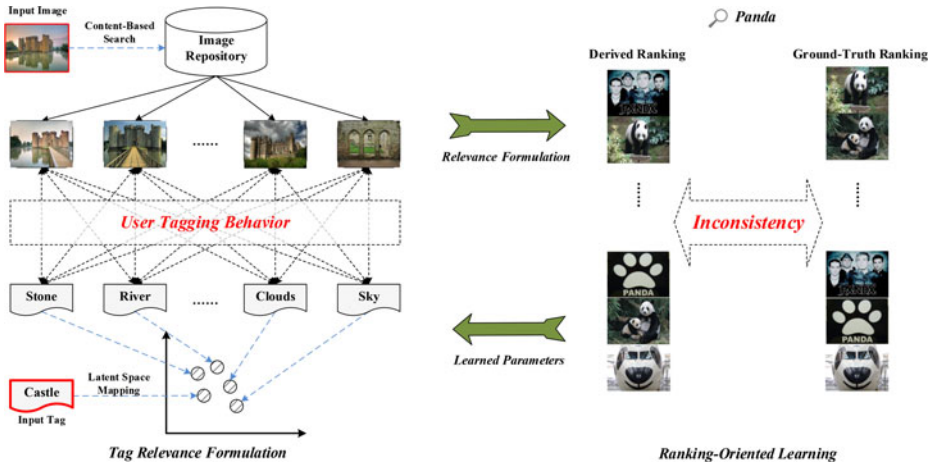


**Fig. 1** Architecture of the proposed ranking-oriented neighbor voting framework

authors searched for the optimal reconstructed association matrix that is consistent with the visual similarity matrix, the tag correlation matrix, and the original observed association matrix. In [27], the authors proposed a ranking-based multi-correlation tensor factorization method, to jointly model the ternary relations among users, images and tags, and further to reconstruct the user-aware image-tag associations.

Graph-based tag diffusion is another type of transduction-based methods. The basic idea is to construct a graph wherein each node corresponds to a specific image and each edge is weighted in terms of the similarity between images. The initial tag relevance score is assigned to each node, and all nodes then spread their relevance scores to their nearby neighbors via the weighted graph. In [40], the initial relevance score was estimated based on the similarity between the given tag and the tag set of the image, and the gaussian kernel was used to compute the visual similarity between images. Tang et al. [34] proposed to sparsely reconstruct an image from its neighbors in visual feature space, and the reconstruction coefficients were further used as similarity measurements to perform the graph-based tag diffusion. Despite encouraging results reported, the main problem of transduction-based methods lies in their insufficient capacity to adapt to the dynamic changes of social tagging systems. Once a novel image or tag is added, transduction-based methods may need to re-perform the learning process.

## 3 Framework

In this section, we present a novel framework to facilitate effective tag relevance learning. The architecture of our system is illustrated in Fig. 1. It consists of two main components: (1) tag relevance formulation and (2) ranking-oriented learning. With visual neighbors, a tag relevance function is formulated by explicitly modeling neighbor weights and tag correlations under the neighbor voting scheme. Based on the formulation, the ranking-oriented learning determines the parameters to directly optimize the ranking performance of tag-based image search. In the following, we elaborate on each of the components and give a full

**Table 1** Summary of symbols and definitions

| Symbols | Definitions |
|---|---|
| $\mathcal{X}$ | Social image collection |
| $\mathcal{T}$ | Tag vocabulary |
| $x$ | Tagged image, $x \in \mathcal{X}$ |
| $t_i$ | $i$-th tag in the vocabulary, $t_i \in \mathcal{T}$ |
| $z_l^x$ | $l$-th visual neighbor of $x$, $z_l^x \in \mathcal{X}$ |
| $v_l$ | Weight of $z_l^x$ |
| $w_{ij}$ | Correlation between $t_i$ and $t_j$ |
| $s_{t_i}$ | Number of images tagged with $t_i$ |
| $s$ | Number of total images, $s = \lvert\mathcal{X}\rvert$ |
| $m$ | Number of unique tags, $m = \lvert\mathcal{T}\rvert$ |
| $k$ | Number of visual neighbors |
| $p$ | Dimension of the latent space |
| $\mathbf{u}_i$ | Representation vector of $t_i$ in the latent space |
| $\mathbf{e}_i$ | Unit vector with 1 in the $i$-th position |
| $\mathcal{Y}$ | Set of all possible rankings over images |
| $Y$ | Ranking over images, $Y \in \mathcal{Y}$ |
| $Y_q^*$ | Ground-truth ranking of the images with respect to $t_q$, $Y_q^* \in \mathcal{Y}$ |
| $\mathcal{I}_{t_q}^+$ | Set of the relevant images with respect to $t_q$ |
| $\mathcal{I}_{t_q}^-$ | Set of the irrelevant images with respect to $t_q$ |
| $n$ | Number of training instances |
| $d$ | Dimension of image feature vector |

description of the associated algorithms. For clarity, we first list some important symbols and their definitions used throughout the paper in Table 1.

## 3.1 Tag relevance formulation

### *3.1.1 Visual neighbor search*

As a prerequisite to realize the neighbor voting scheme, we first need to find the visual neighbors of the given image. Visual neighbor search has been extensively studied across several communities, including multimedia, computer vision and machine learning [7]. Some works are concerned with developing fast indexing and matching techniques to speed up the search process [18]. Meanwhile, recent years have witnessed a surge of research efforts in distance metric learning [7], which applies machine learning techniques to optimize distance metrics by exploiting multi-modal information associated with images. It has been demonstrated that these techniques can significantly improve the performance in visual neighbor search, but the benefit also comes with the price of high requirements for computation and storage cost. Therefore, in this paper, we directly leverage content-based image search techniques to accomplish this task.

The related process comprises two steps: feature extraction and similarity measure. In the first step, we use five types of low-level visual features to represent each image. These

features include: (1) 64-dimensional color histogram, (2) 144-dimensional color correlogram, (3) 73-dimensional edge direction histogram, (4) 128-dimensional wavelet texture, and (5) 225-dimensional block-wise color moment. These features are the standard visual features provided in the benchmark dataset NUS-WIDE-LITE [6], and they can effectively characterize images from different perspectives of color, shape, and texture. Besides, these features are easily extracted and have been widely used by extensive studies, e.g., [24, 25, 40].

There are two general strategies to combine different features, namely the early fusion and the late fusion [33]. The early fusion strategy integrates individual features before learning tag relevance scores; while the late fusion strategy uses individual features to learn tag relevance scores separately, and then integrates the scores. In [16], a systematic analysis was presented on the two strategies in the context of neighbor voting model, and empirical results showed that there is no significant difference between them for tag relevance learning. However, a major disadvantage of the late fusion strategy is its expensiveness in terms of the learning efforts, as each kind of features requires a separate learning stage, which inevitably leads to much more computational cost. For this reason, the early fusion strategy is adopted in our approach. Specifically, we concatenate different visual features of an image into a single vector. Since the range of values varies greatly among different visual features, we separately normalize each dimension of the feature vector into the [0, 1] range. Afterwards, we use the Euclidean metric to measure the visual distance between images. Given an image, all images are ranked by their distance from it and the $k$ nearest neighbors are subsequently discovered.

### *3.1.2 Tag relevance function*

Our framework formulates a tag relevance function based on the neighbor voting scheme, where the relevance score of a tag is inferred by the tagging information of the visual neighbors of the given image. We first propose to explicitly model the individual weight of each visual neighbor. Given a tag $t_i \in \mathcal{T}$ and an image $x \in \mathcal{X}$, we define $r(t_i, x)$ as the relevance score of $t_i$ with respect to $x$:

$$r(t_i, x) = \sum_{l=1}^{k} v_l \varphi(z_l^x, t_i) \,, \tag{1}$$

where $z_l^x$ denotes the $l$-th nearest neighbor of $x$, and $\mathbf{v} \in \mathbb{R}^{k\times 1}$ is a vector of parameters whose $l$-th element $v_l$ indicates the weight of $z_l^x$. Note that we treat an image itself as its first nearest neighbor, i.e., $z_1^x = x$.

In (1), $\varphi(z_l^x, t_i)$ represents the voting power of $z_l^x$ concerning $t_i$. Previous works [15, 16] usually adopt an indicator function to represent the information, that is, the voting power allocated to a neighbor image is 0 or 1. However, this is an excessively coarse representation. The tags of an image cannot equally describe the visual contents, and a neighbor image should have distinct voting powers for different tags. Here, we set $\varphi(z_l^x, t_i)$ to be the presence probability of $t_i$ given $z_l^x$, which is approximately estimated by a multiple Bernoulli process with a beta prior:

$$\varphi(z_l^x, t_i) = \frac{\mu \delta_{z_l^x, t_i} + s_{t_i}}{\mu + s} \,, \tag{2}$$

where $\delta_{z_l^x, t_i}$ indicates the tagging observation on $z_l^x$, i.e., $\delta_{z_l^x, t_i} = 1$ if $z_l^x$ is tagged with $t_i$ in the image collection and zero otherwise. $\mu$ is a smoothing parameter associated with $\delta_{z_l^x, t_i}$. $s_{t_i}$ denotes the number of images tagged with $t_i$, and $s$ is the total number of images.

It is generally observed that better performance can be achieved by mining the information of tag correlations in many applications [25, 31, 39]. Inspired by this, we seek to exploit the potential of tag correlations in the context of neighbor voting. Co-occurrence statistics and WordNet similarity are the most commonly used correlation measurements. However, the study in [36] has shown that they do not provide an obvious benefit in tag relevance learning under the framework of neighbor voting. More importantly, apart from the positive correlations, there also exist many important negative correlations among tags. For instance, if the tag 'desert' has been assigned to an image, we may have high confidence that the tag 'fish' is irrelevant to the visual content of that image. Unfortunately, limited by their non-negative property, both co-occurrence statistics and WordNet similarity cannot reflect these potential negative correlations.

Given the drawbacks of existing correlation measurements, we further propose to explicitly model the pairwise tag correlations and incorporate them into the neighbor voting model as follows:

$$r(t_i, x) = w_{ii} \sum_{l=1}^{k} v_l \varphi(z_l^x, t_i) + \sum_{j=1, j\neq i}^{m} w_{ij} \sum_{l=1}^{k} v_l \varphi(z_l^x, t_j) \,, \tag{3}$$

where $W \in \mathbb{R}^{m\times m}$ is a parameter matrix whose $(i, j)$-th entry $w_{ij}$ captures the correlation between the tag $t_i$ and the tag $t_j$, and $w_{ii}$ represents the self-correlation of $t_i$. $m$ is the total number of unique tags. We assume that $W$ is a symmetric matrix, i.e., $w_{ij} = w_{ji}$, and both positive and negative values are allowed in $W$. It can be intuitively understood that, when estimating $r(t_i, x)$, we exploit not only the relevant confidence of $t_i$ inferred with the neighbors of $x$, but also the evidences provided by all the other tags. For the simplicity of the expression, we introduce a supplementary matrix $\Phi_x \in \mathbb{R}^{k\times m}$ whose $(l, j)$-th entry is equal to $\varphi(z_l^x, t_j)$. As a result, (3) can be written in a concise form:

$$r(t_i, x) = \mathbf{e}_i{}^T W {\Phi_x}^T \mathbf{v} \,. \tag{4}$$

A potential problem with the above formulation is that it requires the huge amount of parameters to capture the correlation between each pair of tags. From the viewpoint of statistical learning theory, too many parameters may degrade the model stability and generalization in performance. The existing work [47] on text information processing has demonstrated that the semantic space spanned by textual keywords can be approximated by a smaller set of *latent factors*. As one kind of text information, image tags are consequently subject to such low-rank property [49]. In accordance with this principle, we introduce a low-rank prior into the parameter $W$ with $W = U^T U$, which results in the new formulation of the relevance function as follows:

$$r(t_i, x) = \mathbf{e}_i{}^T U^T U {\Phi_x}^T \mathbf{v} \,, \tag{5}$$

where $U \in \mathbb{R}^{p\times m}$ and $p$ is the dimensionality of a latent space. Let $\mathbf{u}_i$ denote the $i$-th column of $U$, which actually corresponds to the representation vector of $t_i$ in the latent space. The correlation $w_{ij}$ is thus measured by the dot product of $\mathbf{u}_i$ and $\mathbf{u}_j$ in the latent space, which is commonly used to measure the matching between textual vectors. Because the intrinsic dimensionality of the latent space is typically much smaller than that of the original space (i.e., $p \ll m$), the number of parameters in (5) is significantly reduced.

### 3.2 Ranking-oriented learning

#### *3.2.1 Problem transformation*

Our framework seeks to learn the parameters $\mathbf{v}$ and $U$ in a supervised fashion. To this purpose, we first consider the relevance function $r$ as a ranking function for tag-based image search. The image ranking for a query tag is derived by descending the tag relevance of each image. As a result, the problem of tag relevance learning can be approached from a new perspective of learning to rank.

Without loss of generality, assume that a set of training instances for the first $n$ tags is available:

$$\{(t_q, Y_q^*) \in \mathcal{T} \times \mathcal{Y} : q = 1, \ldots, n\},$$

where $t_q$ is a query tag and $Y_q^*$ is the true ranking of the images with respect to $t_q$. $\mathcal{T}$ is the tag vocabulary and $\mathcal{Y}$ is the set of all possible rankings over images. Similar to previous work [46], we represent any ranking $Y \in \mathcal{Y}$ as a matrix of pair orderings, where the $(i, j)$-th entry $y_{ij} = +1$ if the image $x_i$ is ranked ahead of the image $x_j$, $y_{ij} = -1$ if $x_i$ is ranked behind $x_j$, and $y_{ij} = 0$ if $x_i$ and $x_j$ have equal rank. Note that $Y_q^*$ is a weak ranking with only two relevance levels, i.e., relevant and irrelevant. We denote by $\mathcal{I}_{t_q}^+$ and $\mathcal{I}_{t_q}^-$ the sets of the relevant and irrelevant images with respect to $t_q$.

Following the setup of learning to rank, our goal is transformed into learning a ranking hypothesis $h : \mathcal{T} \to \mathcal{Y}$. For the query tag $t_q$, $h(t_q)$ needs to correspond to the image ranking in descending order of $r(t_q, x)$. Towards this end, we first construct a compatibility function $f(t_q, Y) : \mathcal{T} \times \mathcal{Y} \to \mathbb{R}$, which measures how well a possible image ranking $Y$ fits for $t_q$:

$$\begin{aligned} f(t_q, Y) &= \sum_{x_i \in \mathcal{I}_{t_q}^+} \sum_{x_j \in \mathcal{I}_{t_q}^-} y_{ij} \left( \frac{r(t_q, x_i) - r(t_q, x_j)}{|\mathcal{I}_{t_q}^+| \cdot |\mathcal{I}_{t_q}^-|} \right) \\ &= \sum_{x_i \in \mathcal{I}_{t_q}^+} \sum_{x_j \in \mathcal{I}_{t_q}^-} y_{ij} \left( \frac{\mathbf{e}_q{}^T U^T U (\Phi_{x_i} - \Phi_{x_j})^T \mathbf{v}}{|\mathcal{I}_{t_q}^+| \cdot |\mathcal{I}_{t_q}^-|} \right). \end{aligned} \tag{6}$$

Then, $h(t_q)$ is defined by maximizing $f(t_q, Y)$ over all possible $Y \in \mathcal{Y}$:

$$h(t_q) = \arg\max_{Y \in \mathcal{Y}} f(t_q, Y) . \tag{7}$$

In (6), $f(t_q, Y)$ is decomposed into a series of pairwise components, i.e., $y_{ij}(r(t_q, x_i) - r(t_q, x_j))$. For a fixed $\mathbf{v}$ and $U$, $h(t_q)$ can be attained by maximizing each component individually: if $r(t_q, x_i) > r(t_q, x_j)$, $y_{ij}$ is set to $+1$; otherwise, it is set to $-1$. Note that this is the same procedure as sorting the images by $r(t_q, x)$, and $h(t_q)$ proves to be equivalent to the ranking in descending order of $r(t_q, x)$.

#### *3.2.2 Optimization formulation*

With the set of training instances, we can learn the ranking hypothesis $h(t_q)$ (i.e., the parameters $\mathbf{v}$ and $U$) by minimizing the empirical ranking risk,

$$R_\Delta(h) = \frac{1}{n} \sum_{q=1}^{n} \Delta(Y_q^*, h(t_q)) , \tag{8}$$

where the loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ quantifies the inconsistency between the derived ranking $h(t_q)$ and the true ranking $Y_q^*$. In this study, we associate $\Delta$ with specific ranking evaluation criteria, and the learning process is thus conducted by directly optimizing the ranking performance of tag-based image search. Specifically, we define $\Delta$ in our experiments based on the average precision (AP) score:

$$\Delta(Y_q^*, h(t_q)) = 1 - AP(Y_q^*, h(t_q)) \,, \tag{9}$$

and minimizing the empirical risk is equivalent to maximizing the measure of mean average precision (MAP). Note that other ranking evaluation criteria (e.g., Precision and NDCG) can also be incorporated into the loss function $\Delta$.

We adopt the structural SVM [13] as the backbone of our learning algorithm, since it supports the optimization of various ranking evaluation criteria under a unified framework. In (6), we notice that the parameters $\mathbf{v}$ and $U$ are independent of the summation indices, and thus $f$ can be rewritten as:

$$f(t_q, Y) = \left\langle U^T U \otimes \mathbf{v},\ \mathbf{e}_q \otimes \Psi(t_q, Y) \right\rangle_F \,, \tag{10}$$

$$\Psi(t_q, Y) = \sum_{x_i \in \mathcal{I}_{t_q}^+} \sum_{x_j \in \mathcal{I}_{t_q}^-} y_{ij} \frac{\Phi_{x_i} - \Phi_{x_j}}{|\mathcal{I}_{t_q}^+| \cdot |\mathcal{I}_{t_q}^-|} \,, \tag{11}$$

where $\otimes$ denotes the Kronecker product, and $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product. $\Psi(t_q, Y)$ encodes the *joint feature representation* of the input-output pair $(t_q, Y)$. By representing $f$ as a linear function of $\Psi(t_q, Y)$, the structural SVM is employed to learn $\mathbf{v}$ and $U$ through the following optimization problem [13]:

**Optimization Problem 1**

$$\min_{\mathbf{v}, U, \xi} \quad \frac{\lambda}{2} \|\mathbf{v}\|_2^2 + \frac{\lambda}{2} \|U\|_F^2 + \xi \tag{12}$$

$$\text{s.t.} \quad \forall (Y_1, \ldots, Y_n) \in \mathcal{Y}^n :$$

$$\frac{1}{n} \sum_{q=1}^{n} \left[ f(t_q, Y_q^*) - f(t_q, Y_q) \right] \geqslant \frac{1}{n} \sum_{q=1}^{n} \Delta(Y_q^*, Y_q) - \xi \,. \tag{13}$$

Different from the original structural SVM, our learning method needs to optimize $\mathbf{v}$ and $U$ simultaneously. Therefore, we replace the standard regularization term by $\frac{\lambda}{2} \|\mathbf{v}\|_2^2 + \frac{\lambda}{2} \|U\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius norm. $\xi$ is the only slack variable shared across all constraints. The constraints enforce the requirement that the average score for the true rankings should be greater than that for any other set of possible rankings. Note that the set $(Y_1^*, \ldots, Y_n^*)$ is not excluded from the constraints, because it corresponds to the non-negative constraint on $\xi$. It is demonstrated that under these constraints, $\xi$ is an upper bound on the empirical risk $R_\Delta$. As a result, the parameter $\lambda$ in the objective function essentially controls the tradeoff between the model complexity and the corresponding empirical risk.

### *3.2.3 Learning algorithm*

The main difficulty of Optimization Problem 1 lies in that there are as many as $|\mathcal{Y}|^n$ constraints to be considered. To solve it efficiently, we employ the cutting plane algorithm [13]. The basic principle of the cutting plane algorithm is to find a subset of constraints so that the solution for this subset can also satisfy all the constraints at an error tolerance of $\varepsilon$. The pseudo-code of the algorithm is presented in Algorithm 1. It starts with an empty working

set $\mathcal{W}$ of active constraints (step 1). Then it iteratively finds the $\widehat{Y}_q$ which generates the most violated constraint for each $t_q$ with current $\mathbf{v}$ and $U$ (step 3–5). If the corresponding constraints are on average violated by more than $\varepsilon$, the algorithm adds $(\widehat{Y}_1, \ldots, \widehat{Y}_n)$ into the working set $\mathcal{W}$, and re-optimizes (12) over the updated $\mathcal{W}$ (step 6–9). The algorithm terminates when no constraints are added into $\mathcal{W}$ anymore (step 10).

In Algorithm 1, there are two critical issues that remain to be addressed. One is how to search for the most violated constraint (step 4), i.e., to solve the maximization problem:

$$\widehat{Y}_q \leftarrow \arg\max_{Y \in \mathcal{Y}} \Delta(Y_q^*, Y) + f(t_q, Y) \ . \tag{14}$$

For the MAP-related loss in this study, this step can be accomplished with the algorithm presented in [46]. Although our problem involves two parameters $\mathbf{v}$ and $U$, the algorithm can still be easily adapted because the compatibility function $f$ is formulated into the same form as a linear function of $\Psi(t_q, Y)$. For other loss functions with different ranking measures, many methods have also been presented to solve the problem, such as [12] for Precision-related loss and [3] for NDCG-related loss.

**Algorithm 1** Cutting Plane Algorithm

**Input:** training instances $(t_1, Y_1^*), \ldots, (t_n, Y_n^*)$, regularization trade-off $\lambda$, error tolerance $\varepsilon$
**Output:** model parameters $\mathbf{v}$ and $U$, slack variable $\xi$

1: Initialize $\mathcal{W} \leftarrow \emptyset$
2: **repeat**
3: **for** $q = 1, 2, \ldots, n$ **do**
4: $\widehat{Y}_q \leftarrow \arg\max_{Y \in \mathcal{Y}} \Delta(Y_q^*, Y) + f(t_q, Y)$
5: **end for**
6: **if** $\frac{1}{n} \sum_{q=1}^{n} \Delta(Y_q^*, \widehat{Y}_q) - \frac{1}{n} \sum_{q=1}^{n} \left[ f(t_q, Y_q^*) - f(t_q, \widehat{Y}_q) \right] > \xi + \varepsilon$ **then**
7: $\mathcal{W} \leftarrow \mathcal{W} \cup \left\{ (\widehat{Y}_1, \ldots, \widehat{Y}_n) \right\}$
8: Optimize Eq. (12) over $\mathcal{W}$
9: **end if**
10: **until** $\mathcal{W}$ has no change during iteration

**Algorithm 2** Pegasos Algorithm

**Input:** working set $\mathcal{W}$, regularization trade-off $\lambda$, initialization values $\mathbf{v}_1$ and $U_1$, maximum iteration $T$
**Output:** model parameters $\mathbf{v}_{T+1}$ and $U_{T+1}$

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Find $(\widehat{Y}_1, \ldots, \widehat{Y}_n) \in \mathcal{W}$ achieving the largest margin
3: Set $\eta_t = 1/(\lambda t)$
4: Compute $\nabla_{\mathbf{v}_t}$ and $\nabla_{U_t}$
5: Update $\mathbf{v}_{t+1} = \mathbf{v}_t - \eta_t \nabla_{\mathbf{v}_t}$
$U_{t+1} = U_t - \eta_t \nabla_{U_t}$
6: Project $\mathbf{v}_{t+1} = \min \left\{ 1, \frac{1/\sqrt{\lambda}}{||\mathbf{v}_{t+1}||_2} \right\} \mathbf{v}_{t+1}$
$U_{t+1} = \min \left\{ 1, \frac{1/\sqrt{\lambda}}{||U_{t+1}||_F} \right\} U_{t+1}$
7: **end for**

**Table 2** Statistics of the experimental datasets

| Statistics | Dataset I | Dataset II |
|---|---|---|
| Number of images | 55,615 | 25,000 |
| Number of tags | 1,000 | 356 |
| Average number of tags per image | 7.34 | 2.08 |
| Number of concepts with ground-truth | 75 | 18 |

The other key issue is to re-optimize (12) when a new most violated constraint is added (step 8). Note that by exploiting the low-rank approximation, the problem becomes not convex. However, since the problem at this step only differs in a single constraint from iteration to iteration, we can restart the optimizer from the previous optimal values, which may greatly speed up the convergence rate. In this study, we implement a effective iterative algorithm adapted from the Pegasos algorithm [28] to solve the problem. At iteration $t$ of the algorithm, we first compute the subgradients with respect to $\mathbf{v}$ and $U$, respectively. Note that $\xi$ is the point-wise maximum of a set $\{\xi_1, \xi_2, \ldots\}$, where $\xi_i$ is the margin for the $i$th constraint in $\mathcal{W}$. Therefore, the subgradients can be computed in terms of the single constraint $(\widehat{Y}_1, \ldots, \widehat{Y}_n)$ that achieves the current largest margin:

$$\begin{aligned} \nabla_{\mathbf{v}} &= \lambda \mathbf{v} - \tfrac{1}{n} \sum_{q=1}^{n} \delta\Psi(t_q, \widehat{Y}_q) U^T U \mathbf{e}_q \\ \nabla_{\mathrm{U}} &= \lambda U - \tfrac{1}{n} \sum_{q=1}^{n} \left( \mathbf{v}^T \delta\Psi(t_q, \widehat{Y}_q) \otimes \mathbf{u}_q \right. \\ &\quad \left. + \mathbf{e}_q{}^T \otimes U \delta\Psi(t_q, \widehat{Y}_q)^T \mathbf{v} \right) \end{aligned} , \tag{15}$$

where $\delta\Psi(t_q, \widehat{Y}_q) = \Psi(t_q, Y_q^*) - \Psi(t_q, \widehat{Y}_q)$. Then, we update $\mathbf{v}$ and $U$ at iteration $t$ with the step size $\eta_t = 1/(\lambda t)$. Finally, we project $\mathbf{v}$ and $U$ onto the sphere of radius $1/\sqrt{\lambda}$ to accelerate the convergence of the algorithm (see [28]). Algorithm 2 describes these steps in detail.

Once the parameters $\mathbf{v}$ and $U$ are learned, the neighbor weights and tag correlations during the neighbor voting process are fixed. Given a new image, we first find its visual neighbors following the procedures stated in Section 3.1.1, and the relevance of each tag with respect to the new image can then be easily estimated by (5).

## 4 Experimental configuration

This section introduces the experiment configuration for our performance evaluation. All tag relevance learning methods evaluated in this study have been fully implemented in Matlab platform and tested on a server equipped with 2.20GHz Intel Xeon processor and 12GB RAM.

### 4.1 Data collections

To ensure accuracy and fairness of the empirical results, we adopt two benchmark image datasets that are collected from Flickr in our evaluation. On both datasets, some *concepts* have been manually labeled with ground-truth matching images. These concepts correspond to some tags in Flickr and can be used as query tags in our approach.

Dataset I is NUS-WIDE-LITE [6], which consists of 55,615 images with their associated tags. Since the tags provided in the dataset are rather noisy, a pre-processing step is performed to filter out these tags. Specifically, the collection of tags is limited to the ones appearing at least 50 times. Each tag is also matched with entries in the WordNet thesaurus and those tags that do not exist in the WordNet are subsequently removed. Finally, 1,000 tags are left and the average number of tags per image is 7.34. 75 concepts with their ground-truth labeled images are provided in the dataset.

Dataset II is MIRFlickr [11], which contains 25,000 images and 1,386 tags. We perform the same pre-processing steps as on Dataset I to filter out the tags, and 356 unique tags are consequently retained. Each image is annotated with an average of 2.08 tags. Ground-truth labeling for 18 concepts is also available in the dataset. Table 2 summarizes the basic information about our datasets.

### 4.2 Methodology and evaluation metrics

In order to conduct a comprehensive performance comparison of different methods, the proposed framework and the competitors are tested on the following two tasks.

#### *4.2.1 Task I: Tag-based image search*

We first evaluate on a tag-based search scenario. For a test query tag, we sort images by descending predicted tag relevance of each image. We study the performance of different methods with regard to the number of the visual neighbors, i.e., the hyper-parameter $k$.

For our approach, on Dataset I, we take half of the total concepts as query tags during training, and keep the rest for testing. On Dataset II, in order to verify the generalization ability of our approach, we conduct a cross-set evaluation as suggested by [22]. Specifically, we learn our model using the training data of Dataset I, and directly test it on Dataset II. In other words, all concepts on Dataset II are only used for testing.

We use the mean average precision (MAP) as the evaluation metric. Let $\pi^*$ be the ground-truth ranking and $\pi$ the predicted ranking, the average precision (AP) is defined as:

$$AP(\pi^*, \pi) = \frac{1}{r} \sum_{j:rel(j)=1} Precision@j \ ,$$

where $r$ is the number of relevant images in $\pi^*$, $Precision@j$ is the percentage of relevant images in the top $j$ images of $\pi$, and $rel(j)$ is an indicator function equaling 1 if the $j$th image in $\pi$ is relevant and zero otherwise. MAP is the mean of the average precision scores over all query tags.

#### *4.2.2 Task II: Automatic tag recommendation*

Although the proposed tag relevance learning framework is formulated to optimize the performance of tag-based image search, we still want to examine whether it is applicable to other applications. To this end, we compare different methods in the scenario of automatic tag recommendation.

The process of automatic tag recommendation does not require users to provide initial tags [24]. Given an image, different from the setting of previous methodologies [1, 22, 37]

**Table 3** Guidelines for rating a recommended tag for a test image

| Relevance Level | Description |
|---|---|
| Relevant | I affirm that the tag is saliently present in or applicable to the contents of the image. |
| Partially Relevant | I think that the tag is in some way relevant to the contents of the image, but could not be entirely confident. |
| Irrelevant | I believe that the tag is totally irrelevant to the contents of the image. |

where the recommendation candidates are restricted to only those concepts with ground-truth, in this evaluation, all tags are taken into account and ranked in descending order by their relevance regarding the image. The top 10 ranked tags are then added into the recommendation list. Through this way, we can test the generalization capability of our learned tag relevance model across a broad range of tags. We randomly choose 1,000 images from the two datasets as evaluation testbed. According to the scheme used in [11], five volunteers are invited to rate each recommended tag with one of the three relevance levels: Relevant (score 3), Partially Relevant (score 2) and Irrelevant (score 1). The guidelines for rating are shown in Table 3.

As the ground-truth tag ranking for a single image is not available, the discounted cumulative gain (DCG) is used to evaluate the quality of the recommendation list. The DCG at the $n$-th position is computed as:

$$DCG@n = \sum_{i=1}^{n} \frac{2^{rel(i)} - 1}{\log(1+i)} ,$$

where $rel(i)$ is the relevance level of the $i$-th tag. The average value of DCG@n ($n = 5, 10$) over all test images is reported to evaluate the overall performance.

### 4.3 Competitors for performance comparison

We compare our approach with several state-of-the-art methods on tag relevance learning. For these methods, the parameters are tuned via 5-fold cross validation. Specifically, the competitors are:

- *Baseline*: This method simply treats the raw user tagging information as a relevance indicator, i.e., the relevance of each tag to an image is 0 or 1.
- *NVote* [19]: NVote is the original neighbor voting algorithm which assesses tag relevance by counting the difference between tag frequency in the local neighbor set and the entire image collection.
- *WVote* [15]: WVote goes a step further than NVote by assigning different weights to the visual neighbors. The weight of a neighbor is set according to its visual distance to the given image.
- *Graph* [40]: Graph adopts a semi-supervised learning method to predict the tag relevance by leveraging the tag diffusion over the $k$-nearest neighbor graph of labeled and unlabeled images.
- *SVM* [17]: This method uses SVM to learn a binary classifier for each tag and estimates tag relevance by the classifier output score. The SVM classifier is trained with $\chi^2$ kernel, which has been shown to achieve superior performance for visual categorization.

– *TagProp* [38]: TagProp employs neighbor voting plus distance metric learning. It automatically finds the optimal combination of distances to define the visual neighbors, which are then assigned with rank or distance-based weights. The parameters in the model are learned by maximizing the likelihood of the annotations of training images.
– *RankVote*: Our proposed framework for tag relevance learning.

In task I, we compare the performance of all methods listed above. Note that our approach finds out reliable tags of images through tag relevance learning, and directly applies these tags as indexing keywords for tag-based image search. Our approach is different from those prior efforts, which focus on the automatic detection of indexing keywords from various contextual information of images. It is an important issue to study the automatic detection of indexing keywords, but the topic is beyond the scope of this paper. For this reason, the methods on the automatic detection of indexing keywords are excluded from the comparative study. In task II, SVM is excluded from the evaluation because of its excessive computational cost in the need of learning a separate classifier for each tag.

### 4.4 Parameter settings

There are several hyper-parameters in our framework. For the parameter $\mu$ in (2), we experiment with several values and find that the performance of our approach is not very sensitive to its change, so we set $\mu = 4s$ empirically where $s$ denotes the size of the image set. For the dimensionality of the latent space $p$ in (5), we use $p = 100$ for most of our results, and its effect on the performance will be discussed later. For the trade-off parameter $\lambda$ in (12), we choose $\lambda = 100$ via 5-fold cross validation on both datasets. In task II, the parameter $k = 300$ is fixed for all the competing methods, as suggested in [38].

## 5 Experiment results

### 5.1 Result for image search

Figure 2 shows the results of different methods for tag-based image search on Dataset I. Note that the variation of $k$ does not affect the performance of Baseline and SVM. Perhaps

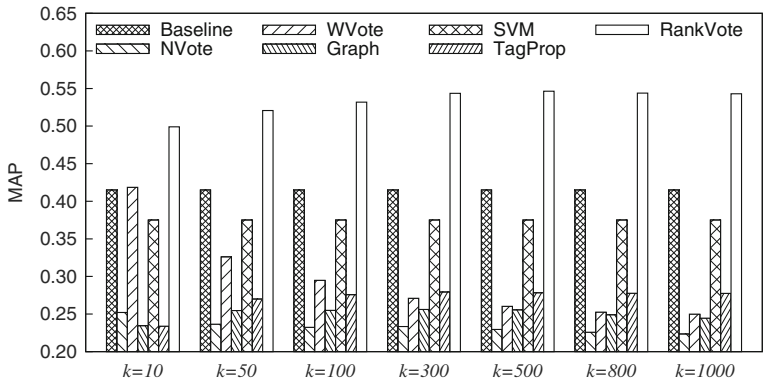


**Fig. 2** Performance comparison for tag-based image search on Dataset I with the variation of $k$

surprising, we notice that the existing approaches on tag relevance learning all fall behind Baseline with most values of $k$. In comparison to Baseline, they suffer at least 10 % relative drops on average. The results suggest that these approaches do not offer potential benefits to current tag-based search systems in terms of MAP. As stated in Introduction, we believe the reason lies in that they do not target at improving the performance of image search when predicting tag relevance, and thus the resultant relevance scores do not necessarily yield good search results. On the other hand, RankVote consistently achieves the best performance in all cases, reaching at least 20 % relative improvement over Baseline. To further analyze the results, we perform paired $t$-test [32] to compare the difference between RankVote and the other methods, and find that the improvement of RankVote is statistically significant at a significance level of 0.05. Besides, RankVote is also more robust to the number of visual neighbors. For example, even though tested with only 10 neighbors, RankVote still remains a relatively high performance level of 49.9 %. The best result is gained when $k$ is around 500, and the performance keeps relatively steady with larger values of $k$.

Figure 3 shows the comparison results on Dataset II. As expected, although we only use the model without training on Dataset II, RankVote still outperforms the other competitors with statistically significant improvement in most cases, which is also verified with paired $t$-test. For example, in the case of $k = 800$, on average, around 38.1 % relative improvement can be gained from RankVote. The findings indicate that the proposed method can learn a good tag relevance function and the learned relevance model can also be generalized well in real applications. Besides, it is clearly shown that all the existing approaches yield better performance than Baseline with different values of $k$. One possible reason is that the images are associated with relatively few tags on Dataset II, which increases the difficulty for Baseline to only rely on these tags to achieve the good search performance. The results also verify the necessity of tag relevance learning for tag-based image search, especially when the original tagging information is limited.

## 5.2 Result for tag recommendation

Table 4 reports the empirical results of different methods for automatic tag recommendation. It is clearly shown that RankVote outperforms the other approaches with statistical significance in both evaluation criteria. More precisely, the maximum relative increases are 36.4 % and 25.8 % in terms of DCG@5 and DCG@10, whereas the minimum gains still

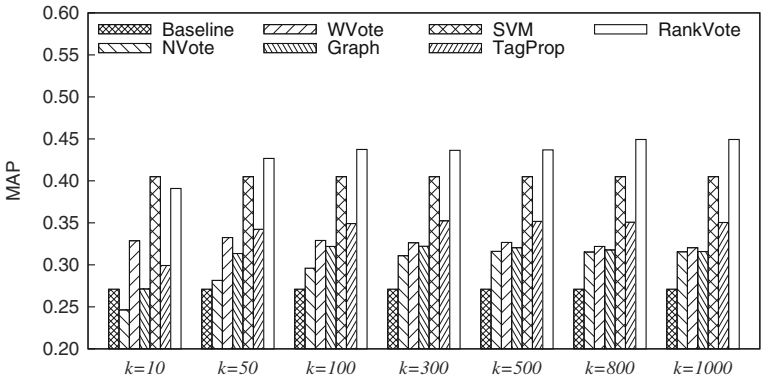


**Fig. 3** Performance comparison for tag-based image search on Dataset II.

**Table 4** Results of different methods for automatic tag recommendation, where the best performance is highlighted in boldface and + indicates it is significantly better than the runner-up by paired $t$-test

| | Methods | | | | | |
|---|---|---|---|---|---|---|
| Evaluation Criteria | Baseline | NVote | WVote | Graph | TagProp | RankVote |
| DCG@5 | 7.11 | 7.15 | 6.88 | 5.97 | 7.18 | **8.14**$^+$ |
| DCG@10 | 9.82 | 10.32 | 9.99 | 9.00 | 10.36 | **11.32**$^+$ |

achieve 13.4 % and 9.3 %, respectively. Moreover, the relative improvements on DCG@5 are more significant than those on DCG@10, which is a nice property as users are usually more interested in the top results in recommendation. From above, we can conclude that the proposed method is a highly effective technique for automatic tag recommendation.

Except for Baseline, all the methods compared above rely on the effectiveness of the visual neighbor search. Therefore, we further study how the methods behave at different accuracy levels of the visual neighbor search. Since manually assessing the accuracy of neighbor search results for each test image is laborious, we estimate the accuracy as follows. For each test image, we gather the recommended tags rated as "Relevant", and regard them as the ground-truth tags of the test image. Following the setting in [19], we consider a neighbor image accurate if it shares at least one of the ground-truth tags with the test image. In this way, we count the number of accurate neighbors and subsequently compute the accuracy of the visual neighbor search. We categorize the accuracy of the visual neighbor search into three levels, i.e., low (accuracy $< 0.05$), medium ($0.05 \leq$ accuracy $\leq 0.20$), and high (accuracy $> 0.20$).

Table 5 summarizes the performance comparison in terms of DCG@5 given different visual neighbor search accuracies. We can observe that RankVote substantially outperforms the other competitors when the neighbor search accuracy is low and medium, leading to more than 33.1 % and 7.5 % relative improvement, respectively. Note that the majority of the test images have unsatisfactory neighbor search results (40.8 % low and 45.4 % medium). Therefore, we believe that RankVote is more practical in real applications. In the environment of high neighbor search accuracy, all the methods obtain obvious performance gains, and the difference among their results is rather small. The observation implies that the higher accuracy in visual neighbor search plays a critical role in improving the performance for these methods.

**Table 5** Performance comparison in terms of DCG@5 given different neighbor search accuracies, where the best performance is highlighted in boldface and + indicates it is significantly better than the runner-up by paired $t$-test

| | Methods | | | | |
|---|---|---|---|---|---|
| Neighbor Accuracy | NVote | WVote | Graph | TagProp | RankVote |
| Low | 3.88 | 3.84 | 3.69 | 4.20 | **5.59**$^+$ |
| Medium | 8.14 | 7.73 | 6.31 | 8.17 | **8.78**$^+$ |
| High | 13.54 | 13.06 | 11.58 | 12.75 | **13.55** |

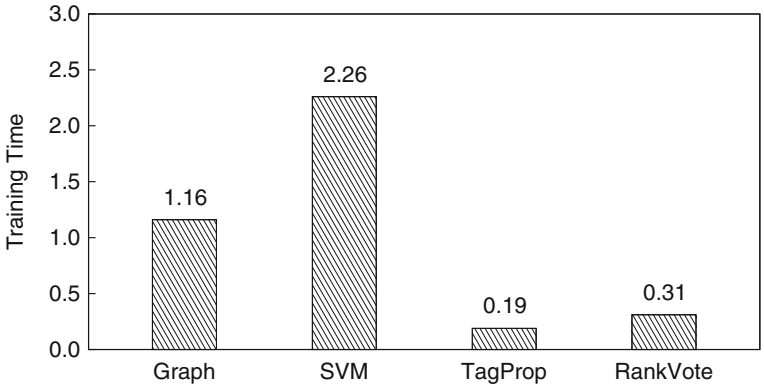


**Fig. 4** Training time comparison (in hours)

### 5.3 Analysis on computational efficiency

In this subsection, we examine the efficiency of the proposed framework. Our analysis leaves out the operations of visual feature extraction and tag preprocessing, which only need to be executed once in an offline manner.

To build upon the neighbor voting scheme, an imperative computation is to search the visual neighbors from the image collection, which has a complexity of $O(sd + k \log s)$ per image. Note that this complexity is drawn from a straightforward implementation of the nearest neighbor search, and it can also be substantially reduced by adopting more efficient techniques [26]. The most computational cost of our approach results from the time for training. Theoretically, the cutting plane algorithm, as shown in Algorithm 1, loops for $O(\frac{1}{\lambda\varepsilon})$ iterations. In actual experiments, it usually terminates within 5 iterations under $\lambda = 100$ and $\varepsilon = 0.01$. In each iteration, the algorithm for searching the most violated constraint is called $O(n)$ times, whose time complexity is $O(s \log s)$ [46]. To further evaluate the efficiency quantitatively, we report the runtime of training phase for RankVote in comparison with that of the other supervised competitors, i.e., Graph, SVM and TagProp on Dataset I. The runtime is measured with $k = 500$, since all the competitive methods yield relatively good performance under this setting. Figure 4 lists the empirical results of different methods. Clearly, RankVote shows substantial reduction in the runtime of training phase in comparison with Graph and SVM. Compared with TagProp, RankVote takes over 1.5 times longer for training, but it has a significant superiority in accuracy as shown in Figs. 2 and 3. We believe the gain outweighs the loss. Moreover, it is worth noting that the training process of our algorithm can still be speeded up by adopting the recently developed more efficient optimizers [35] or using the standard C implementation[3] of SVM$^{struct}$.

Once training is completed, given a test image, our approach can first find its nearest neighbors in $O(sd + k \log s)$ time, and then predict the relevance of each tag with respect to the image within $O(mk)$ time. According to the measured elapsed time during testing, we find that our approach takes an average of 0.53 seconds to produce the image ranking for a

[3] http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html

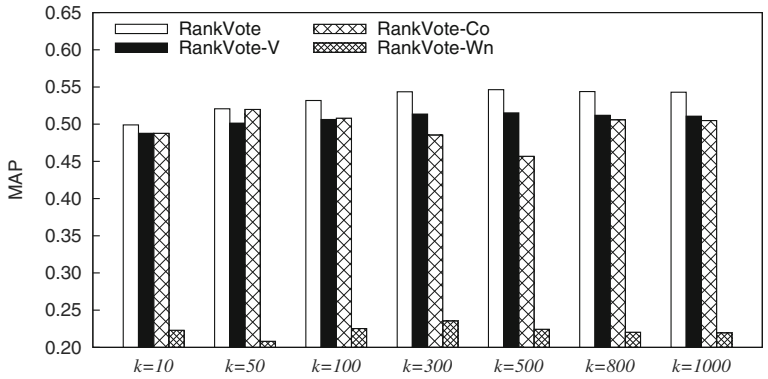


**Fig. 5** Comparison between different variants of our framework on Dataset I

query tag in the task of tag-based image search. This means that our trained model can be used interactively by users without any perceived delay.

### 5.4 Benefit from explicit modeling

In the proposed framework, both neighbor weights and tag correlations are explicitly modeled simultaneously. To investigate the efficacy of each modeling component, we design the following three variants of our original framework, and compare them with RankVote in the task of tag-based image search:

- *RankVote-V*: This method is a simplified version of RankVote that only models the neighbor weights. That is, the tag relevance function is defined only with the parameter $\mathbf{v}$ as in (1).
- *RankVote-Co*: Different from RankVote explicitly learning the tag correlations in a supervised manner, this method simply uses the co-occurrence statistics as the correlation measurement.
- *RankVote-Wn*: It is similar to RankVote-Co, but uses the WordNet similarity instead of the co-occurrence statistics.

Figure 5 summarizes the empirical study on Dataset I. In comparison with the results of NVote and WVote shown in Fig. 2, we can see that RankVote-V enjoys an average of 117.6 % and 76.6 % relative increases, respectively. The results point clearly to the importance of explicitly modeling the neighbor weights for the neighbor voting scheme. When adopting the existing measurements to capture the tag correlations, RankVote-Co does not exhibit substantial performance gains over RankVote-V, whereas RankVote-Wn even experiences a sharp degradation in performance and falls far behind RankVote-V. As we mentioned in Introduction, the significant degradation of RankVote-Wn may result from the fact that WordNet similarity cannot directly reflect how people tag images. Many tags frequently appearing together in social tagging are often weakly related according to the WordNet ontology. On the contrary, RankVote consistently maintains superior performance over RankVote-V, resulting in more than 5.1 % relative gains on average, which is statistically significant at a significance level of 0.05. The above results confirm the importance of explicitly modeling the tag correlations to achieve the effectiveness of tag relevance learning.

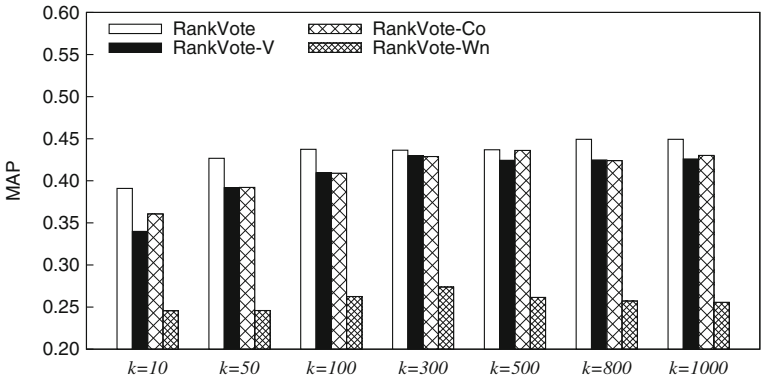


**Fig. 6** Comparison between different variants on Dataset II

Figure 6 shows the comparison results on Dataset II. Again, RankVote outperforms the variants with different values of $k$. One thing worth noting is that the average relative gains of RankVote over RankVote-V increase from 5.1 % on Dataset I to 6.7 % on Dataset II. We believe that the gains arise because RankVote uses twice the number of training instances on Dataset II during training, from which it can get additional hints to model the tag correlations more accurately. This observation also reveals that when sufficient training data is available, it is necessary to exploit both information of neighbor weights and tag correlations for obtaining a desirable model.

### 5.5 Illustration of learned parameters

To the best of our knowledge, this is the first study that effectively estimates the tag relevance, while jointly learning the neighbor weights and tag correlations in a unified framework. In this subsection, we investigate the latent properties of the neighbor weights and tag correlations discovered by our approach.

Figure 7 plots the learned weights of the visual neighbors on a log-log scale. The x-axis represents the sequence of the top 1,000 nearest neighbors, and the y-axis refers to the individual weight of each neighbor. We can see that the variation trend of neighbor weights is in accordance with the intuition that close neighbors are more important than distant ones.

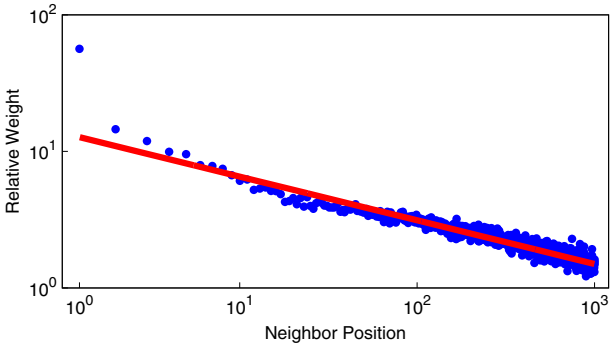


**Fig. 7** Illustration of the learned individual weight of each visual neighbor

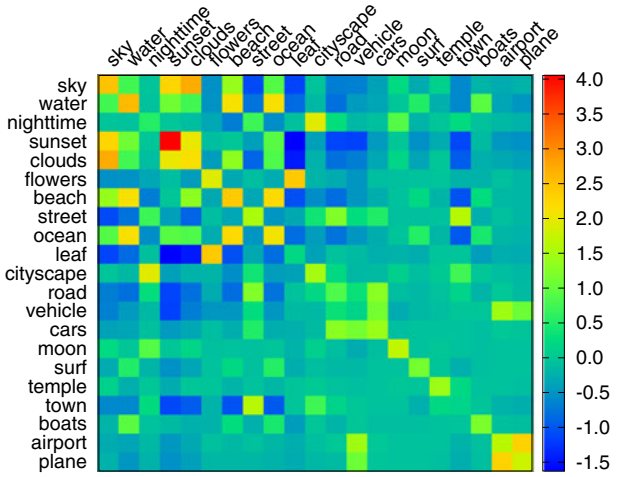


**Fig. 8** Illustration of the learned pairwise correlations between tags

More precisely, we find that the weight of a neighbor conditioned on its rank in the neighbor sequence can be well fitted by a power-law relationship, which has been indicated by a red line. Intuitively, this property implies that the relevance of a tag is dominantly determined by its relevant confidence regarding the top nearest neighbors. It also explains why even though with a limited number of neighbors, our approach can still maintain acceptable performance as shown in Figs. 2 and 3. In addition, we notice that the weight of the first nearest neighbor greatly exceeds those of the other neighbors. Recall that the first nearest neighbor is fixed to be the given image itself. Its high weight suggests that, although many of them may be inaccurate and subjective, the user-generated tags of an image are still the most valuable evidences for tag relevance learning. We believe the above findings are helpful to guide further research on neighbor voting.

Figure 8 illustrates the learned pairwise correlations among a group of frequent tags, where a color map is used to indicate the magnitude of the correlations. Note that the range of the learned correlation values is asymmetric between the positive and negative sides. As expected, the relatively higher values are assigned to the diagonal elements which represent the self-correlation of each tag. Among the pairwise elements, the higher correlations are also assigned to the pairs of tags that commonly co-occur such as (sky, clouds), (ocean, beach) and (leaf, flowers), or the tags with the same or similar meanings such as (road, street) and (water, ocean). On the other hand, those rarely co-occurring tags like (sunset, leaf) and (ocean, town) are assigned with lower negative correlation values. From the figure, we may conclude that the learned correlations can properly encode the various kinds of relationships among tags.

### 5.6 Effect of latent space dimension

To gain a good understanding of the performance of the parameter $p$ used in (5), which denotes the dimensionality of the latent space, we conduct a robustness test to examine whether the key results remain consistent when changing $p$ value from 10 to 200. 5-fold cross validation is applied to the experiment. Due to space limits, we only report the experiments for tag-based image search. Figure 9 plots the results, where five curves fluctuate,

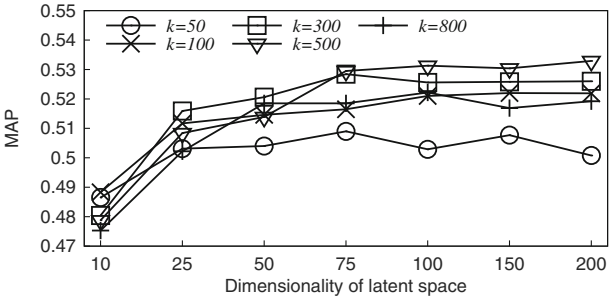


**Fig. 9** Effect of $p$ for tag-based image search.

indicating the effects of $p$ when using $k$ = 50, 100, 300, 500 and 800, respectively. All curves start from a low MAP value and gradually go up with the increase of $p$. The performance peaks when $p$ is 75 or 100 and keeps relatively steady with larger $p$ values. This demonstrates the robustness of our approach to the changes in $p$ values.

## 6 Conclusions

In this paper, we investigated the problem of tag relevance learning from a new perspective of learning to rank, and sought the improved accuracy through introducing a supervision step into the neighbor voting scheme. The individual weight of each visual neighbor was explicitly modeled, and the pairwise correlations between tags were also captured through a low-rank approximation. The learning process was conducted by directly optimizing the ranking performance of tag-based image search. Extensive experiments were conducted on two benchmark datasets in comparison with the state-of-the-art methods. Experimental results demonstrated the effectiveness of our approach.

Our future work will focus on three directions. Firstly, we plan to gather additional training instances for more robust modeling. Secondly, we intend to incorporate the distance metric learning techniques for discovering more accurate visual neighbors. Finally, we will further exploit more efficient optimization algorithms such that the proposed approach can work on larger datasets.

**Acknowledgments** This work is supported by the Natural Science Foundation of China (61272240, 71402083, 61103151), the Doctoral Fund of Ministry of Education of China (20110131110028), and the Natural Science Foundation of Shandong province (ZR2012FM037). Dr. Jialie Shen is supported by Singapore Ministry of Education under Academic Research Fund Tier-2 (MOE Ref: MOE2013-T2-2-156).

## References

1. Ballan L, Bertini M, Uricchio T, Del Bimbo A (2014a) Data-driven approaches for social image and video tagging. Multimedia Tools and Applications 74(4):1443–1468
2. Ballan L, Uricchio T, Seidenari L, Del Bimbo A (2014b) A cross-media model for automatic image annotation. In: Proceedings of ACM International Conference on Multimedia Retrieval, pp 73–80
3. Chakrabarti S, Khanna R, Sawant U, Bhattacharyya C (2008) Structured learning for non-smooth ranking losses. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 88–96

4. Chen L, Xu D, Tsang IW, Luo J (2012) Tag-based image retrieval improved by augmented features and group-based refinement. IEEE Transactions on Multimedia 14(4):1057–1067
5. Cheng Z, Shen J, Miao H (2014) The effects of multiple query evidences on social image retrieval. Multimedia Systems. doi:10.1007/s00530-014-0432-7
6. Chua TS, Tang J, Hong R, Li H, Luo Z, Zheng Y (2009) Nus-wide: a real-world web image database from national university of singapore. In: Proceedings of ACM International Conference on Image and Video Retrieval, pp 48:1–48:9
7. Cui C, Ma J, Lian T, Chen Z, Wang S (2015) Improving image annotation via ranking-oriented neighbor search and learning-based keyword propagation. Journal of the Association for Information Science and Technology 66(1):82–98
8. Feng S, Feng Z, Jin R (2015) Learning to rank image tags with limited training examples. IEEE Trans Image Process 24(4):1223–1234
9. Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Mikolov T et al (2013) Devise: A deep visual-semantic embedding model. In: Proceedings of Advances in Neural Information Processing Systems, pp 2121–2129
10. Gong Y, Ke Q, Isard M, Lazebnik S (2014) A multi-view embedding space for modeling internet images, tags, and their semantics. Int J Comput Vis 106(2):210–233
11. Huiskes M, Lew M (2008) The mir flickr retrieval evaluation. In: Proceedings of ACM International Conference on Multimedia Information Retrieval, pp 39–43
12. Joachims T (2005) A support vector method for multivariate performance measures. In: Proceedings of the International Conference on Machine learning, pp 377–384
13. Joachims T, Finley T, Yu C (2009) Cutting-plane training of structural svms. Mach Learn 77(1):27–59
14. Kennedy LS, Chang SF, Kozintsev IV (2006) To search or to label?: Predicting the performance of search-based automatic image classifiers. In: Proceedings of ACM International Workshop on Multimedia Information Retrieval, pp 249–258
15. Lee S, De Neve W, Ro YM (2014) Visually weighted neighbor voting for image tag relevance learning. Multimedia tools and applications 72(2):1363–1386
16. Li X (2014) Tag relevance fusion for social image retrieval. Multimedia Systems. doi:10.1007/s00530-014-0430-9
17. Li X, Snoek CG (2009) Visual categorization with negative examples for free. In: Proceedings of ACM International Conference on Multimedia, pp 661–664
18. Li X, Chen L, Zhang L, Lin F, Ma WY (2006) Image annotation by large-scale content-based image retrieval. In: Proceedings of ACM International Conference on Multimedia, pp 607–610
19. Li X, Snoek C, Worring M (2009) Learning social tag relevance by neighbor voting. IEEE Trans Multimed 11(7):1310–1322
20. Li X, Snoek C, Worring M (2010) Unsupervised multi-feature tag relevance learning for social image retrieval. In: Proceedings of ACM International Conference on Image and Video Retrieval, pp 10–17
21. Li X, Snoek CG, Worring M, Koelma D, Smeulders AW (2013) Bootstrapping visual categorization with relevant negatives. IEEE Trans Multimed 15(4):933–945
22. Li X, Uricchio T, Ballan L, Bertini M, Snoek CGM, Bimbo AD (2015) Socializing the semantic gap: a comparative survey on image tag assignment, refinement and retrieval. CoRR arXiv:1503.08248
23. Lin Z, Ding G, Hu M, Wang J, Ye X (2013) Image tag completion via image-specific and tag-specific linear sparse reconstructions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 1618–1625
24. Liu D, Hua XS, Yang L, Wang M, Zhang HJ (2009) Tag ranking. In: Proceedings of ACM International Conference on World Wide Web, pp 351–360
25. Liu D, Hua XS, Wang M, Zhang HJ (2010) Image retagging. In: Proceedings of ACM International Conference on Multimedia, pp 491–500
26. Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: Proceedings of the International Conference on Computer Vision Theory and Applications, pp 331–340
27. Sang J, Xu C, Liu J (2012) User-aware image tag refinement via ternary semantic analysis. IEEE Trans Multimed 14(3):883–895
28. Shalev-Shwartz S, Singer Y, Srebro N, Cotter A (2011) Pegasos: Primal estimated sub-gradient solver for svm. Math Program 127(1):3–30
29. Shen J, Wang M, Yan S, Hua XS (2011) Multimedia tagging: past, present and future. In: Proceedings of ACM International Conference on Multimedia, pp 639–640
30. Shen Y, Fan J (2010) Leveraging loosely-tagged images and inter-object correlations for tag recommendation. In: Proceedings of ACM International Conference on Multimedia, pp 5–14

31. Sigurbjrnsson B, Van Zwol R (2008) Flickr tag recommendation based on collective knowledge. In: Proceedings of ACM International Conference on World Wide Web, pp 327–336
32. Smucker MD, Allan J, Carterette B (2007) A comparison of statistical significance tests for information retrieval evaluation. In: Proceedings of ACM International Conference on Information and Knowledge Management, pp 623–632
33. Snoek CGM, Worring M, van Gemert JC, Geusebroek JM, Smeulders AWM (2006) The challenge problem for automated detection of 101 semantic concepts in multimedia. In: Proceedings of ACM International Conference on Multimedia, pp 421–430
34. Tang J, Hong R, Yan S, Chua TS, Qi GJ, Jain R (2011) Image annotation by knn-sparse graph-based label propagation over noisily tagged web images. ACM Transactions on Intelligent Systems and Technology 2(2):14:1–14:15
35. Teo CH, Vishwanthan S, Smola AJ, Le QV (2010) Bundle methods for regularized risk minimization. J Mach Learn Res 11:311–365
36. Truong BQ, Sun A, Bhowmick SS (2012) Content is still king: the effect of neighbor voting schemes on tag relevance for social image retrieval. In: Proceedings of ACM International Conference on Multimedia Retrieval, pp 9:1–9:8
37. Uricchio T, Ballan L, Bertini M, Del Bimbo A (2013) An evaluation of nearest-neighbor methods for tag refinement. In: Proceedings of IEEE International Conference on Multimedia and Expo, pp 1–6
38. Verbeek J, Guillaumin M, Mensink T, Schmid C (2010) Image annotation with tagprop on the mirflickr set. In: Proceedings of ACM International Conference on Multimedia Information Retrieval, pp 537–546
39. Wang J, Zhou J, Xu H, Mei T, Hua XS, Li S (2014) Image tag refinement by regularized latent dirichlet allocation. Comput Vis Image Underst 124:61–70
40. Wang M, Yang K, Hua X, Zhang H (2010a) Towards a relevant and diverse search of social images. IEEE Trans Multimed 12(8):829–842
41. Wang Z, Feng J, Zhang C, Yan S (2010b) Learning to rank tags. In: Proceedings of ACM International Conference on Image and Video Retrieval, pp 42–49
42. Weston J, Bengio S, Usunier N (2011) Wsabie: Scaling up to large vocabulary image annotation. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp 2764–2770
43. Wu L, Yang L, Yu N, Hua X (2009) Learning to tag. In: Proceedings of ACM International Conference on World Wide Web, pp 361–370
44. Wu L, Jin R, Jain AK (2013) Tag completion for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(3):716–727
45. Yang Y, Yang Y, Huang Z, Shen H, Nie F (2011) Tag localization with spatial correlations and joint group sparsity. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp 881–888
46. Yue Y, Finley T, Radlinski F, Joachims T (2007) A support vector method for optimizing average precision. In: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, pp 271–278
47. Zhao R, Grosky W (2002) Narrowing the semantic gap-improved text-based web document retrieval using visual features. IEEE Trans Multimed 4(2):189–200
48. Zhou B, Jagadeesh V, Piramuthu R (2014) Conceptlearner: Discovering visual concepts from weakly labeled image collections. CoRR abs/1411.5328, arXiv:1411.5328
49. Zhu G, Yan S, Ma Y (2010) Image tag refinement towards low-rank, content-tag prior and error sparsity. In: Proceedings of ACM International Conference on Multimedia, pp 461–470
50. Zhu X, Nejdl W, Georgescu M (2014) An adaptive teleportation random walk model for learning social tag relevance. In: Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, pp 223–232