Multimed Tools Appl
DOI 10.1007/s11042-013-1811-3

Image auto-annotation via tag-dependent random search over range-constrained visual neighbours

Zijia Lin · Guiguang Ding · Mingqing Hu

Abstract The quantity setting of visual neighbours can be critical for the performance of many previously proposed visual-neighbour-based (VNB) image auto-annotation methods. And in those methods, each candidate tag of a to-be-annotated image would be better to have its own trustworthy part of visual neighbours for score prediction. Hence in this paper we propose to use a constrained range rather than an identical and fixed number of visual neighbours for VNB methods to allow more flexible choices of neighbours, and then put forward a novel tag-dependent random search process to estimate the tag-dependent trust degrees of visual neighbours for each candidate tag. We further propose an effective image auto-annotation method termed TagSearcher based on a widely-used conditional probability model for auto-annotation, considering image-dependent weights of visual neighbours, tag-dependent trust degrees of visual neighbours and votes for a candidate tag from visual neighbours. Extensive experiments conducted on both a benchmark dataset and real-world web images present that the proposed TagSearcher can yield inspiring annotation performance and also reduce the performance sensitivity to the quantity setting of visual neighbours.

Keywords Image auto-annotation · TagSearcher · Tag-dependent random search · Range-constrained visual neighbours

Z. Lin (✉)
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
e-mail: linzijia07@tsinghua.org.cn

G. Ding
School of Software, Tsinghua University, Beijing, 100084, China
e-mail: dinggg@tsinghua.edu.cn

M. Hu
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China
e-mail: humingqing@ict.ac.cn

Published online: 03 January 2014

1 Introduction

In recent years, with the prevalence of social network and digital photography, billions of Internet users are allowed to upload and share their pictures on the Internet, leading to the explosion of the number of web images. For example, more than 250 billion images have been uploaded to the popular social network Facebook[1] and more than 350 million images are uploaded every day on average [1]. To handle the large-scale and rapidly-increasing web images, effective techniques for image management and retrieval are necessitated.

Generally, web images can be retrieved by their associated texts or their content. For text-based image retrieval (TBIR), given a textual query, images are retrieved according to the relevance between their associated texts and the query. The associated texts of an image are usually obtained by exploiting its contextual information, like the metadata, surrounding textual descriptions, and manually labelled tags, etc. Yet unfortunately, there are innumerable web images associated with little or even no available contextual information, and thus TBIR will not work for them. For content-based image retrieval, low-level features, which can be global features like color histogram or local feature points like SIFT [21], are extracted from each image to describe its content. And then given a query image, indexed images will be retrieved according to the similarities between their features and those of the query one. Yet due to the so-called “semantic gap” between low-level features and high-level semantic concepts, the former may not express the latter exactly and thus the retrieved images could sometimes be irrelevant though visually similar in features.

Image auto-annotation, which aims to automatically and objectively assign an image with appropriate textual tags describing its semantic content, can be a potential approach to overcoming the weak points of both TBIR and CBIR. By adding semantically related tags to those images with little or no available contextual information, auto-annotation can enable TBIR to work for them. Moreover, image auto-annotation seems to be a promising solution to bridging the mentioned semantic gap by mapping the low-level features into intermediate textual tags, which, as revealed by A.G. Hauptmann [8], can be less difficult to be further mapped into high-level semantic concepts. And thus image auto-annotation has been attracting much attention from both academia and industry for years.

Previous researches on image auto-annotation can be roughly categorized into model-based [2–4, 6, 9, 12, 16, 27] and visual-neighbour-based (VNB) methods [7, 13, 15, 23, 30–32, 34]. Model-based methods perform image auto-annotation by modelling the relation between the image features and the tags, with generative models such as topic model, or discriminative models like multi-label classifier. Then to annotate an unlabelled image, the learnt model will determine which tags can be selected, based on their relations to the features. Though effective and elegant, model-based methods are generally sophisticated and may need a time-consuming model learning process, especially when the training set is quite large. Differently, visual-neighbour-based methods perform image auto-annotation via propagating tags from visually similar images. Specifically, to annotate an unlabelled image, VNB methods will firstly retrieve its visual neighbours and take their associated tags as candidates for annotation. Then by estimating a confidence score for each candidate tag to be associated with the image in some certain manner, they will take those tags with higher scores as the annotation result. Visual neighbours of an image are actually images that are visually similar, containing some identical objects. And they are generally determined by the similarity between image features. Recently, with the large-scale and rapidly-increasing

[1] See: https://www.facebook.com/

web images, VNB methods tend to be more attractive and preferable due to their concision and effectiveness.

For most previous VNB methods, to annotate an unlabelled image, they would generally take a fixed quantity of the most similar images as its visual neighbours. The quantity setting of visual neighbours is generally critical, since insufficient neighbours cannot provide enough tag information for exploiting while redundant ones probably introduce much noise. And thus the performance of VNB methods can be sensitive to the quantity setting of visual neighbours. Moreover, different to-be-annotated images can even have different image-dependent optimal quantity settings. Therefore, in this paper we propose to utilize a constrained range rather than an identical and fixed number of visual neighbours for tag propagation, introducing a strong upper bound and a weak upper bound to constrain the number of visual neighbours. Both bounds respectively determine the strongly-related and the weakly-related ranges of visual neighbours for each to-be-annotated image. Note that the latter range will always cover the former one. Neighbours in the strongly-related range are supposed to be reliable for exploiting tag information, while those out of the weakly-related range are assumed to be unhelpful. And thus the optimal quantity settings for different to-be-annotated images are supposed to mostly lie between both bounds. Compared with seeking an identical optimal quantity setting for all to-be-annotated images, it may be more effortless and reasonable to determine the bounds. And in this paper, visual neighbours retrieved in the proposed way with introduced range constraint are termed range-constrained visual neighbours.

Additionally, most previous VNB auto-annotation methods assume that the probabilities for visual neighbours to be selected are identical for all candidate tags when predicting their scores for a to-be-annotated image. In this paper, however, we propose that the probabilities are better to be tag-dependent, and denote them as the tag-dependent "trust degrees" of visual neighbours *w.r.t* a candidate tag. The proposal is based on a widely-used conditional probability model for image auto-annotation, as will be explained later. Here to facilitate understanding, we give an intuitive illustration in Fig. 1, where the score of a candidate tag "tree" for a to-be-annotated image is being predicted. We can see that only the 1st and the 3rd visual neighbours are labelled with "tree". Then to predict its score, both neighbours are expected to be more likely to be selected for exploiting the tag information, meaning that both are more trustworthy neighbours for "tree" and could derive higher trust degrees. In the same case, assuming that the probabilities for visual neighbours to be selected are the same for all candidate tags, as most previous VNB methods did, may implicitly weaken the positive contributions of trustworthy neighbours (e.g. the 1st and the 3rd ones) and strengthen the negative effects of less trustworthy ones (e.g. the 2nd one), resulting in tag-dependent noise for score prediction.

Therefore, in this paper we propose that the probabilities for visual-neighbours to be selected should be tag-dependent, which is termed tag-dependent trust degrees of visual neighbours, meaning that each candidate tag can have its own trustworthy part of neighbours for score prediction. Furthermore, we put forward a novel tag-dependent random search process over the range-constrained visual neighbours to estimate their tag-dependent trust degrees *w.r.t* each candidate tag. With the range constraint for visual neighbours and the tag-dependent random search process, we further propose an effective and robust image auto-annotation method, TagSearcher, based on a widely-used conditional probability model for auto-annotation, considering image-dependent weights of visual neighbours, tag-dependent trust degrees of visual neighbours and votes for a candidate tag from visual neighbours.

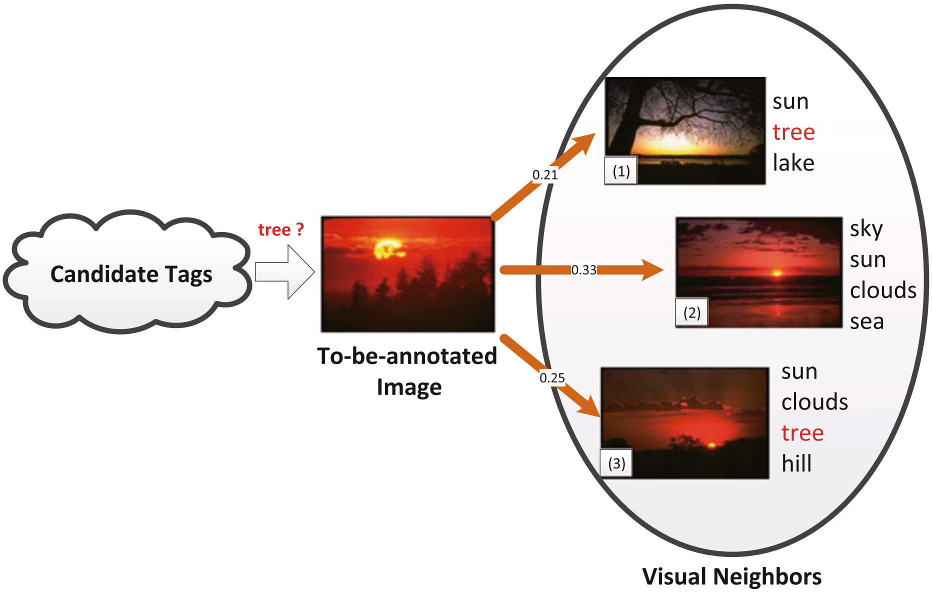

Fig. 1 An illustration of our proposal that the probabilities for visual neighbours to be selected for predicting tag scores are better to be tag-dependent, with higher values on edges corresponding to higher probabilities

Specifically, to annotate an image, the proposed TagSearcher will firstly retrieve its strongly-related range and weakly-related range of visual neighbours, with the former covered by the latter. Each retrieved neighbour will derive an image-dependent weight based on its visual similarity to the to-be-annotated image. Then for any candidate tag in the vocabulary, all neighbours will give their votes for it, according to the correlations of their corresponding annotations with the candidate tag. Moreover, a tag-dependent random search process will be performed over the range-constrained visual neighbours to search for its trustworthy part in the weakly-related range of visual neighbours, and estimate the trust degrees of all neighbours *w.r.t* it. Then the predicted score of the candidate tag will be estimated by considering and merging the weights of visual neighbours, the votes for it from visual neighbours, and the tag-dependent trust degrees of visual neighbours *w.r.t* it. With the scores of all candidate tags estimated, TagSearcher will rank all candidate tags according to their estimated scores in descending order, and take the *top N* as the annotation result for the to-be-annotated image.

The main contributions of our work can be summarized as follows.

1. We propose to utilize a constrained range rather than an identical and fixed number of visual neighbours in VNB auto-annotation methods to allow more flexible choices of neighbours and help to reduce the performance sensitivity.
2. We propose that the probabilities for visual neighbours to be selected are better to be tag-dependent, which are termed the tag-dependent trust degrees of visual neighbours. And we further propose a novel tag-dependent random search process over the range-constrained visual neighbours to derive their trust degrees *w.r.t* each candidate tag.
3. We propose an effective and robust image auto-annotation method based on a widely-used conditional probability model, considering image-dependent weights of visual

neighbours,tag-dependent trust degrees of visual neighbours and votes for a candidate tag from visual neighbours.

The remainder of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 elaborates on the proposed image auto-annotation method. Section 4 presents the details of experiments, including experimental settings, results and analyses. Finally we conclude the paper in Section 5.

2 Related work

As mentioned formerly, previous researches on image auto-annotation can be roughly categorized into model-based [2–4, 6, 9, 12, 16, 27] and visual-neighbour-based methods [7, 13, 15, 23, 30–32, 34].

Model-based methods focus on modelling the relation between image features and tags, with generative models or discriminative models. Among the generative models, Jeon et al. [9] proposed a cross-media relevance model to learn the joint distribution of features and tags for images. Feng et al. [6] further proposed a generative learning approach for image auto-annotation based on multiple Bernoulli relevance model. Among the discriminative models, Chang et al. [3] proposed a content-based soft annotation procedure via training an ensemble of binary classifiers for predicting label membership for images. Carneiro et al. [2] proposed a probabilistic formulation for image auto-annotation by defining it as a classification problem with each tag being a class. Though effective and elegant, model-based methods are generally sophisticated and may need a time-consuming model learning process, especially when the training set is quite large. Differently, visual-neighbour-based (VNB) methods perform auto-annotation via propagating tags to a to-be-annotated image from its visual neighbours [7, 13, 15, 23, 30–32, 34]. Generally, VNB methods are based on the assumption that both feature space and tag space share some hidden consistent semantic structures and visually similar images will probably share common tags. Recently, with the large-scale and rapidly increasing web images, VNB methods tend to be more attractive and preferable due to their concision and effectiveness. In [13], X. Li et al. proposed to search visual neighbours from web images and utilize a search result clustering technique to find most representative keywords for annotation. Wang et al. [32] further proposed a divide-and-conquer framework for auto-annotation, which identifies the salient terms from textual descriptions of visual neighbours searched from web images and then filtered out the noisy ones. In [23], Makadia et al. proposed a greedy auto-annotation method termed JEC, which simply associates a to-be-annotated image with frequent tags of its nearest visual neighbours. Wang and Zhang [30, 31] proposed a graph-based semi-supervised approach for tag propagation, which derives the weights of neighbours via an optimal linear reconstruction with features. Guillaumin et al. [7] utilized metric learning for better weight distribution among visual neighbours and trained tag-specific discriminative models for auto-annotation, which until now maintains the state-of-the-art performance.

By surveying previous VNB auto-annotation methods, we realize that nearly all of them utilize an identical and fixed number of visual neighbours for different to-be-annotated images, and the quantity setting of neighbours can be critical for the annotation performance. As insufficient visual neighbours cannot provide enough tag information for exploiting while redundant ones probably introduce much noise, it would be difficult to determine the optimal quantity setting of neighbours for any to-be-annotated image. Moreover, even different to-be-annotated images can have their own optimal quantity settings. Therefore, we propose in this paper to use a constrained range of visual neighbours in VNB methods to cover different optimal quantity settings for different to-be-annotated images, which is relatively more effortless and reasonable. Additionally, we find that most previous VNB methods assume that the probabilities for visual neighbours to be selected for score prediction are identical for all candidate tags. Yet according to the widely-used conditional probability model for auto-annotation and the illustration given formerly, the probabilities are better to be tag-dependent. And thus we further propose a novel tag-dependent random search process over the range-constrained visual neighbours to derive their tag-dependent trust degrees *w.r.t* each candidate tag, expecting to enhance the annotation performance and its robustness.

The proposed tag-dependent random search process is in some way similar to random walk, which is a well-known optimization method for graphical models and widely-used in the field of image analysis. Liu et al. [17] performed random walk over tag similarity graphs for tag ranking. Liu et al. [19] used random walk over an image similarity graph as a main approach to image auto-annotation. In [18, 28], random walk over tag similarity graphs was utilized for refining annotation results. Following these previous researches, we can build a graphical model with the range-constrained visual neighbours, with vertices corresponding to images and edges corresponding to image relations. To exploit the graphical model for estimating the tag-dependent trust degrees of visual neighbours, random walk seems unsuitable to be applied, due to that it is basically tag-independent, as will be detailed later and demonstrated by experiments. And thus in this paper we propose an alternative, the tag-dependent random search process, which determines the directions of random search by considering both image-image similarities and tag-tag correlations, and it is well demonstrated by experiments to be superior to random walk in this case.

3 Proposed TagSearcher

3.1 Overview

Given a labelled image database, to annotate an unlabelled image, the proposed TagSearcher will firstly retrieve its visual neighbours and use their tags as candidates for annotation. Then TagSearcher estimates the conditional probability of a candidate tag t_i given the to-be-annotated image I as many previous auto-annotation methods [6, 9, 20], i.e. $P\left(t_i|I\right)$. As the probability of the appearance of I, i.e. $P(I)$, is constant for all candidate tags, we can derive the following formula.

$$P\left(t_i|I\right) = \frac{P(I, t_i)}{P(I)} \propto P\left(I, t_i\right) \tag{1}$$

Following previous VNB methods, we assume that the association between a candidate tag and the to-be-annotated image, i.e. $P\left(I, t_i\right)$, can be inferred by the visual neighbours. Then based on the law of total probability, we derive the following formula.

$$P\left(I, t_i\right) \sim \sum_{I_j \in \mathbb{VN}(I)} P\left(I_j\right) P\left(t_i, I|I_j\right) \tag{2}$$

where $\mathbb{VN}(I)$ is the set of visual neighbours of the to-be-annotated image I, and $P\left(I_j\right)$ denotes the probability of the appearance of I_j. Furthermore, as the probability $P\left(t_i, I|I_j\right)$ would be difficult to be directly estimated, it is generally decomposed as follows.

$$P\left(t_i, I|I_j\right) = \xi\left(I, t_i, I_j\right) P\left(t_i|I_j\right) P\left(I|I_j\right) \tag{3}$$

where $\xi\left(I, t_i, I_j\right)$ is a compensation factor for the decomposition, since t_i and I are probably not conditionally independent given I_j. Then by considering all the formulas above together, we can finally derive the following formula for estimating the conditional probability $P(t_i|I)$.

$$P(t_i|I) \sim \sum_{I_j \in \mathbb{VN}(I)} \left(P\left(I_j\right)\xi\left(I, t_i, I_j\right)\right) P\left(t_i|I_j\right) P\left(I|I_j\right) \tag{4}$$

To facilitate understanding, the conditional probabilities $P\left(I|I_j\right)$ is termed the image-dependent weight of I_j, as it is generally estimated with image-image similarities, and $P\left(t_i|I_j\right)$ is termed the vote for t_i from I_j. Moreover, $P\left(I_j\right)\xi\left(I, t_i, I_j\right)$ as an integral can be seen as an expression of the probability for I_j to be selected for predicting the score of t_i given I.

Generally, previous VNB methods made an assumption that t_i and I are conditionally independent given I_j, meaning that $\xi\left(I, t_i, I_j\right) = 1$ and $P\left(I_j\right)\xi\left(I, t_i, I_j\right) = P\left(I_j\right)$. Moreover, they generally assume $P\left(I_j\right)$ to be a uniform prior probability, which would be an identical constant value for any I_j. Hence they simplified formula (4) as $P(t_i|I) \sim \sum_{I_j \in \mathbb{VN}(I)} P\left(t_i|I_j\right) P\left(I|I_j\right)$ and focused their work on estimating $P\left(t_i|I_j\right)$ or $P\left(I|I_j\right)$. However, according to formula (4) and the illustration given formerly, i.e. Fig. 1, these strong assumptions concerning t_i, I and $P\left(I_j\right)$ in previous VNB methods may not be reasonable. And thus in this paper we propose that the probability for a visual neighbour to be selected for score prediction, i.e. $P\left(I_j\right)\xi\left(I, t_i, I_j\right)$, should be tag-dependent, which in this paper is termed the tag-dependent trust degree.

By considering and estimating image-dependent weights of visual neighbours, votes for a candidate tag from visual neighbours and tag-dependent trust degrees of visual neighbours, the proposed TagSearcher predicts the score of a candidate tag t_i for the to-be-annotated image I using the following formula.

$$s(I, t_i) = \sum_{I_j \in \mathbb{U}(I)} w\left(I, I_j\right) v\left(I_j, t_i\right) c\left(I_j, I, t_i\right) \tag{5}$$

Here $s(I, t_i)$ is the predicted score of t_i, which is expected to be proportional to the conditional probability $P(t_i|I)$, $\mathbb{U}(I)$ is the weakly-related range of visual neighbours and $w\left(I, I_j\right)$, $v\left(I_j, t_i\right)$, $c\left(I_j, I, t_i\right)$ are respectively the estimated $P\left(I|I_j\right)$, $P\left(t_i|I_j\right)$ and $P\left(I_j\right)\xi\left(I, t_i, I_j\right)$. Then candidate tags with higher predicted scores are selected to label the to-be-annotated image. Note that to exploit more helpful information for the to-be-annotated image, here we utilize its weakly-related range of visual neighbours, i.e. $\mathbb{U}(I)$, rather than the strongly-related one. And by introducing the tag-dependent trust degrees of visual neighbours, i.e. $c\left(I_j, I, t_i\right)$, negative effects of noise in the weakly-related range are expected to be automatically minimized. The estimated image-dependent weight of I_j, i.e. $w\left(I, I_j\right)$, is derived based on its visual distance to I and rank position among all neighbours. And the estimated vote for t_i from I_j, i.e. $v\left(I_j, t_i\right)$, is derived under a conditional probability model considering tag correlations. As for the estimated tag-dependent trust degree of I_j *w.r.t* t_i, i.e. $c\left(I_j, I, t_i\right)$, it is derived with the proposed tag-dependent random search process over the range-constrained visual neighbours, considering both visual similarities and tag correlations. Since in formula (5) the image-dependent weights of visual neighbours and votes for a candidate tag from visual neighbours, i.e. $w\left(I, I_j\right)$ and $v\left(I_j, t_i\right)$, have been extensively studied in many previous auto-annotation methods [6, 7, 9, 12, 18, 20], in this paper we will focus our work on estimating the tag-dependent trust degrees of visual neighbours, i.e. $c\left(I_j, I, t_i\right)$, and just utilize some naïve methods to estimate $w\left(I, I_j\right)$ and $v\left(I_j, t_i\right)$, so as to better present the importance and effectiveness of introducing $c\left(I_j, I, t_i\right)$.

3.1.1 Image-dependent weights of visual neighbours

Similar to previous VNB auto-annotation methods, to determine the visual neighbours of any to-be-annotated image, the given labelled images will be ranked according to the distances between their corresponding feature vectors and that of the to-be-annotated image in ascending order, and those ranked at the top will be selected. Considering that it may be difficult and less reasonable to determine an optimal quantity setting of visual neighbours for all to-be-annotated images, the proposed TagSearcher resorts to utilizing a constrained range rather than an identical and fixed number of neighbours. Specifically, we set a strong upper bound and a weak one for the quantity settings, respectively corresponding to a strongly-related range and a weakly-related range of visual neighbours. The strongly-related range is supposed to be reliable for exploiting tag information though it may be insufficient. And the weakly-related range, which completely covers the strongly-related one, can be much larger and richer in tag information though it probably contains much more noise. As for images ranked out of the weak upper bound, they are assumed to be unrelated to the to-be-annotated image. Therefore, when estimating image-dependent weights of visual neighbours, it is supposed to be performed over the whole weakly-related range, and images ranked out of the range will be assigned with zero weights. In the proposed TagSearcher, image-dependent weights of visual neighbours are estimated with the following formula.

$$w\left(I, I_j\right) = \frac{1}{d\left(I, I_j\right)} \log\left(\frac{U+1}{j}\right) \tag{6}$$

where $w\left(I, I_j\right)$ is the estimated weight of the visual neighbour I_j for the to-be-annotated image *I*, *U* and *j* are respectively the weak upper bound and the rank position of I_j among the neighbours of *I*, and $d\left(I, I_j\right)$ is the visual distance between *I* and I_j. To better differentiate the image-dependent weights of visual neighbours, here we propose that the formula above should be both distance-based and rank-based, which assigns larger weights to neighbours ranked at the top. It is because that being only distance-based cannot well differentiate the weights of visual neighbours when their corresponding distances from the to-be-annotated image are similar, while being only rank-based cannot well handle the case that images ranked nearby are largely different in their corresponding distances from the to-be-annotated image.

3.1.2 Votes for a candidate tag

When estimating the votes for a candidate tag from visual neighbours, it is intuitive for neighbours containing the tag to return 1, and 0 otherwise. However, it can be better to take tag correlations into consideration and give a soft vote for the otherwise case, since tag correlations are generally important clues for exploiting valuable tag information, as revealed by previous work [17, 18, 28]. And thus in this paper we utilize a conditional probability model as follows to estimate the vote for a candidate tag from a visual neighbour not containing it.

$$v\left(I_j, t_i\right) \sim P\left(t_i \middle| \left\{t_{j1}, t_{j2}, ..., t_{jn}\right\}\right) \; s.t. \; t_i \notin \left\{t_{j1}, t_{j2}, ..., t_{jn}\right\} \tag{7}$$

where $\left\{t_{j1}, t_{j2}, \ldots, t_{jn}\right\}$ is the set of tags contained in the visual neighbour I_j. With the assumption of tag correlations, the conditional probability cannot be directly factorized and calculated by treating each tag independently. Moreover, as t_i and $\left\{t_{j1}, t_{j2}, \ldots, t_{jn}\right\}$ rarely appear together, a direct estimation of the conditional probability with frequencies of both tag sets in the given labelled image database will probably introduce serious biases. Therefore, in the proposed TagSearcher we resort to using the average vote for t_i from the tags contained in I_j as the estimated $v(I_j, t_i)$, shown as follows.

$$v\left(I_j, t_i\right) \sim \frac{1}{n}\sum_{k=1}^{n} v\left(t_{jk}, t_i\right) \sim \frac{1}{n}\sum_{k=1}^{n} P\left(t_i \mid t_{jk}\right) \sim \frac{1}{n}\sum_{k=1}^{n} \frac{|t_i \cap t_{jk}|}{|t_{jk}|} \tag{8}$$

where n is the number of tags contained in I_j and $v\left(t_{jk}, t_i\right)$ is the estimated vote for the candidate tag t_i from t_{jk}. Similarly, $v\left(t_{jk}, t_i\right)$ can be seen as a conditional probability between tags, i.e. $P\left(t_i \mid t_{jk}\right)$, which is further approximated with tag frequencies as [26], i.e. $|t_i \cap t_{jk}|$ and $|t_{jk}|$ where the former is the number of images containing both t_i and t_{jk} while the latter is the number of images containing t_{jk}. Then the votes for a candidate tag from visual neighbours can be estimated with the following integral formula.

$$v\left(I_j, t_i\right) = \begin{cases} 1, & t_i \in \left\{t_{j1}, t_{j2}, \ldots, t_{jn}\right\} \\ \frac{1}{n}\sum_{k=1}^{n} \frac{|t_i \cap t_{jk}|}{|t_{jk}|}, & t_i \notin \left\{t_{j1}, t_{j2}, \ldots, t_{jn}\right\} \end{cases} \tag{9}$$

3.1.3 Tag-dependent trust degrees of visual neighbours

As proposed formerly, different candidate tags can have their own selection of trustworthy neighbours for score prediction. A more trustworthy visual neighbour of a candidate tag is an image that has stronger evidence for the appearance of the tag, which is determined based on both its visual similarity to the to-be-annotated image and the correlations between its associated tags and the candidate tag. In this paper, we present an effective approach termed tag-dependent random search to estimating the tag-dependent trust degrees of visual neighbours for each candidate tag.

As illustrated in Fig. 2, to annotate an image, a graphical model with images as vertices, is firstly built with its visual neighbours in the weakly-related range (i.e. the dashed box). Then for each candidate tag, to estimate the tag-dependent trust degrees of visual neighbours, a random search process starting from the to-be-annotated image will be performed over the weakly-related range of neighbours. Note that after the first step, the to-be-annotated image is left out. Then at each subsequent step, the proposed random search process will determine the probability of moving forward from each vertex, which is tag-dependent and varies with the step. Specifically, if a vertex is labelled with the candidate tag, the process will stay at the vertex after reaching it. Otherwise, the process will determine whether to move forward or stay, by considering both the depth of search step and the correlations between the candidate tag and the contained tags of the vertex. When moving forward from a vertex, the process will randomly choose one of its strongly-related neighbours as a successive vertex and move on. Note that here each visual neighbour also has its own strongly-related range of neighbours, which is constrained within the weakly-related range of the to-be-annotated image to ensure the visual similarity of the found trustworthy neighbours. The proposed tag-dependent random search process can be demonstrated to be convergent as

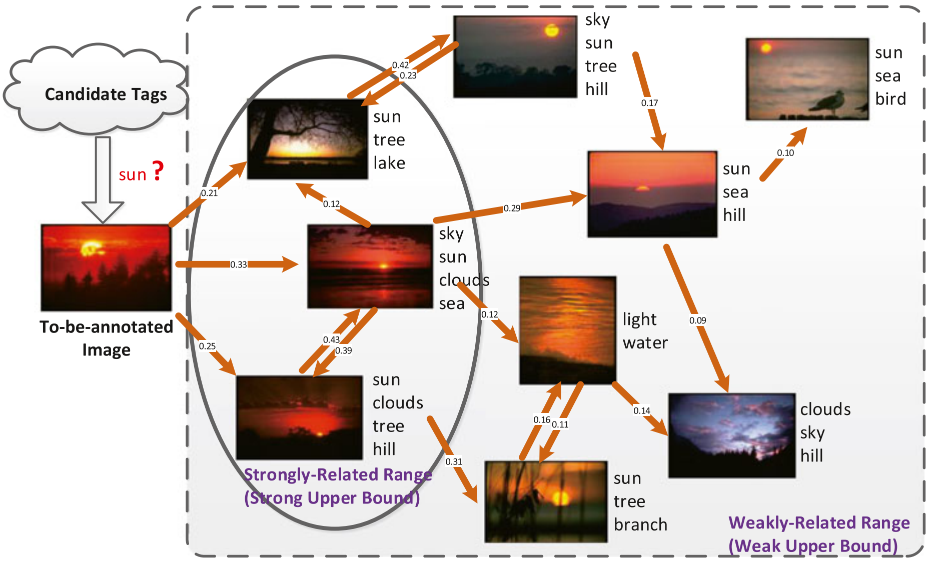

Fig. 2 An illustration of the proposed tag-dependent random search process over the range-constrained visual neighbours for estimating their tag-dependent trust degrees *w.r.t* the candidate tag "sun". Note that edges with zero weights in the graphical model above are omitted for clarity

it moves on, and the probability for the process to stay at each vertex will be utilized for estimating the trust degree of corresponding visual neighbour *w.r.t* the candidate tag.

In following subsections, we will further elaborate on the proposed tag-dependent random search process, and compare it with the well-known random walk.

3.2 Tag-dependent random search

As illustrated in Fig. 2, the built graphical model for tag-dependent random search is directed, where a vertex V_a denotes the corresponding visual neighbour I_a and the weight on a directed edge denotes the probability for one to choose the other as the successive vertex in a further step. Note that in the graphical model, all vertices except the to-be-annotated image can be the successive vertex of others. And the to-be-annotated image can be seen as the source of the graphical model. In the proposed TagSearcher, the weight on a directed edge is estimated as follows, similar to the estimation of image-dependent weights of visual neighbours.

$$s_{a,b} = \begin{cases} \frac{\eta}{d(I_a,I_b)} \log\left(\frac{U+1}{r(I_a,I_b)}\right), & I_b \in \mathbb{V}(I_a) \\ 0, & I_b \notin \mathbb{V}(I_a) \end{cases} \tag{10}$$

where $s_{a,b}$ is the derived weight on the directed edge from V_a to V_b, U is the weak upper bound for visual neighbours, $\mathbb{V}(I_a)$ is the strongly-related range of visual neighbours of I_a, $d(I_a, I_b)$ is the visual distance between low-level features of I_a and I_b, $r(I_a, I_b)$ is the rank position of I_b among the neighbours of I_a, and η is a normalizing factor to guarantee that all the weights on the edges from a vertex will sum up to 1. As revealed by formula (10), for any vertex in the graphical model, only neighbours in the strongly-related range will be chosen as a successive vertex. It is because that noisy data may bring considerable risky biases to the random search process, and thus here we cautiously utilize the more reliable range of neighbours. Based on the built graphical model, we can derive a matrix $\mathbf{S}_{U\times U}$ representing the successive relations between neighbours, of which the entry $\mathbf{S}_{ij}$ is the weight on the directed edge from V_i to V_j, i.e. the probability for V_i to choose V_j as its successive vertex in a further step. Assuming that at first only the source of the graphical model is assigned with 1 while others with 0, we can derive the expectation values of other vertices after the first step, denoted as a U-dimensional initial value vector $\mathbf{p}^{(0)}$. Apparently, $\mathbf{p}_i^{(0)}$ equals the weight on the edge from the source to V_i, and $\mathbf{p}^{(0)}$ sums up to 1. After that, the proposed tag-dependent random search process can be further performed for estimating the trust degrees of visual neighbours.

3.2.1 Tag-dependent random search at a specific step

To better present the inside details of the proposed tag-dependent random search process, here we firstly focus on calculating the expectation values of all vertices at a specific step. Specifically, the expectation value of a vertex at the kth step is calculated by considering both its initial value and the values it derived from other vertices at the last step, as shown in the following formula.

$$\mathbf{p}_i^{(k)} = \delta \mathbf{p}_i^{(0)} + (1-\delta) \sum_{j \leqslant U, j \neq i} \mathbf{p}_j^{(k-1)} \mathbf{f}_j^{(k-1)} \mathbf{S}_{ji}^{(k-1)} \tag{11}$$

where $\mathbf{p}_i^{(0)}$ is the initial value of V_i, $\mathbf{p}_j^{(k-1)}$ is the expectation value of V_j at the $(k-1)$th step, $\mathbf{f}_j^{(k-1)}$ and $\mathbf{S}_{ji}^{(k-1)}$ are respectively the probability of moving forward from V_j and the probability for V_i to be chosen as the successive vertex of V_j at the $(k-1)$th step, and δ is a weighting parameter between 0 and 1 for balancing the initial value and the derived values from other vertices. As the initial value vector $\mathbf{p}^{(0)}$ is calculated based on image similarities, which have been considered for estimating image-dependent weights of visual neighbours, here we propose to set δ with a small value. Note that at the $(k-1)$th step, the matrix $\mathbf{S}^{(k-1)}$ denoting the successive relations between vertices is derived in nearly the same way as formula (10), while the number of candidate successive vertices decreases by a ratio $1/\lambda$ ($\lambda > 1$) as the step goes deeper, in order to avoid reaching too many less-related neighbours.

The probability of moving forward from each vertex, i.e. $\mathbf{f}_j^{(k-1)}$, is critical to finding the more trustworthy visual neighbours for a candidate tag. As implied formerly, the probabilities of moving forward from vertices that are labelled with the candidate tag or strongly correlated tags will respectively be zero or a small value, meaning that the tag-dependent random search process will probably choose to stay at such vertices. For clarity, the probability for the tag-dependent random search process to stay at a vertex is termed the staying probability at the vertex. Note that for each vertex, the staying probability and the probability of moving forward will always sum to 1. Then it is evident that the staying probability at any vertex labelled with the candidate tag or strongly correlated tags will be relatively high, as the corresponding probability of moving forward is low. According to formula (11), it can be seen that such vertices will transfer only small parts of their values to other vertices in a further step of the tag-dependent random search process, and thus they will finally get larger expectation values. On the contrary, vertices with weakly correlated tags will get lower staying probabilities and finally derive smaller expectation values. Hence the expectation values of vertices can well reflect their corresponding trust degrees *w.r.t* the candidate tag. To estimate the probability of moving forward from each vertex, both tag-tag correlations and image-image similarities are considered, as will be detailed in the following paragraphs. And thus the tag-dependent trust degrees of visual neighbours are actually determined by both.

In the proposed tag-dependent random search process *w.r.t* a candidate tag $\widehat{t}$, the probability of moving forward from a vertex V_j at the kth step is estimated as follows.

$$\mathbf{f}_j^{(k)} = \begin{cases} 0, & \widehat{t} \in \left\{t_{j1}, t_{j2}, \ldots, t_{jn}\right\} \\ 1 - \frac{\alpha_j \exp(k)}{\alpha_j \exp(k)+\beta_j}, & \widehat{t} \notin \left\{t_{j1}, t_{j2}, \ldots, t_{jn}\right\} \end{cases} \tag{12}$$

where α_j is the conditional probability of the appearance of $\widehat{t}$ given the corresponding image I_j and is approximated as the vote for $\widehat{t}$ from I_j in our experiments, i.e. $v(I_j, \widehat{t})$ in formula (9). And β_j is the expectation value of the conditional probability at a further step, which is estimated as follows with the law of total probability.

$$\beta_j = \sum_{m \leqslant U, m \neq j} \mathbf{S}_{jm}^{(k)} \alpha_m \tag{13}$$

It can be seen from formula (12) and (13) that a larger β_j will lead to a larger $\mathbf{f}_j^{(k)}$, meaning that the random search process tends to move forward and seek more trustworthy neighbours at the further step. Otherwise it is more probable for the process to stay at the vertex. It can be seen that the probability of staying at a vertex for the proposed random search process is tag-dependent and varies with visual neighbours. And it also increases with the step of random search due to the introduced weighting factor "exp (k)" in formula (12) for setting higher staying probabilities in deeper steps (i.e. larger k) to avoid reaching less-related images.

With the probability of moving forward from each vertex at the kth step, i.e. $\mathbf{f}^{(k)}$, a diagonal matrix $\mathbf{F}^{(k)}$ can be derived with its diagonal entries set as $\mathbf{F}_{ii}^{(k)} = \mathbf{f}_i^{(k)}$. Since the diagonal entries of the matrix $\mathbf{S}^{(k-1)}$ which denotes the successive relations between vertices will all be zero, formula (11) can be further extended with matrix notations as follows to calculate the expectation values of all vertices at the kth step.

$$\mathbf{p}^{(k)} = \delta \mathbf{p}^{(0)} + (1-\delta)\left(\mathbf{S}^{(k-1)^T} \mathbf{F}^{(k-1)}\right) \mathbf{p}^{(k-1)} \tag{14}$$

3.2.2 Termination of tag-dependent random search

According to formula (14), as the random search process keeps on, we can derive the following formula to calculate the final value vector $\mathbf{p}_\pi$ when the search step goes to positive infinity, i.e. $\mathbf{p}_\pi = \lim_{n \to \infty} \mathbf{p}^{(n)}$.

$$\begin{aligned} \mathbf{p}_\pi = \lim_{n \to \infty} \delta \left(1 + \sum_{k=1}^{n-1} (1-\delta)^k \prod_{h=1}^{k} \left(\mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)}\right)\right) \mathbf{p}^{(0)} + \\ \left(\prod_{h=1}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)}\right)\right) \mathbf{p}^{(1)} \end{aligned} \tag{15}$$

It can be demonstrated that the proposed tag-dependent random search process is convergent and the second part of formula (15) will tend to be zero when $n \to \infty$. For details of demonstration, one can refer to the Appendix. And thus formula (15) can be simplified as follows.

$$\mathbf{p}_{\pi} \sim \lim_{n\to\infty}\left(1+\sum_{k=1}^{n-1}(1-\delta)^{k}\prod_{h=1}^{k}\left(\mathbf{S}^{(n-h)^{T}}\mathbf{F}^{(n-h)}\right)\right)\mathbf{p}^{(0)} \tag{16}$$

And in practice, with a given convergence threshold ϵ, the tag-dependent random search process can terminate at the nth step when

$$\|\mathbf{p}^{(n)}-\mathbf{p}^{(n-1)}\|_{2}<\epsilon \tag{17}$$

where $\|\cdot\|_2$ is the *L2 norm* of a vector. Then by normalizing $\mathbf{p}^{(n)}$ to ensure it to sum up to 1, we can get a feasible approximation of $\mathbf{p}_{\pi}$. And then the tag-dependent trust degree of the visual neighbour I_j *w.r.t* the candidate tag t_i, i.e. $c\left(I_j, I, t_i\right)$ in formula (5), will be estimated as the jth entry of $\mathbf{p}_{\pi}$.

3.2.3 Comparisons with random walk

As mentioned formerly, we propose that the well-known random walk could be inferior to the proposed tag-dependent random search process for exploiting the graphical model built by range-constrained visual neighbours to derive the tag-dependent trust degrees of visual neighbours. Firstly and most seriously, random walk is basically tag-independent, since the initial value vector, successive relations between vertices and the probability of moving forward from each vertex at any step in random walk are invariant for all candidate tags. Secondly, for a random walk process, the staying probabilities at all vertices at any step are always zero, but actually the more trustworthy neighbours of a candidate tag are expected to be associated with higher non-zero staying probabilities, making it more probable for the process to stay at such vertices and leading them to deriving larger expectation values that can reflect their higher tag-dependent trust degrees *w.r.t* the candidate tag. Finally, the number of successive vertices remains invariant in a random walk process as the step goes deeper, while it would be better to decrease, since a deeper step can probably lead to reaching more less-related neighbours, especially when the weakly-related range of visual neighbours is large.

3.3 Refinement for TagSearcher

By analysing the basic TagSearcher presented above, we realize that there exist some drawbacks *w.r.t* the rare tags whose frequencies in the given labelled image database are quite low, as detailed in the following paragraphs. Rare tags can be identified with a predefined frequency threshold, and any tag with its frequency below the threshold will be a rare one. Similar to the sophisticated TagProp [7] that develops tag-specific logistic regression models for the rare tags, here we propose some potential heuristic refinement approaches for them to improve the annotation performance of TagSearcher.

Firstly, the votes for rare tags from visual neighbours tend to be smaller, resulting in their lower predicted tag scores. It is because that a rare tag generally does not co-occur with other tags in sufficient images, which leads to a smaller numerator in formula (8) when estimating votes for it from neighbours. And thus rare tags occupy a disadvantaged position when predicting tag scores. To overcome that, we resort to the widely-used common knowledge base, WordNet [24], for completing the tagging matrix of the given labelled image database. Note that each row and each column of the tagging matrix respectively correspond to an image and a tag, and the entries of the tagging matrix always lie in {0, 1} indicating whether an image contains a tag (1) or not (0). Specifically, for each rare tag, all the tagging vectors of other tags will be summed with their corresponding semantic similarities to the rare tag as weights, which in our experiments are estimated with the semantic similarity measurement proposed by J. Jiang et al. [10]. Then the zero positions in the tagging vector of the rare tag will be changed to 1 if corresponding values in the summed vector are above some predefined threshold, which essentially improves the frequency of the rare tag by finding its potential associations with more images. After completing the tagging matrix with WordNet, the tag correlations and the votes for tags from visual neighbours are re-estimated.

Secondly, for a to-be-annotated image, the weight distribution among visual neighbours in the basic TagSearcher remains the same for both frequent and rare candidate tags. In most cases, however, a rare tag just appears as a bundled attachment in visual neighbours, which may be even unrelated with the to-be-annotated image. For instance, the nearest visual neighbour generally contains frequent tags that can describe most visual content of a to-be-annotated image, while it may also be attached with a few unrelated rare tags describing its own visual content. Therefore, we propose that the weight distribution among visual neighbours for rare tags should be more insensitive to the rank positions of neighbours. Then we adjust formula (6) as follows for the rare tags.

$$w\left(I, I_j\right) = \frac{1}{d\left(I, I_j\right)} \log\left(\frac{U+1}{\lceil \mu j \rceil}\right) \tag{18}$$

where μ is a parameter in (0, 1) controlling the impact of rank position on the image-dependent weight of a visual neighbour, and $\lceil \cdot \rceil$ is a ceiling function. It can be seen that here we utilize an echelon decline curve to estimate the weight distribution among visual neighbours for any rare tag.

Note that the proposed refinement approaches for rare tags are respectively about votes for a candidate tag from visual neighbours and image-dependent weights of visual neighbours. Both do help to enhance the performance of the proposed TagSearcher, as will be demonstrated by our experiments. Definitely, many other more sophisticated and effective refinement approaches can be applied, which will be further investigated in our future work.

4 Experiments

4.1 Experimental settings

In our experiments, we use the benchmark dataset Corel5k, which is widely used in previous researches on image auto-annotation, to evaluate the proposed TagSearcher and make comparisons with previous work. Moreover, as the vocabularies of many benchmark datasets like ESPGame and IAPRTC-12 [7] are relatively small, we build a new web image dataset named Flickr30Concepts with a larger vocabulary, and utilize it for evaluating TagSearcher in a real-world case of image auto-annotation. Some statistics of both datasets are presented in Table 1.

Corel5k is one of the most important evaluation benchmarks in the community of image auto-annotation, containing around 5,000 images that are manually annotated with 1 to 5 tags. And a fixed set of 499 images is split out for test, with the remaining ones working as the training set, i.e. the given labelled database. Note that here we utilize the standard

Table 1 Statistics of both benchmark Corel5k and real-world Flickr30Concepts. Counts of images and tags are given in the format "mean / maximum"

	Corel5k	Flickr30Concepts
Vocabulary size	260	1,128
Nr. of Images	4,999	29,998
Tags per image	3.4 / 5	7.0 / 70
Images per tag	65.5 / 1,067	184.9 / 2,891

split for Corel5k, which is given by P. Duygulu et al. [5] and utilized by many previous researches [2, 6, 7, 11, 14, 18, 20, 23, 29, 33, 35]. There are totally 260 tags existing in both training set and test set. And with accurate manual annotations, the dataset contains little noise and is widely-used for evaluating auto-annotation methods.

The new-built dataset, Flickr30Concepts, is collected from the popular photo sharing community Flickr[2] by submitting 30 non-abstract concepts[3] as queries. And for each query, the top 1,000 retrieved images are gathered. Following previous experiments on other datasets like ESPGame, IAPRTC-12 and Corel30k [7, 18, 23, 29, 35], we randomly select ten percent of the dataset to be a test set and others to be a training set. Then we utilize WordNet for stemming and filtering the raw tags, and finally get a vocabulary of 1,128 distinct words appearing in both training and test sets. Compared with Corel5k, Flickr30Concepts has a much larger vocabulary and contains much more noise in the given annotations of the training set, which can be seen as a real-world case of image auto-annotation.

In our experiments, for each image, eleven kinds of low-level features[4] are extracted with the open-source project Lire [22], including global and local features, color and texture features, etc. With kinds of image features, auto-annotation methods are enabled to deal with different levels of semantic, such as object-level semantic (e.g. "dog") and scene-level semantic (e.g. "winter"). To measure the visual similarity between two images, Lire is further utilized for calculating the distances between their corresponding feature vectors. Namely, we utilize *L1-norm distance* for Color Correlogram, RGB Color Histogram and Scalable Color, *Euclidean distance* for HSV Color Histogram, Jpeg Coefficient Histogram and SURF, *Tanimoto distance* for CEDD, FCTH and JCD, and other distance metrics defined in MPEG-7 standard [25] for Color Layout and Edge Histogram. Then all feature distances are normalized and merged with equal weights as JEC [23] to denote the visual similarities between images.

Following previous work [2, 6, 7, 11, 14, 18, 20, 23, 29, 33, 35], to evaluate an auto-annotation method, each test image is annotated with 5 tags, and then precision p and recall

[2] See: http://www.flickr.com/

[3] The 30 non-abstract concepts are: aircraft, ball, beach, bike, bird, book, bridge, car, chair, child, clock, countryside, dog, door, fire, fish, flower, house, kite, lamp, mountain, mushroom, pen, rabbit, river, sky, sun, tower, train, tree.

[4] The features include: Color Correlogram, Color Layout, CEDD, Edge Histogram, FCTH, HSV Color Histogram, JCD, Jpeg Coefficient Histogram, RGB Color Histogram, Scalable Color, SURF with bag-of-words model.

r for all tags in the vocabulary are calculated to measure its performance. Specifically, p and r are respectively defined as follows.

$$p = \frac{1}{|T|} \sum_{t_i \in T} \frac{N_c(t_i)}{N_a(t_i)} \tag{19}$$

$$r = \frac{1}{|T|} \sum_{t_i \in T} \frac{N_c(t_i)}{N_g(t_i)} \tag{20}$$

where $|T|$ is the size of the vocabulary T, and for each tag t_i, $N_c(t_i)$ is the number of correctly annotated images, $N_a(t_i)$ is the number of images annotated with t_i by the auto-annotation method, and $N_g(t_i)$ is the number of images containing t_i in ground-truth. Additionally, the number of tags with non-zero recall, denoted as N^+, is another important metric for evaluating auto-annotation performance. Note that for both datasets we use the associated tags of each test image as its ground truth, since manual judgement would be quite time-consuming and the associated tags are mostly correct.

In our experiments, we firstly evaluate the proposed TagSearcher and its refined variants on both the benchmark Corel5k and the real-world Flickr30Concepts, making comparisons with previous work. Then we give an inside analysis concerning the proposed refinement approaches for TagSearcher. And finally we conduct experiments to validate the reasonableness of the proposed tag-dependent random search process over range-constrained visual neighbours via analysing the effects of its parameters.

4.2 Annotation performance

Table 2 gives an overview of the annotation performance in terms of precision p, recall r and N^+ of the proposed TagSearcher and those reported in other remarkable earlier researches of image auto-annotation on the benchmark Corel5k. JEC* is our implementation of the widely-used baseline JEC [23], and TagProp* refers to the published implementation of TagProp [7] by the author M. Guillaumin, both using the eleven kinds of image features extracted by Lire here. Note that TagProp has several variants like σRK, σML, etc. And in our experiments we select the one with the best performance on a validate set split out from the training set for comparison, with corresponding parameters carefully tuned. Additionally, to compare random walk with the proposed tag-dependent random search, we introduce another baseline denoted as RW, which uses the same annotating framework as TagSearcher (i.e. formula (5)) but utilizes random walk instead of tag-dependent random search for exploiting the graphical model built with visual neighbours and estimating trust degrees of visual neighbours that are identical for all candidate tags. For the proposed TagSearcher, we empirically set the strong and the weak upper bounds for the quantity setting of visual neighbours as 10 and 60, the decreasing ratio $1/\lambda$ for the proposed tag-dependent random search process as 0.5 (i.e. $\lambda = 2$), and δ in formula (16) as zero for reducing computational complexity. Additionally, we denote the refined variant of TagSearcher with tagging matrix completion using WordNet as TS+WN, and that with weight distribution adjustment for rare tags as TS+WDA ($\mu = 0.5$). Furthermore, we merge both refinement approaches in TagSearcher, denoted as TS+Both. In our experiments, for the refined variants, we set the frequency threshold for identifying rare tags as the median value of all tag frequencies, hoping to compensate for more tags that are less labelled. And for TS+WN, the threshold for flipping entries of the tagging vector of a rare tag from 0 to 1 is set as 0.8, which is tuned on the validate set. The proposed TagSearcher is implemented using Matlab 8.1 and conducted on a PC with an Intel Core i5-2400 CPU and 4G RAM. Using a single thread, it takes about

Table 2 Annotation performance in terms of precision (p), recall (r) and N^+ of the proposed TagSearcher, and those reported in a selection of remarkable earlier researches. JEC* and TagProp* respectively refer to the corresponding methods using our features, while JEC [23] and TagProp [7] are respectively the best reported results in corresponding published papers. RW refers to the same annotating framework using random walk instead of the proposed tag-dependent random search process. And here we present results for both the basic TagSearcher (i.e. TS) and its refined variants (i.e. TS+WN, TS+WDA, TS+Both)

		p	r	N^+
	DCMRM[20]	0.23	0.28	135
	SML[2]	0.23	0.29	137
	MBRM[6]	0.24	0.25	122
	TGLM[18]	0.25	0.29	131
	MSC[29]	0.25	0.32	136
	JEC[23]	0.27	0.32	137
	MPMF[14]	0.27	0.34	135
	HDGM[11]	0.29	0.30	146
	GS[35]	0.30	0.33	146
	En-CRF[33]	0.32	0.33	148
	TagProp[7]	**0.33**	**0.42**	**160**
	JEC*	0.29	0.33	139
	TagProp*	0.30	0.32	141
	RW	0.29	0.34	141
TagSearcher	TS	0.30	0.34	142
	TS+WN	0.31	0.33	142
	TS+WDA	0.31	**0.36**	146
	TS+Both	**0.32**	0.35	**149**

0.3 seconds for TagSearcher to annotate a test image of Corel5k on average, which can be further reduced with parallel computing.

From Table 2 we can see that the annotation performance of JEC* on Corel5k is similar to those reported in previous researches with their own implementations and features, making it relatively fair to compare with their reported results. Then we can get the following observations. (1) The proposed TagSearcher and its refined variants outperform most previous remarkable auto-annotation methods, and achieve comparable annotation performance to the state-of-the-art TagProp [7]. The superiority of TagProp can be due to its sophisticated metric learning methods for better estimating the image-dependent weights of visual neighbours and its tag-specific logistic regression models for boosting annotation performance of rare tags, while in the proposed TagSearcher we just use a naïve weight estimation method for visual neighbours and other heuristic refinement approaches for rare tags since both are not the main focus of our work. Moreover, the reported experiments in TagProp utilized more kinds of low-level image features (15 kinds), which are supposed to yield better performance. (2) When comparisons are strictly made with the same kinds of features, the proposed TagSearcher and its refined variants outperform JEC* and the selected best variant of TagProp. Both observations above well demonstrate the effectiveness of the proposed TagSearcher for image auto-annotation. (3) With the same annotating framework, the proposed TagSearcher yields slightly better performance than the baseline RW, which gives first evidence that the proposed tag-dependent random search process is superior to the widely-used random walk for estimating tag-dependent trust degrees of visual neighbours and it is reasonable for each candidate tag to have its own selection of trustworthy visual neighbours for score prediction. (4) The refined variants of TagSearcher, i.e. TS+WN, TS+WDA and TS+Both, generally yield superior annotation performance to that of the basic TagSearcher, especially TS+Both, which demonstrates the effectiveness of the proposed heuristic refinement approaches for rare tags, i.e. tagging matrix completion with WordNet (i.e. WN) and weight distribution adjustment among visual neighbours (i.e. WDA).

To evaluate the proposed TagSearcher in a real-world case of image auto-annotation, we further conduct experiments on the new-built Flickr30Concepts. Table 3 gives an overview of the experimental results of the proposed TagSearcher and representative baselines, i.e. the widely-used baseline JEC [23], the state-of-the-art TagProp [7] and the introduced baseline RW. Note that here TagProp* is the best variant of TagProp on Flickr30Concepts via re-selecting its variants on a validate set, while TagProp*$_{corel5k}$ is the same variant as the former experiments on Corel5k. Here for the proposed TagSearcher, due to the larger vocabulary, it takes around 0.9 seconds to annotate a test image of Flickr30Concepts on average, using a single thread and conducted on the same PC as Corel5k.

From Table 3 we can observe that TagSearcher and its refined variants yield superior performance to that of JEC and TagProp*$_{corel5k}$ in terms of precision p ($\geqslant$38 %), recall r ($\geqslant$89 %) and N^+ ($\geqslant$45 %), but they are still a little inferior to TagProp*. However, it can be seen that TagProp* and TagProp*$_{corel5k}$, which both perform sophisticated metric learning for estimating image-dependent weights of visual neighbours and train tag-specific logistic regression models for rare tags on Flickr30Concepts, yield quite different annotation performance. And thus for TagProp, it would be necessary to re-select the best variants for different datasets in practical applications, which can be complex and time-consuming for large-scale datasets. On the contrary, the proposed TagSearcher maintains robust and superior annotation performance to many previous researches on both benchmark and real-world datasets, with its parameters effortlessly set with empirical values. Moreover, since TagProp focuses on estimating the image-dependent weights of visual neighbours with sophisticated metric learning methods while the proposed TagSearcher focuses on estimating the tag-dependent trust degrees of visual neighbours, it would be interesting to integrate both in a unified annotating framework, e.g. formula (5). In Table 3, the comparison between the performance of RW and that of TagSearcher further gives strong evidence for that the proposed tag-dependent random search is superior to random walk for estimating tag-dependent trust

Table 3 Annotation performance of the proposed TagSearcher and representative baselines on the real-world Flickr30Concepts, in terms of precision p, recall r and N^+. TagProp* is the selected best variant of TagProp on Flickr30Concepts, while TagProp*$_{corel5k}$ is the same variant as the former experiments on Corel5k

		p	r	N^+
	JEC*	0.25	0.16	535
	TagProp*	**0.47**	**0.38**	919
	TagProp*$_{corel5k}$	0.29	0.19	637
	RW	0.34	0.20	699
TagSearcher	TS	0.39	0.34	894
	TS+WN	0.40	0.34	896
	TS+WDA	0.39	0.36	**921**
	TS+Both	0.40	0.36	**921**

degrees of visual neighbours and it is reasonable for each candidate tag to have its own selection of trustworthy visual neighbours for score prediction. Additionally, the performance enhancements achieved by refinement approaches for rare tags further demonstrate the reasonableness of our analyses concerning the drawbacks of the basic TagSearcher and the effectiveness of the proposed heuristic refinement approaches. It is also interesting to find that the superiority of the proposed tag-dependent random search to random walk is much more evident on Flickr30Concepts than that on Corel5k, which can be attributed to the higher tag frequencies on Flickr30Concepts that help to make the estimation of tag correlations more reliable for the proposed tag-dependent random search process.

Figure 3 gives samples of the annotation results of the proposed TagSearcher on both the benchmark Corel5k (the upper row) and the real-world Flickr30Concepts (the lower row), with the red tags being the ground-truth and the black ones being the annotation results of TagSearcher. From the samples we can see that although sometimes a few of the annotations given by TagSearcher are not in the ground-truth, e.g. "clouds" in the top left image, they are generally related to the image content, which further demonstrates the effectiveness of TagSearcher.

4.3 Inside analyses on refinement for TagSearcher

In Fig. 4, we further give inside details of the effects brought by the proposed heuristic refinement approaches for rare tags on basic TagSearcher. Specifically, for either Corel5k

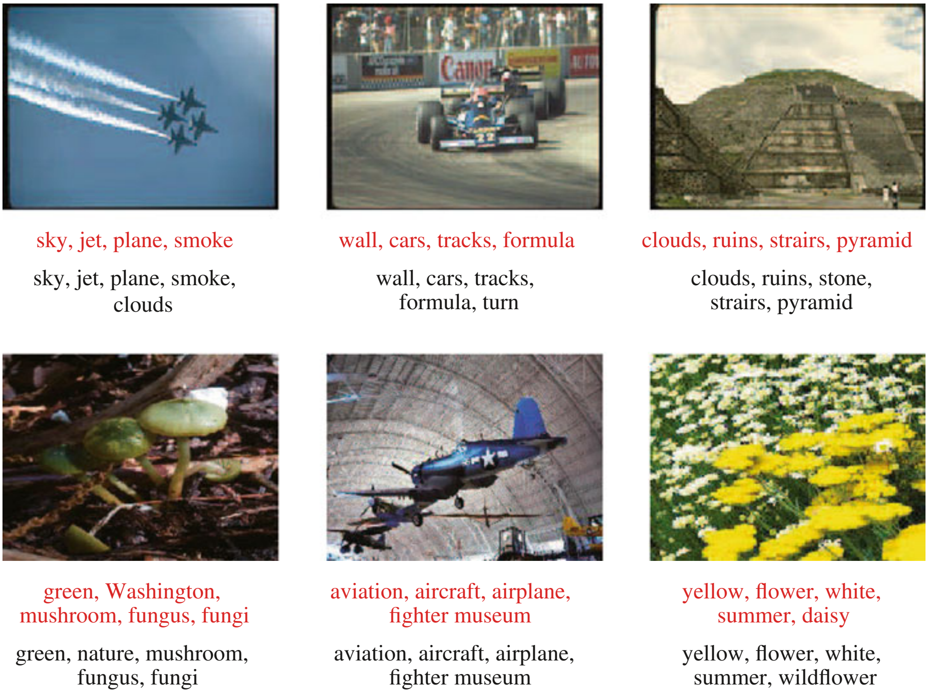
sky, jet, plane, smoke
sky, jet, plane, smoke, clouds
wall, cars, tracks, formula
wall, cars, tracks, formula, turn
clouds, ruins, strairs, pyramid
clouds, ruins, stone, strairs, pyramid
green, Washington, mushroom, fungus, fungi
green, nature, mushroom, fungus, fungi
aviation, aircraft, airplane, fighter museum
aviation, aircraft, airplane, fighter museum
yellow, flower, white, summer, daisy
yellow, flower, white, summer, wildflower

Fig. 3 Samples of the annotation results of the proposed TagSearcher on both the benchmark Corel5k (*the upper row*) and the real-world Flickr30Concepts (*the lower row*), with the red tags being the ground-truth and the black ones being the annotation results of TagSearcher

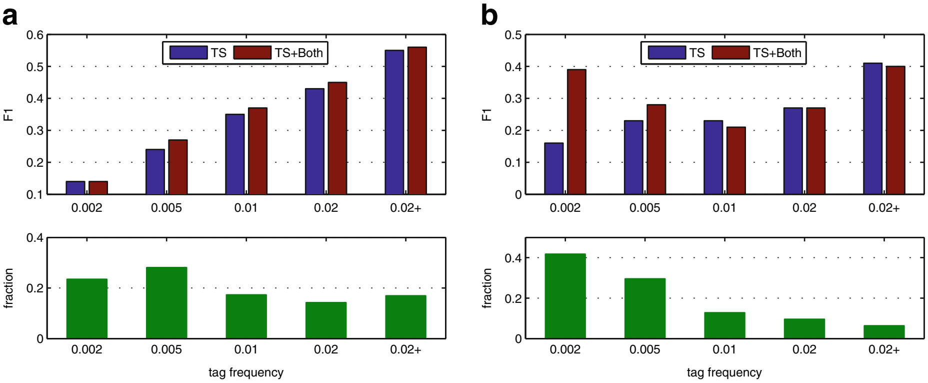

Fig. 4 Mean *F1-score* of tags in Corel5k (subfigure (**a**)) and Flickr30Concepts (subfigure (**b**)) for the basic TagSearcher (i.e. TS, *blue*) and its refined variant (i.e. TS+Both, *red*), grouped by their relative frequencies in the training set, e.g. the first bin groups tags with relative frequency in (0, 0.002]. The lower bars illustrate the fraction of tags in each bin on Corel5k and Flickr30Concepts, and the upper bars illustrate the mean *F1-score* of tags in each bin

or Flickr30Concepts, all tags in the vocabulary are binned according to their corresponding relative frequencies in the training set, and then we calculate the mean *F1-score* of each bin of tags for both the basic TagSearcher and its refined variant (i.e. TS+Both). The relative frequency of a tag t is defined as $\frac{|t|}{N}$, where $|t|$ is the number of images containing t and N is the number of all given labelled images. The *F1-score* of each tag is defined as $F1 = 2\frac{precision \cdot recall}{precision+recall}$, which is an integral performance metric considering both precision and recall. Since the mean relative frequency of tag in either the benchmark Corel5k or the real-world Flickr30Concepts is around 0.01, we utilize 0.002, 0.005, 0.01 and 0.02 as boundaries for grouping the tags. From Fig. 4, it can be seen that the proposed heuristic refinement approaches do benefit the rare tags and bring inspiring performance improvement, even in different datasets with distinct tag frequency distributions. Moreover, though the refinement approaches for rare tags may sometimes worsen the annotation performance of frequent tags, as illustrated in Fig. 4(b), the corresponding performance degradation is generally quite slight, thus leading to an overall performance enhancement.

4.4 Effects of parameters

To further validate the reasonableness of the proposed TagSearcher, especially that of the proposed tag-dependent random search process over range-constrained visual neighbours, we conduct experiments to see the effects of its key parameters on the annotation performance, including the decreasing ratio $1/\lambda$ for tag-dependent random search, the strong upper bound and the weak upper bound for the quantity setting of visual neighbours. To avoid the risky biases brought by noise in a dataset, here experiments are conducted on the benchmark Corel5k.

As mentioned formerly, in a tag-dependent random search process the number of candidate successive vertices is supposed to decrease with a ratio $1/\lambda$ as the step goes deeper, in order to avoid reaching too many less-related neighbours. Then with all other parameters fixed and identical to former experiments on Corel5k, we vary λ from 0.5 to 4 to see its effects. Note that $0 < \lambda < 1$ (i.e. $1/\lambda > 1$) means that the number of candidate successive vertices will increase as the step goes deeper, while $\lambda > 1$ (i.e. $0 < 1/\lambda < 1$) means that the number will decrease. And $\lambda = 1$ (i.e. $1/\lambda = 1$) means that the number will keep invariant, which is similar to random walk. Figure 5 illustrates the effects of λ on annotation performance of TagSearcher in terms of mean *F1-score*. It can be seen that when λ changes from $0 < \lambda < 1$ to $\lambda = 1$ or from $\lambda = 1$ to $\lambda > 1$, the annotation performance achieves a relatively significant enhancement. Additionally, the annotation performance of $\lambda > 1$ consistently outperforms that of $0 < \lambda < 1$ and $\lambda = 1$, which well demonstrates our proposal concerning the number of candidate successive vertices. Nevertheless, a large λ may also degrade the annotation performance, since it quickly constrains the random search process to insufficient successive vertices. As shown in Fig. 5, the optimal λ is around 2, which is identical to the one we used in former experiments.

Then to analyse the effects of both the strong and the weak upper bounds for the quantity setting of visual neighbours, we keep one bound invariant and investigate the performance of TagSearcher as the other varies, with other parameters fixed and identical to former experiments on Corel5k. Specifically, with the strong upper bound fixed as 10, we vary the weak one from 10 to 200, denoted as TagSearcher_WeakUpperBound. Similarly, with the weak upper bound fixed as 60, we vary the strong one from 1 to 60, denoted as TagSearcher_StrongUpperBound. Additionally, to present the effects of introducing the tag-dependent trust degrees of visual neighbours, we use a simple but typical VNB baseline termed VNvote, which predicts tag scores using formula (5) without $c\left(I_j, I, t_i\right)$, similar to most previous VNB methods. And thus the only difference between VNvote and the proposed TagSearcher is that the latter uses a constrained range of visual neighbours and performs tag-dependent random search to estimate tag-dependent trust degrees of visual neighbours for each candidate tag.

Figure 6 illustrates the performance of TagSearcher with the strong or the weak upper bound varying and that of VNvote with the quantity setting of visual neighbours changing, in terms of precision p, recall r and N^+. From that, we can draw the following conclusions. (1) Compared with VNvote, the annotation performance of the proposed TagSearcher is much less sensitive to the bound settings of visual neighbours, which is attributed to the range constraint for visual neighbours and the proposed tag-dependent random search process for estimating the tag-dependent trust degrees of visual neighbours. (2) The annotation performance of TagSearcher remains comparable to or even better than the best

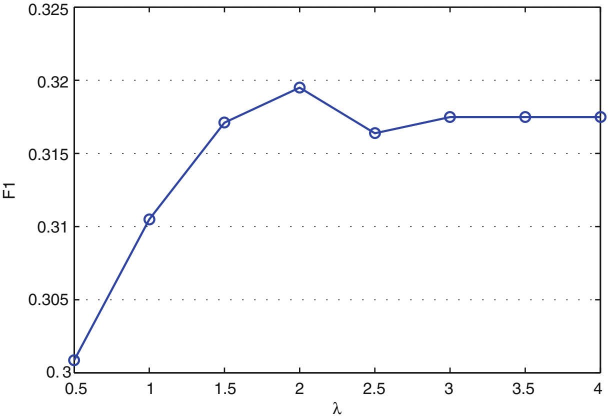

Fig. 5 Effects of λ (i.e. the abscissa) on annotation performance of the proposed TagSearcher on the benchmark Corel5k, in terms of mean *F1-score*

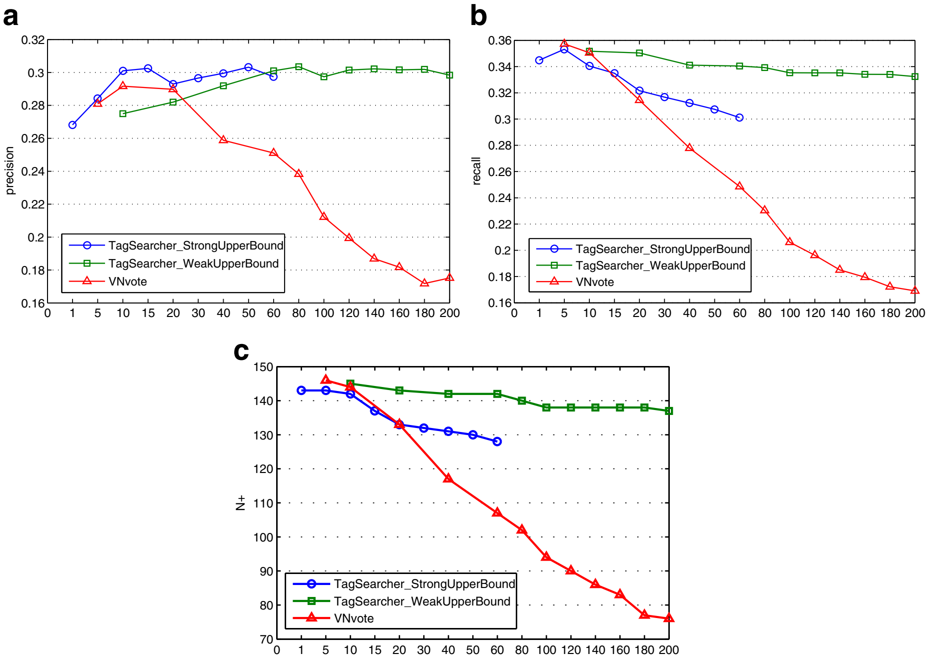

Fig. 6 The annotation performance of TagSearcher on Corel5k with the strong or the weak upper bounds varying, compared with VNvote as a baseline, in terms of precision p (subfigure (**a**)), recall r (subfigure (**b**)) and N^+ (subfigure (**c**)). The abscissa is the quantity setting of visual neighbours for VNvote, and also the value of the strong or the weak upper bound for the constrained range of visual neighbours in TagSearcher

performance of VNvote, even though the value of either the strong upper bound or the weak one varies in quite a large range, which well demonstrates the robustness and effectiveness of the proposed TagSearcher and tag-dependent random search process. (3) The strong upper bound for the constrained range of visual neighbours has a more significant effect on annotation performance than the weak one. It is because that in a tag-dependent random search process, we rely much more on the strongly-related range of visual neighbours than the weakly-related one.

5 Conclusions

In this paper, with the observation that the quantity setting of visual neighbours can be critical for the performance of many previously proposed visual-neighbour-based image auto-annotation methods and each candidate tag for a to-be-annotated image is better to have its own trustworthy part of visual neighbours for score prediction, we propose to use a constrained range rather than an identical and fixed number of visual neighbours and further put forward a novel tag-dependent random search process to estimate their tag-dependent trust degrees *w.r.t* each candidate tag. Furthermore, based on a conditional probability model widely-used for image auto-annotation, we propose an effective image auto-annotation method termed TagSearcher, considering image-dependent weights of visual neighbours, tag-dependent trust degrees of visual neighbours and votes for a candidate tag from visual neighbours. The proposed TagSearcher is evaluated with extensive experiments on both a benchmark dataset and real-world web images, with experimental results well demonstrating its reasonableness and revealing that it can not only yield inspiring auto-annotation performance but also help to reduce the performance sensitivity.

In the future, to further enhance the proposed TagSearcher, we will utilize sophisticated metric learning methods as the state-of-the-art TagProp for better estimating image-dependent weights of visual neighbours, and investigate more other effective refinement approaches for rare tags.

Acknowledgments This research was supported by the National Basic Research Project of China (Grant No. 2011CB707000) and the National Natural Science Foundation of China(Grant No. 61271394, 61005045).

Appendix

A1. Convergence proof

Note that in formula (15), each column of the matrix $\mathbf{S}^{(n-h)^T}$ denoting the successive relations between vertices is *L1* normalized to sum up to 1, and the entries of any $\mathbf{S}^{(n-h)^T}$ or $\mathbf{F}^{(n-h)}$ are all between 0 and 1. For any δ lying in (0, 1), there always exists $\gamma < 1$ subject to $1 - \delta < \gamma$, and thus we can derive that:

$$
\begin{aligned}
&\sum_j \left(\prod_{h=1}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right) \right)_{ji} \\
&= \sum_j \sum_k \left((1-\delta)\, \mathbf{S}^{(n-1)^T} \mathbf{F}^{(n-1)} \right)_{jk} \left(\prod_{h=2}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right) \right)_{ki} \\
&= \sum_k \left(\prod_{h=2}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right) \right)_{ki} (1-\delta) \sum_j \left(\mathbf{S}^{(n-1)^T} \mathbf{F}^{(n-1)} \right)_{jk} \\
&\leqslant (1-\delta) \sum_k \left(\prod_{h=2}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right) \right)_{ki} \sum_j \left(\mathbf{S}^{(n-1)^T} \right)_{jk} \\
&= (1-\delta) \sum_k \left(\prod_{h=2}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right) \right)_{ki} \\
&\leqslant \gamma \sum_k \left(\prod_{h=2}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right) \right)_{ki} \\
&\leqslant \gamma \left(\gamma \sum_k \left(\prod_{h=3}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right) \right)_{ki} \right) \\
&\leqslant \dots \\
&\leqslant \gamma^{n-1}
\end{aligned}
$$

Therefore, the sum of each column of $\prod_{h=1}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right)$ will tend to be zero as n goes to positive infinity. Since $(1 - \delta)$, $\mathbf{S}^{(n-h)^T}$ and $\mathbf{F}^{(n-h)}$ are all non-negative, $\prod_{h=1}^{n-1} \left((1-\delta)\, \mathbf{S}^{(n-h)^T} \mathbf{F}^{(n-h)} \right)$ will also be non-negative and thus

$\left(\prod_{h=1}^{n-1}\left((1-\delta)\,\mathbf{S}^{(n-h)^T}\mathbf{F}^{(n-h)}\right)\right)\mathbf{p}^{(1)}$ will converge to a zero vector. Then formula (15) can be simplified as follows.

$$\mathbf{p}_\pi = \lim_{n\to\infty}\delta\left(1+\sum_{k=1}^{n-1}(1-\delta)^k\prod_{h=1}^{k}\left(\mathbf{S}^{(n-h)^T}\mathbf{F}^{(n-h)}\right)\right)\mathbf{p}^{(0)}$$

Since we only focus on the ratios between entries of $\mathbf{p}_\pi$ rather than their values, the formula above can be further simplified as follows.

$$\mathbf{p}_\pi \sim \lim_{n\to\infty}\left(1+\sum_{k=1}^{n-1}(1-\delta)^k\prod_{h=1}^{k}\left(\mathbf{S}^{(n-h)^T}\mathbf{F}^{(n-h)}\right)\right)\mathbf{p}^{(0)}$$

And it can be seen that entries of $\mathbf{p}_\pi$ keep increasing as n increases. Since entries of any $\mathbf{S}^{(n-h)^T}$, $\mathbf{F}^{(n-h)}$ and $\mathbf{p}^{(0)}$ are all non-negative and lie in [0, 1], we can further derive that:

$$\begin{aligned}&\lim_{n\to\infty}\left(1+\sum_{k=1}^{n-1}(1-\delta)^k\prod_{h=1}^{k}\left(\mathbf{S}^{(n-h)^T}\mathbf{F}^{(n-h)}\right)\right)\mathbf{p}^{(0)}\\ &=\lim_{n\to\infty}\mathbf{p}^{(0)}+\sum_{k=1}^{n-1}(1-\delta)^k\prod_{h=1}^{k}\left(\mathbf{S}^{(n-h)^T}\mathbf{F}^{(n-h)}\right)\mathbf{p}^{(0)}\\ &\preceq\lim_{n\to\infty}\left(\mathbf{p}^{(0)}+\sum_{k=1}^{n-1}(1-\delta)^k\,\mathbb{I}\right)\end{aligned}$$

where $\mathbb{I}$ is a column vector with entries all being 1, and "$\preceq$" means entry-wise "$\leqslant$". Since $(1-\delta)$ lies in (0, 1), $\sum_{k=1}^{n-1}(1-\delta)^k$ will converge as n goes to positive infinity. Then $\mathbf{p}_\pi$ is a monotonically increasing function *w.r.t* the step n and has an upper bound. Therefore, it will converge as n goes to positive infinity, meaning that convergence of the proposed tag-dependent random search process is guaranteed.

References

1. A focus on efficiency - a whitepaper from facebook, ericsson and qualcomm (2013)
2. Carneiro G, Chan AB, Moreno PJ, Vasconcelos N (2007) Supervised learning of semantic classes for image annotation and retrieval. IEEE Trans Pattern Anal Mach Intell 29(3):394–410
3. Chang E, Goh K, Sychay G, Wu G (2003) Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines. IEEE Trans Circ Syst Video Technol 13(1):26–38
4. Chen X, Hu X, Zhou Z, Lu C, Rosen G, He T, Park EK (2010) A probabilistic topic-connection model for automatic image annotation. In: Proceedings of the 19th ACM international conference on information and knowledge management
5. Duygulu P, Barnard K, de Freitas JFG, Forsyth DA (2002) Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In: Proceedings of the 7th European conference on computer vision
6. Feng SL, Manmatha R, Lavrenko V (2004) Multiple bernoulli relevance models for image and video annotation. In: Proceedings of the IEEE conference on computer vision and pattern recognition
7. Guillaumin M, Mensink T, Verbeek J, Schmid C (2009) Tagprop: discriminative metric learning in nearest neighbor models for image auto-annotation. In: Proceedings of the IEEE international conference on computer vision

8. Hauptmann AG (2005) Lessons for the future from a decade of informedia video analysis research. In: Proceedings of the 4th international conference on image and video retrieval
9. Jeon J, Lavrenko V, Manmatha R (2003) Automatic image annotation and retrieval using cross-media relevance models. In: Proceedings of the 26th international ACM SIGIR conference on research and development in informaion retrieval
10. Jiang JJ, Conrath DW (1997) Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of the international conference on Research in Computational Linguistics
11. Ke X, Li S, Cao D (2012) A two-level model for automatic image annotation. Multimedia Tools Appl 61(1):195–212
12. Lavrenko V, Manmatha R, Jeon J (2003) A model for learning the semantics of pictures. In: advances in neural information processing systems
13. Li X, Chen L, Zhang L, Lin F, Ma W (2006) Image annotation by large-scale content-based image retrieval. In: Proceedings of the 14th annual ACM international conference on multimedia
14. Li Z, Liu J, Zhu X, Liu T, Lu H (2010) Image annotation using multi-correlation probabilistic matrix factorization. In: Proceedings of the international conference on multimedia
15. Li X, Snoek CGM, Worring M (2009) Annotating images by harnessing worldwide user-tagged photos. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing
16. Li J, Wang JZ (2003) Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Trans Pattern Anal Mach Intell 25(9):1075–1088
17. Liu D, Hua X, Yang L, Wang M, Zhang H (2009) Tag ranking. In: Proceedings of the 18th international conference on world wide web
18. Liu J, Li M, Liu Q, Lu H, Ma S (2009) Image annotation via graph learning. Pattern Recognit 42(2):218–228
19. Liu J, Li M, Ma W, Liu Q, Lu H (2006) An adaptive graph model for automatic image annotation. In: Proceedings of the 8th ACM international workshop on multimedia information retrieval
20. 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
21. Lowe DG (1999) Object recognition from local scale-invariant features. In: The proceedings of the 7th IEEE international conference on computer vision
22. Lux M, Chatzichristofis SA (2008) Lire: lucene image retrieval: an extensible java cbir library. In: Proceedings of the 16th ACM international conference on multimedia
23. Makadia A, Pavlovic V, Kumar S (2008) A new baseline for image annotation. In: Proceedings of the 10th european conference on computer vision
24. Miller GA (1995) Wordnet: a lexical database for english. Commun ACM 38(11):39–41
25. Salembier P, Sikora T (2002) Introduction to MPEG-7: multimedia content description interface. Wiley
26. Sigurbjörnsson B, Zwol RV (2008) Flickr tag recommendation based on collective knowledge. In: Proceedings of the 17th international conference on World Wide Web
27. Wang H, Huang H, Ding C (2009) Image annotation using multi-label correlated green's function. In: IEEE 12th international conference on computer vision
28. Wang C, Jing F, Zhang L, Zhang H (2006) Image annotation refinement using random walk with restarts. In: Proceedings of the 14th annual ACM international conference on multimedia
29. Wang C, Yan S, Zhang L, Zhang H (2009) Multi-label sparse coding for automatic image annotation. In: Proceedings of the IEEE conference on computer vision and pattern recognition
30. Wang F, Zhang C (2006) Label propagation through linear neighborhoods. In: Proceedings of the 23rd international conference on machine learning
31. Wang F, Zhang C (2008) Label propagation through linear neighborhoods. IEEE Trans Knowl Data Eng 20:55–67
32. Wang X, Zhang L, Li X, Ma W (2008) Annotating images by mining image search results. IEEE Trans Pattern Anal Mach Intell 30(11)
33. Xu X, Jiang Y, Peng L, Xue X, Zhou Z (2011) Ensemble approach based on conditional random field for multi-label image and video annotation. In: Proceedings of the 19th ACM international conference on multimedia
34. Yang Y, Wu F, Nie F, Shen HT, Zhuang Y, Hauptmann AG (2012) Web and personal image annotation by mining label correlation with relaxed visual graph embedding. IEEE Trans Image Process
35. Zhang S, Huang J, Huang Y, Yu Y, Li H, Metaxas DN (2010) Automatic image annotation using group sparsity. In: Proceedings of the IEEE conference on computer vision and pattern recognition

Zijia Lin received his B.Sc. degree from School of Software, Tsinghua University, Beijing, China in 2011, and currently is a Ph.D. candidate in Department of Computer Science and Technology in the same campus. His research interests include multimedia information retrieval and machine learning.

Guiguang Ding received his Ph.D degree in electronic engineering from the University of Xidian. He is currently an associate professor of School of Software, Tsinghua University. Before joining School of Software in 2006, he worked as a postdoctoral researcher in Automation Department of Tsinghua University. His current research centers on the area of multimedia information retrieval and mining, in particular, visual object classification, automatic semantic annotation, content-based multimedia indexing, and personal recommendation. He has published about 40 research papers in international conferences and journals and applied for 18 Patent Rights in China.

Mingqing Hu received the B.Sc. and M.Sc. degrees from Xidian University, Xian, China, in 1999 and 2002, respectively, and the Ph.D. degree from the University of Trento, Trento, Italy, in February 2007. For a year and a half before his doctoral studies, he worked in the industry. Currently, he is an Assistant Researcher at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. His research interests include machine learning (in particular kernel methods) and computer vision.