Ranking-Oriented Nearest-Neighbor Based Method for Automatic Image Annotation

Chaoran Cui[1]
bruincui@gmail.com

Jun Ma[1]
majun@sdu.edu.cn

Tao Lian[1]
liantao1988@gmail.com

Xiaofang Wang[1,2]
ise_wangxf@ujn.edu.cn

Zhaochun Ren[3]
z.ren@uva.nl

[1] School of Computer Science and Technology, Shandong University, Jinan, China
[2] School of Information Science and Engineering, University of Jinan, Jinan, China
[3] Intelligent Systems Lab Amsterdam, University of Amsterdam, Amsterdam, The Netherlands

ABSTRACT

Automatic image annotation plays a critical role in keyword-based image retrieval systems. Recently, the nearest-neighbor based scheme has been proposed and achieved good performance for image annotation. Given a new image, the scheme is to first find its most similar neighbors from labeled images, and then propagate the keywords associated with the neighbors to it. Many studies focused on designing a suitable distance metric between images so that all labeled images can be ranked by their distance to the given image. However, higher accuracy in distance prediction does not necessarily lead to better ordering of labeled images. In this paper, we propose a ranking-oriented neighbor search mechanism to rank labeled images directly without going through the intermediate step of distance prediction. In particular, a new learning to rank algorithm is developed, which exploits the implicit preference information of labeled images and underlines the accuracy of the top-ranked results. Experiments on two benchmark datasets demonstrate the effectiveness of our approach for image annotation.

Categories and Subject Descriptors

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

Keywords

image annotation; nearest-neighbor based scheme; learning to rank

1. INTRODUCTION

In recent decades, the number of digital images has been growing rapidly and there is an increasingly urgent demand for effective image retrieval techniques. Users often prefer searching images with a textual query, which can be achieved by first annotating images manually, and then searching over the annotations using the query. However, manual image annotation is a laborious and time-consuming process. Therefore, many efforts have been devoted to the research on automatic image annotation.

The goal of automatic image annotation is to assign a few relevant keywords to an image that can reflect its visual content. Recently, the nearest-neighbor based scheme [4] has become increasingly attractive because of its superior performance and straightforward framework. It is on the assumption that visually similar images are more likely to share common keywords. Given a new image, the nearest-neighbor based scheme first finds a set of its most similar neighbors from labeled images, and then propagates the keywords associated with the neighbors to it.

In spite of the simplicity of the nearest-neighbor based annotation framework, there are some critical issues that remain to be addressed. One important aspect of the framework is how to perform the process of the nearest neighbor search effectively. Many studies [4, 5, 6] focused on designing a suitable visual distance metric between images, which is then used to rank all labeled images according to their distance to the new image. Typically, the work aimed to weight and combine the distances from different dimensions in visual feature space. Despite the encouraging results achieved, we argue that higher accuracy in distance prediction does not necessarily lead to better ordering of labeled images, which is the ultimate goal of our problem. For example, let u and v denote two labeled images whose true distances to the new image are 4 and 5 respectively. Suppose a method has predicted the distance to be 5 for u and 4 for v. Although it is a desirable result in terms of prediction error, it fails in ensuring the correct ordering of u and v. In light of this, it is necessary to shift our attention from approximating the absolute distance between images to directly predicting the relative ordering of labeled images.

In this paper, we propose a ranking-oriented neighbor search mechanism, which uses the learning to rank [3] techniques to directly produce the ordering of all labeled images for a new image, without going through the intermediate step of distance prediction. Unlike regular learning to rank methods, our proposed ranking algorithm exploits the implicit preference information hidden in the training images. In addition, since only the k nearest neighbors are generally

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

considered in the nearest-neighbor based scheme, we enforce the ranking model to focus more on the correctness of the results in top-k positions. A boosting algorithm [1] is utilized to solve the resulting optimization problem in our approach.

2. RANKING-ORIENTED NEIGHBOR SEARCH MECHANISM

In this section, we introduce our ranking-oriented neighbor search mechanism in details. For the ease of explanation, we first give some notations.

Let $\mathcal{X}$ be an image collection, and all keywords appearing in the collection are $\mathcal{W} = \{w_1, w_2, \ldots, w_c\}$, where c is the total number of unique keywords. In image annotation task, we are given n labeled training images, $\mathcal{T} = \{x^{(i)} \in \mathcal{X} \mid i = 1, \ldots n\}$, each of which is associated with a c-dimensional label vector $y^{(i)} \in \{0,1\}^c$, where $y_j^{(i)} = 1$ if $x^{(i)}$ is labeled by the jth keyword w_j and $y_j^{(i)} = 0$ otherwise. Given a query image $q \in \mathcal{X}$, our goal is to find a ranking function $H : \mathcal{X} \times \mathcal{T} \to R$ such that $H(q, x^{(i)})$ represents the relevance of the labeled image $x^{(i)}$ with respect to q, and $x^{(i)}$ is ranked before $x^{(j)}$ if $H(q, x^{(i)}) > H(q, x^{(j)})$.

To resolve the above challenge, we seek to exploit the learning to rank (hereinafter referred to as LTR for short) techniques to learn the optimal ranking function H from the training data. Although LTR has been extensively studied [3], it is not straightforward to directly apply regular LTR techniques to our problem for the following two reasons. First, unlike standard LTR tasks where some preference information (in the forms of pairwise or listwise constraints) is often explicitly given to supervise the learning process, in our problem, preference information is only implicitly available in the training set. Moreover, generally we only consider the k nearest neighbors for a new image to prohibit the potential noisy keywords introduced by those distant neighbors. Therefore, in our ranking problem, the correct ordering of the top-k results is crucial, and the mistakes in low ranks may not deteriorate the final performance. It is necessary to redesign the training procedure to ensure the top-k results are as accurate as possible.

Based on the above analysis, to facilitate our ranking task, in the following, we first generate the implicit preference information hidden in the training data. With the preference information, we further present a new LTR algorithm that underlines the accuracy of the top-k results.

2.1 Generation of Preference Information

As no explicit preference information is given for our problem, the first step before LTR is to derive some preference information from the training data. Specifically, we separately submit each labeled image as a query and look for the information that could indicate the relative ordering among the other labeled images with respect to it.

It is notable that the intuition behind the nearest-neighbor based methods is that similar images should share more common keywords. This means that given an image, its close neighbors may have a higher keyword agreement with it compared to those distant neighbors. In accordance with this principle, we consider measuring the relative distance between labeled images by the consistency of their keywords. Given two label vectors y and $\hat{y}$, their consistency $CON(y, \hat{y})$ is estimated in a manner similar to F_1 measure:

$$CON(y, \hat{y}) = \frac{2pr}{p+r} \quad p = \frac{y^T \hat{y}}{\hat{y}^T \hat{y}} \quad r = \frac{y^T \hat{y}}{y^T y} \ . \tag{1}$$

With this definition, for a labeled image $x^{(i)}$, we define $\mathcal{R}_i = \{r_i^1 \ldots r_i^{i-1}, r_i^{i+1} \ldots r_i^n\}$ to be the relevance degrees of the other corresponding labeled images, i.e., $\mathcal{T} \backslash x^{(i)} = \{x^{(1)} \ldots x^{(i-1)}, x^{(i+1)} \ldots x^{(n)}\}$, where $r_i^j = CON(y^{(i)}, y^{(j)})$. Then, let π_i denote the total ranking of $\mathcal{T} \backslash x^{(i)}$ with respect to $x^{(i)}$, which can be derived in descending order of $\mathcal{R}_i$, and $\pi_i(x_j)$ stands for the position of image $x_j \in \mathcal{T} \backslash x^{(i)}$.

Although π_i seems a natural way of representing the preference information associated with $x^{(i)}$, the variations in the importance of partial orderings with different ranking positions cannot be easily reflected in such a form of linear ordering. To allow encoding this kind of information, in this paper, we construct the preference information in the form of ordered pairs of images, and also assign each pair a weight to represent the importance of its being satisfied. In particular, a set of p ordered pairs $\mathcal{W}_i = \{w(x_{j_m} \succ_i x_{q_m}) \mid m = 1, \ldots, p\}$ is further randomly picked up from π_i, where $x_j \succ_i x_q$ denotes a ordered pair indicating that the labeled image x_j is ranked before x_q in π_i, and $w(x_j \succ_i x_q)$ is its corresponding importance weight. To determine the value of $w(x_j \succ_i x_q)$, we first define π_i' as a new ranking of $\mathcal{T} \backslash x^{(i)}$, which exchanges the positions of x_j and x_q in π_i. Then we calculate the NDCG@k metric of π_i':

$$NDCG_{\pi_i'}@k = \frac{1}{N_k} \sum_{l=1}^{k} \frac{2^{r_l} - 1}{\log_2(1+l)} \ , \tag{2}$$

where r_l denotes the relevance degree of the image with position l in π_i', and N_k is a normalization factor chosen so that the NDCG@k of the original ranking π_i is 1. Furthermore, the exact form of $w(x_j \succ_i x_q)$ is given as

$$w(x_j \succ_i x_q) = \begin{cases} 1 - NDCG_{\pi_i'}@k & \pi_i(x_j) \le k \\ \eta & otherwise \end{cases} \ . \tag{3}$$

In the above, if x_j or x_q involves the top-k instances in π_i, we take the drops of π_i' in terms of NDCG@k as the value of $w(x_j \succ_i x_q)$. Intuitively, as NDCG includes a position discount factor in its definition, incorrectly ordering higher ranks of π_i can lead to greater losses in terms of NDCG@k. As a result, a large weight will be assigned to the ordered pair at high positions. On the contrary, if both x_j and x_q appear behind the kth position in π_i, there is no effect on the value of NDCG@k when their positions exchange. Therefore, we set $w(x_j \succ_i x_q)$ to be η, which is a small constant. Finally, we repeat the above process for each labeled image and give the ultimate set of preference information as $\mathcal{P} = \cup_{i=1}^{n} \mathcal{W}_i$, which will be used as input training data for the following LTR algorithm.

2.2 Top-k Focused Ranking Algorithm

With the derived preference information set $\mathcal{P}$, we now present the formulation of the proposed top-k focused LTR algorithm. The basic idea is that the optimal ranking function H should be consistent with the preference information in $\mathcal{P}$ as much as possible. To this end, we define the ranking error of H with respect to $\mathcal{P}$ as follows:

$$err = \sum_{x_j \succ_i x_q \in \mathcal{P}} W_{ijq} I(H_{iq} \ge H_{ij}) \ . \tag{4}$$

Here, we introduce $W_{ijq} = w(x_j \succ_i x_q)$ and $H_{ij} = H(x^{(i)}, x_j)$ for the simplicity of description. $I(\cdot)$ is an indicator function that outputs 1 if the input boolean variable is true and zero otherwise. In fact, err measures the weighted number of the preference pairs misordered by H. As described in Section 2.1, the preference pairs at high positions have relatively larger weights, thus the incorrect orders of these pairs will result in more severe ranking errors; whereas the pairs only involving the instances behind the kth positions have been assigned small weights, and misordering them may affect little on the error. As a result, through minimizing err, we can find the optimal ranking function H that gives priority to ensuring the correctness of the top-k results.

However, the ranking error defined in (4) is a non-smooth function as the indicator function $I(\cdot)$ is non-smooth. It is well known that directly optimizing a non-smooth function is computationally infeasible. To address the problem, we follow the idea of AdaBoost algorithm by replacing the indicator function $I(x \geq y)$ with an exponential function $exp(x - y)$. The resulting new ranking error is:

$$\widehat{err} = \sum_{x_j \succ_i x_q \in P} W_{ijq} \exp(H_{iq} - H_{ij}) \ . \tag{5}$$

Since it always holds that $exp(x - y) \geq I(x \geq y)$, by minimizing the new error $\widehat{err}$, we effectively reduce the original ranking error err. Besides, another advantage of using $\widehat{err}$ is from the theoretical property of AdaBoost, i.e., minimizing the exponential loss can not only reduce the training errors but also increase the margins of the training samples, and the enlarged margins are the key to ensure a low generalization error for test instances.

In our study, we utilize the RankBoost [1] algorithm to learn the optimal ranking function H by minimizing $\widehat{err}$. To guarantee the correct execution of the algorithm, we need to give a group of ranking features $\mathcal{F} = \{f_1, \ldots, f_g\}$, where each ranking feature f_i defines a linear ordering of the images to be ranked. To this purpose, we calculate the distance of all ranked images to the query image in the space of a certain visual feature, and a ranking feature is generated in ascending order of the distance. It should be noted that the ranking features are only related to the ordering of the ranked images rather than the actual numerical values of their distance. Algorithm 1 shows the details of the RankBoost algorithm. The algorithm operates for T iterations. For each successive iteration $t = 1, 2, \ldots, T$, it maintains a weight distribution $D^{(t)}$ over the preference pairs in $\mathcal{P}$, and denotes $D^{(t)}_{ijq}$ as the weight on the pair $x_j \succ_i x_q$. Initially, the weights are set according to the importance of preference pairs (line 1). At iteration t, a weak ranker h_t is created from $\mathcal{F}$ based on the current weight distribution $D^{(t)}$ (line 3). We use the same generation process of weak ranker as described in [1]. Then the algorithm chooses a weight coefficient α_t for h_t by measuring its ranking accuracy on all preference pairs (line 4). Intuitively, a greater coefficient is given to the more accurate weak ranker. Meanwhile, the weight distribution $D^{(t)}$ is updated according to the performance of h_t (line 5). The preference pairs misordered by h_t have their weights increased, whereas the weights are decreased for those pairs that are ordered correctly. Therefore, the weak ranker in the next iteration h_{t+1} will concentrate more on the "hard" pairs for h_t. Once all the weak rankers have been created, the algorithm outputs the final ranking function H through their weighted combination, and α_t is the corresponding contribution of h_t (line 7).

Algorithm 1 Rankboost algorithm for minimizing $\widehat{err}$

Input: $\mathcal{P}$, $\mathcal{F}$ and T
Output: H

1: Initialize a distribution D over all preference pairs in $\mathcal{P}$: $D^{(1)}_{ijq} = \frac{W_{ijq}}{Z_0}$ where $Z_0 = \sum_{x_j \succ_i x_q} W_{ijq}$
2: **for** $t = 1, \ldots, T$ **do**
3: Create a weak ranker $h_t : R^r \times R^r \to R$ from $\mathcal{F}$
4: Compute $\alpha_t = \frac{1}{2} \ln(\frac{1+r}{1-r})$ where $r = \sum_{x_j \succ_i x_q} D^{(t)}_{ijq}(h_t(x^{(i)}, x_j) - h_t(x^{(i)}, x_q))$
5: Update $D^{(t+1)}_{ijq} = \frac{D^{(t)}_{ijq} \exp(h_t(x^{(i)}, x_q) - h_t(x^{(i)}, x_j))}{Z_t}$ where $Z_t = \sum_{x_j \succ_i x_q} D^{(t)}_{ijq} \exp(h_t(x^{(i)}, x_q) - h_t(x^{(i)}, x_j))$
6: **end for**
7: **return** $H(q, x) = \sum_{t=1}^{T} \alpha_t h_t(q, x)$

After finding the optimal ranking function H, given a new image q, we employ H to produce the total ranking of all labeled images with respect to q, and take the top-k results as its k nearest neighbors, which is denoted by $\mathcal{N}_H(q) = \{NN_1, \ldots, NN_k\}$.

3. IMAGE ANNOTATION WITH DERIVED NEIGHBORS

With the set of neighbor images $\mathcal{N}_H(q)$, the next step is to evaluate the keyword relevance and propagate a certain number of the most relevant keywords to the new image q. In most previous work, researchers determined the relevance of a keyword by the majority or weighted voting of the nearest neighbors. However, there is no theoretical guarantee that the keywords selected by this manner are always the suitable annotations for q. In [2], Li et al. demonstrated that the difference between the keyword frequency in local neighbor set and that in entire image collection is a good keyword relevance indicator. Therefore, in our study, we adopt the similar method to compute the relevance of the keyword w with respect to q:

$$rel(w, q) = kf_{\mathcal{N}_H(q)}(w) - kf_{prior}(w) \ , \tag{6}$$

where $kf_{\mathcal{N}_H(q)}(w)$ is the number of labeled images containing w in $\mathcal{N}_H(q)$, and $kf_{prior}(w)$ denotes the total frequency of w in the entire training collection.

4. EXPERIMENTS

4.1 Experiment Settings

We conduct experiments on two benchmark datasets: Corel 5K and IAPR TC 12. The two datasets have been widely used in previous studies so we can directly compare the experiment results. Each image on both datasets is represented with the same visual features as described in [4].

On Corel 5K and IAPR TC12, all comparative methods are required to annotate each image with 5 most relevant keywords. The quality of predicted annotations is assessed by retrieving test images using the keywords in annotation

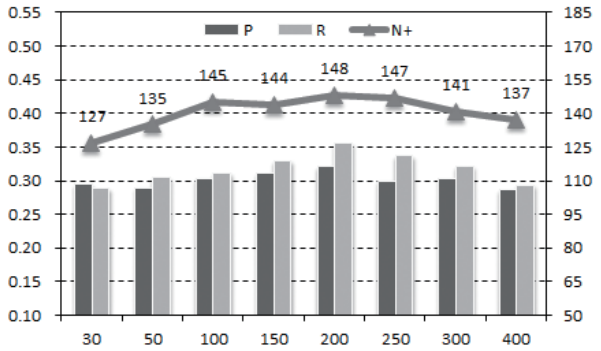

Figure 1: Effect of the variation of k on Corel 5K.

vocabulary. For a keyword w, its precision and recall is computed as follows:

$$Precision(w) = \frac{N_w}{N_p} \quad Recall(w) = \frac{N_w}{N_r}\ , \tag{7}$$

where N_w denotes the number of images correctly annotated with w, N_p denotes the number of images predicted to have w, and N_r is the number of images annotated with w in ground-truth. The average precision (P) and recall (R) are computed over all keywords as two evaluation measures. In addition, we also consider another measure to assess the coverage of correctly annotated keywords, i.e., the number of keywords with non-zero recall ($N+$).

In our approach, the number of neighbors considered in the nearest-neighbor based scheme, k, is a parameter to be determined. In the experiments, the optimal value of k is found via an exhaustive search with a 5-fold cross-validation on training set. Figure 1 presents the performance comparisons by varying k from 30 to 400 on Corel 5K. We can see that the best results can be achieved when $k = 200$, and a very small or large value of k degrades the performance. This is reasonable because a small number of neighbors cannot provide sufficient information to reflect the characteristics of a new image, while too many neighbors may introduce some information irrelevant to that image. On IAPR TC12, similar variation trend of performance can be observed as k changes, and the optimal value of k is around 500. Therefore, we set $k = 200$ for Corel 5K and $k = 500$ for IAPR TC12 in our later experiments.

4.2 Experiment Results

To investigate the efficacy of our ranking-oriented nearest-neighbor based method for image annotation, which is denoted by RNN, we compare it with some previous distance-oriented methods, i.e., MSC [5], JEC [4], LASSO [4] and GS [6]. Besides, we design a modified version of our original method, NW-RNN, which adopts a similar ranking algorithm to RNN but without considering the procedure of preference pair weighting in equation (3). Instead, NW-RNN assigns equal weight to all preference pairs. As a result, it is difficult for NW-RNN to ensure the correctness of the top-ranked results sufficiently. Table 1 shows the annotation results of different approaches.

As clearly observed in the table, on Corel 5K, NW-RNN gains comparable performance with RNN in $N+$, but loses a lot a in terms of P and R respectively. Such results underlines the importance of focusing more on the correctness of the top-ranked results for the success of our ranking-oriented neighbor search mechanism. In addition, RNN outperforms all the distance-oriented approaches listed in different evaluation measures. The performance increase over the best distance-oriented method (GS) still achieves 1% , 1% and 3 in terms of P, R and $N+$. On IAPR TC12, RNN is superior to other approaches as well. These improvements suggest that the annotation results provided by our method is preferable.

Table 1: Performance comparison in terms of $P\%, R\%$ and $N+$ between our method and previous published work.

	Corel 5K			IAPR TC12		
	P%	**R%**	**N+**	**P%**	**R%**	**N+**
MSC	25	32	136	—	—	—
JEC	27	32	139	28	29	250
LASSO	24	29	127	28	29	246
GS	30	33	146	32	29	252
NW-RNN	29	32	149	28	30	259
RNN	**31**	**34**	**149**	**33**	**31**	**255**

5. CONCLUSIONS

In this paper, we have introduced a novel image annotation method, which adapts the conventional nearest-neighbor based approaches with a ranking-oriented neighbor search mechanism. A new learning to rank algorithm is developed to directly produce the ordering of all labeled images. It leverages the implicit preference information of training data and underlines the accuracy of the top-ranked results. Experiments have demonstrated the effectiveness of our method for image annotation. For future study, we plan to examine the scalability of our method and experiment on large-scale web image datasets.

6. ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China (61272240,60970047,61103151), the Doctoral Fund of Ministry of Education of China (20110131110028) and the Natural Science Foundation of Shandong Province (ZR2012FM037).

7. REFERENCES

[1] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933–969, 2003.

[2] X. Li, C. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *Multimedia, IEEE Transactions on*, 11(7):1310–1322, 2009.

[3] T.-Y. Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, 2009.

[4] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*, pages 316–329, 2008.

[5] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding for automatic image annotation. In *CVPR*, pages 1643–1650, 2009.

[6] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. Metaxas. Automatic image annotation using group sparsity. In *CVPR*, pages 3312–3319, 2010.