Multiple-Instance Ranking: Learning to Rank Images for Image Retrieval

Yang Hu[1] yanghu@ustc.edu
Mingjing Li[2] mjli@microsoft.com
Nenghai Yu[1] ynh@ustc.edu.cn

[1]MOE-Microsoft Key Lab of MCC
University of Science and Technology of China
Hefei 230027, China

[2]Microsoft Research Asia
49 Zhichun Road, Beijing 100190, China

Abstract

We study the problem of learning to rank images for image retrieval. For a noisy set of images indexed or tagged by the same keyword, we learn a ranking model from some training examples and then use the learned model to rank new images. Unlike previous work on image retrieval, which usually coarsely divide the images into relevant and irrelevant images and learn a binary classifier, we learn the ranking model from image pairs with preference relations. In addition to the relevance of images, we are further interested in what portion of the image is of interest to the user. Therefore, we consider images represented by sets of regions and propose multiple-instance rank learning based on the max margin framework. Three different schemes are designed to encode the multiple-instance assumption. We evaluate the performance of the multiple-instance ranking algorithms on real-word images collected from Flickr - a popular photo sharing service. The experimental results show that the proposed algorithms are capable of learning effective ranking models for image retrieval.[1]

1. Introduction

The principal function of an image retrieval system is to rank a set of images according to how well they meet user's information needs. In order to facilitate the index and the search of images in the query-by-keyword scenario (QBK, which is widely used in commercial image search engines), much research effort has been devoted to image classification and automatic image annotation over the past years. However, the ranking problem remains far from being solved. Even images indexed or annotated by the same keyword should have different orders during ranking. For example, consider the images shown in Fig.1, which are drawn from Flickr [1], a popular photo sharing service.

(a) (b) (c)

Figure 1. Three images from Flickr, all are tagged by users with "elephant". When the user wants to find representative elephant images, they would like to get a list where (a) ranks higher than (b) and (b) ranks higher than (c).

They all are tagged by users with "elephant". When user wants to find representative elephant images, it is reasonable to rank Fig.1(a) higher than Fig.1(b) because only some unimportant parts of the elephant are shown in Fig.1(b). However, when compared with Fig.1(c), Fig.1(b) is more preferable since Fig.1(c) only contains an elephant toy instead of the real animal.

In this work, we are interested in learning to rank the images. Given a set of images indexed or tagged by the same keyword, we learn a ranking function from some training examples. So for a set of new images, we can compute their ranking scores using the learned model and rank them according to their scores. The problem of learning to rank has been explored a lot recently in text information retrieval [4, 5, 12, 17]. Related work for image retrieval, however, are still limited.

In particular, during training, we consider relative and qualitative examples of the form "image A is more relevant than image B". This kind of feedback is more flexible than traditional relevance feedback for image retrieval, where images are coarsely divided into relevant and irrelevant images. And feedback of this type is supposed to be more easily available than quantitative examples, such as "the relevance score of image A is 2.75". The query logs of the search engines are one example where ranking preferences between image pairs are readily available for learning [17]. Given a ranked list of images for a query, it is rea-

[1]This work was performed at Microsoft Research Asia.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

sonable to assume that users are more interested in images they click than those images that they have observed but decide not to click on. Moreover, quantitative labels provided by different users may be inconsistent, *e.g.* the ratings of 3 by two different users may indicate different degrees of relevance. Therefore, some kinds of calibration may be needed in advance [7]. By converting their ratings to preference relations, we do not need to calibrate them any more.

In addition to the relevance of images, we also want to know what portion of the image is of interest to the user. Therefore, instead of using global information, we would like to represent an image by a set of regions. However, during training, we only have preference relations between image pairs. No explicit information about regions is provided. The gap between the label information and the data representation is quite similar with the setting in multiple-instance (MI) learning. Therefore, in this work, we investigate the rank learning problem in the MI setting and regard our algorithms as *multiple-instance ranking*. Similar to some previous work in rank learning [5, 15, 17] and MI learning [2, 9], we formulate the problem in the large margin framework. We investigate three different schemes to encode the MI assumption, which associate the ranking score of the image with its constituent regions. In the experiment, we compare the performance of these algorithms on images collected from Flickr and show their ability to learn effective ranking model for image retrieval.

The rest of the paper is organized as follows. In Section 2, we discuss related work in rank learning and MI learning. In Section 3, we briefly introduce the Ranking SVM algorithm, which forms the basis of our algorithm. We describe in details the multiple-instance ranking algorithms in Section 4. Our dataset, the experiments and the results are presented in Section 5. Finally we conclude the paper in Section 6.

2. Related Work

The importance of the ranking problem has inspired numerous approaches to handle it especially in the context of information retrieval. One typical direction of rank learning is to formulate it as ordinal regression, *i.e.*, learning the mapping of the examples to an ordered set of numerical ranks [10, 21]. An alternative direction which has also been investigated is to learn the ranking function from relative ranking preferences between example pairs. For example, based on the boosting approach, Freund *et al*. [12] proposed the RankBoost algorithm which learns to rank a set of objects by combining a given collection of preference functions. In [4], Burges *et al*. introduced RankNet, which used neural network and gradient descent methods to learn ranking functions using pairs of training examples. Besides, Herbrich *et al*. [15] applied the large margin principle used in Support Vector methods to the rank learning problem and proposed the Ranking SVM algorithm. In [17], Joachims used the Ranking SVM algorithm to learn retrieval functions from the clickthrough data of the search engine. And Cao *et al*. [5] adapted Ranking SVM to document retrieval by modifying the cost function. Although the idea of learning to rank has been extensively exploited in text information retrieval, related work in image retrieval are still very limited. Moreover, since the text document is usually represented by a single feature vector, these methods can not be directly applied to images represented by a set of regions.

Another set of related work are those that learn distance metric from training data. In [20], Schultz and Joachims investigated the problem of learning a distance metric from relative comparisons in the form of "A is closer to B than A is to C". They formulated the problem in the large margin framework and proposed a method that extended the standard SVM algorithm. The work that is most closely related is that presented in [13]. In this work, based on the work of Schultz and Joachims [20], Frome *et al*. proposed to learn local distance functions for image retrieval and classification. They learned a distance function for each training image. This is quite similar in spirit to our work, which learns the ranking function for each query. However, their training data are in the form of triplets of images, which include the reference image. And our training data consist of image pairs without explicit reference. We believe this assumption is more flexible. For example, the typical elephant images are actually quite various and it is difficult to specify which one should be the reference. Frome's work resembles our work also in that they used a set of patch-based features to represent each image instead of using a fixed-length feature. However, they focused on learning the weights for combining the elementary distances. And we focus on learning the relevance scores for the regions. In [14], Frome *et al*. extended their work to learn globally-consistent local distance functions. This work is still in the context of distance learning and differs our work just like [13].

Multiple-instance learning was first introduced by Dietterich *et al*. [11] in the context of drug activity prediction. Unlike standard supervised learning in which each instance is labeled in the training data, here the labels are associated with sets of patterns, or bags. This problem has been studied by a lot of researchers and many algorithms have been developed, such as Diverse Density [18], EM-DD [24], MI-SVM and mi-SVM [2]. The work that are closely related are the SVM based algorithms. For example, in [2], Andrews *et al*. modified and extended the SVM to deal with MI learning problems. They associated the dual variables with the instances, which leaded to mixed integer quadratic programs. And they proposed two simple optimization heuristics to solve the problem. In [9], Cheung and Kwok provided a regularization framework for MI learning by allowing the use of different loss functions between the outputs of

a bag and its associated instances. Instead of using heuristics, they use the constrained concave-convex procedure to solve the problem, which inspired our treatment of the MI ranking problem. MI learning has been widely exploited for content-based image retrieval [6, 8, 25], however, they are mainly in the classification setting. In this work, we embed the rank learning problem in the MI setting and apply it to image retrieval.

3. The Ranking SVM Algorithm

In a rank learning problem, we are usually given a set of training examples $\{\mathbf{x}_i\}_{i=1}^m \in \mathbb{R}^n$ and a set of relative comparisons between example pairs. Assume that the preference relation that $\mathbf{x}_i$ is preferable to $\mathbf{x}_j$ is denoted by $\mathbf{x}_i \succ \mathbf{x}_j$. The goal is to induce a ranking function $f : \mathbb{R}^n \to \mathbb{R}$ that fulfills the set of constrains

$$\forall\, \mathbf{x}_i \succ \mathbf{x}_j : f(\mathbf{x}_i) > f(\mathbf{x}_j). \tag{1}$$

We refer to the value of $f(\mathbf{x}_i)$ as the ranking score of $\mathbf{x}_i$. In text information retrieval, f is usually assumed to be a linear function. In this case, we have $f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle$.

Unfortunately, this problem proves to be NP-hard. And just like in classification SVMs, an approximate solution can be obtained by introducing (non-negative) slack variables ξ_{ij} and minimizing the upper bound $\sum \xi_{ij}$. In addition, L2 regularization is imposed on $\mathbf{w}$ to maximize the margin between the closest projections. And these lead to the following optimization problem

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \gamma \sum_{i,j} \xi_{ij} \\ \text{s.t.} \quad & \forall\, \mathbf{x}_i \succ \mathbf{x}_j : \langle \mathbf{w}, \mathbf{x}_i \rangle \geq \langle \mathbf{w}, \mathbf{x}_j \rangle + 1 - \xi_{ij}, \\ & \forall\, i, j : \xi_{ij} \geq 0, \end{aligned} \tag{2}$$

which is referred to as the Ranking SVM algorithm [15, 17]. Note that if we rearrange the constrains in Eq.(2) as

$$\forall\, \mathbf{x}_i \succ \mathbf{x}_j : \langle \mathbf{w}, \mathbf{x}_i - \mathbf{x}_j \rangle \geq 1 - \xi_{ij}, \tag{3}$$

the optimization problem becomes equivalent to a classification SVM on pairwise example differences. Therefore, it can be solved using algorithms similar to those used for SVM classification. Assume that $\mathbf{w}^*$ is the solution that optimizes Eq.(2). For a set of new examples, we can get the ranking score for each of them using

$$f(\mathbf{x}) = \langle \mathbf{w}^*, \mathbf{x} \rangle, \tag{4}$$

and then rank these examples according to their scores.

4. Multiple-Instance Ranking

In previous rank learning algorithms, each example is assumed to be represented by a single feature vector, which is reasonable for text data. In this work, however, we assume that an image consists of a set of segments, each characterized by a feature vector. In this case, the previously developed rank learning algorithms, such as Ranking SVM, are not directly applicable to image data, since the preference relations are associated with the whole images, and we don't have explicit labels for the segments. The gap between the label information and the data representation is quite similar with the settings in MI learning. However, most previous work on MI learning mainly focused on classification and regression problems. Here we exploit the ranking problem under the MI assumption.

In multiple-instance ranking, we are given a set of training bags $\{B_i\}_{i=1}^m$, where each bag contains a set of instances, *i.e.* $B_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{in_i}\}$. Each bag corresponds to an image and each instance in the bag corresponds to a sub-region of the image. We are also given a set of preference relations between bag pairs, denoted by $B_i \succ B_j$. Let $\mathcal{X}$ be the instance space and $\mathcal{H}$ be a Reproducing Kernel Hilbert Space (RKHS) of functions $f : \mathcal{X} \to \mathbb{R}$, with associated kernel function k. We assume that the ranking score of the bag B_i is determined by the scores of the instances it contains, *i.e.*

$$g(B_i) = g(\{f(\mathbf{x}_{il})\}_{l=1}^{n_i}). \tag{5}$$

Denote the RKHS norm of $\mathcal{H}$ by $\|f\|_{\mathcal{H}}$. The maximal margin formulation of multiple-instance rank learning is

$$\begin{aligned} \min_{f\in\mathcal{H},\boldsymbol{\xi}} \quad & \frac{1}{2}\|f\|_{\mathcal{H}}^2 + \gamma \sum_{i,j} \xi_{ij} \\ \text{s.t.} \quad & \forall\, B_i \succ B_j :\ g(B_i) \geq g(B_j) + 1 - \xi_{ij}, \\ & \forall\, i, j :\ \xi_{ij} \geq 0. \end{aligned} \tag{6}$$

We use the representer theory to reduce the optimization problem from a possibly infinite-dimensional space to a finite-dimensional space. Without loss of generality, assume that the instances in the training set are ordered as $\{\mathbf{x}_{11}, \ldots, \mathbf{x}_{1n_1}, \mathbf{x}_{21}, \ldots, \mathbf{x}_{m1}, \ldots \mathbf{x}_{mn_m}\}$. And each instance is indexed by $\mathcal{I}(\mathbf{x}_{il}) = \sum_{r=1}^{i-1} n_r + l$. Let $n = \sum_{i=1}^m n_i$ be the total number of instances in the training set. We can obtain a $n \times n$ kernel matrix $\mathbf{K}$ defined on all training instances. Denote the ith column of $\mathbf{K}$ by $\mathbf{k}_i$. Using the representer theorem, we have $f(\mathbf{x}_{il}) = \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha} + b$. Moreover, $\|f\|_{\mathcal{H}}^2 = \boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha}$.

To solve the optimization problem in Eq.(6), we need to figure out how to get the ranking score of a bag given the scores of its member instances. Obviously, there are lots of ways to encode the multiple-instance assumption in Eq.(5), we investigate three of them in the following.

4.1. Using the Average of the Instances' Scores

As the most simple case, we assume that the score of a bag is the average of its member instances' scores, *i.e.*

$g(B_i) = \frac{1}{n_i}\sum_{l=1}^{n_i} \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha} + b$. Therefore, the member instances contribute equally to the score of the bag. We can write Eq.(6) as

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} + \gamma\sum_{i,j}\xi_{ij} \\ \text{s.t.} \quad & \forall\, B_i \succ B_j : \\ & \frac{1}{n_i}\sum_{l=1}^{n_i}\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha} \geq \frac{1}{n_j}\sum_{l=1}^{n_j}\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{jl})}\boldsymbol{\alpha} + 1 - \xi_{ij}, \\ & \forall\, i,j:\ \xi_{ij} \geq 0, \end{aligned} \tag{7}$$

which is a standard QP problem. Note that we omit the bias b in Eq.(7) because it has no influence on the ranking result.

4.2. Using the Max of the Instances' Scores

In multiple-instance classification, a bag is "positive" if at least one of its member instances is a positive instance. To adapt SVM to deal with multiple-instance classification problem, Andreas *et al*. [2] expressed the relation between the bag label information and the instance label information as

$$g(B_i) = \max_{l=1,\ldots,n_i} f(\mathbf{x}_{il}), \tag{8}$$

where f is the function to be learned by SVM.

In the second case, we inherit this assumption and assume that the score of a bag is determined by the member instance with the max score, as shown in Eq.(8). Therefore, we can write Eq.(6) as

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} + \gamma\sum_{i,j}\xi_{ij} \\ \text{s.t.} \quad & \forall\, B_i \succ B_j : \\ & \max_{l=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}) - \max_{l=1,\ldots,n_j}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{jl})}\boldsymbol{\alpha}) \geq 1 - \xi_{ij}, \\ & \forall\, i,j:\ \xi_{ij} \geq 0. \end{aligned} \tag{9}$$

This is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} + \gamma\sum_{i,j}\xi_{ij} \\ \text{s.t.} \quad & \forall\, B_i \succ B_j : \\ & \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{jr})}\boldsymbol{\alpha} - \max_{l=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}) \leq \xi_{ij} - 1, \\ & r = 1,\ldots,n_j, \\ & \forall\, i,j:\ \xi_{ij} \geq 0. \end{aligned} \tag{10}$$

Note that the first constraint is nonlinear but a difference between two convex functions instead. To solve this we use the constrained concave-convex procedure (CCCP) developed by Smola *et al*. [23]. Given the following optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) - g_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) - g_i(\mathbf{x}) \leq c_i, \quad i = 1,\ldots,m, \end{aligned} \tag{11}$$

where $f_i, g_i (i = 0,\ldots,m)$ are real-valued, convex and differentiable functions on $\mathbb{R}^n$ and $c_i \in \mathbb{R}$, the CCCP for the problem in Eq.(11) is shown in Fig. 2. It can be shown that CCCP converges to a local minimum solution.

Initialize $\mathbf{x}$ as $\mathbf{x}^{(0)}$.
repeat
Replace $g_i(\mathbf{x})$ with its first-order Taylor expansion at $\mathbf{x}^{(t)}$, and then set $\mathbf{x}^{(t+1)}$ to the solution of the following relaxed optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) - [g_0(\mathbf{x}^{(t)}) + \langle \partial_{\mathbf{x}} g_0(\mathbf{x}^{(t)}), \mathbf{x} - \mathbf{x}^{(t)}\rangle] \\ \text{s.t.} \quad & \forall\, i : f_i(\mathbf{x}) - [g_i(\mathbf{x}^{(t)}) + \langle \partial_{\mathbf{x}} g_i(\mathbf{x}^{(t)}), \mathbf{x} - \mathbf{x}^{(t)}\rangle] \leq c_i \end{aligned} \tag{12}$$

until convergence of $\mathbf{x}^{(t)}$.

Figure 2. Constrained concave-convex procedure (CCCP) [23].

Since $\max_{l=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha})$ in Eq.(10) is non-smooth, to use the CCCP, we should replace the gradients by the subgradients. We adopt the subgradient of $\max_{l=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha})$ used by Cheung and Kwok in [9]:

$$\partial\big(\max_{l=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha})\big) = \sum_{l=1}^{n_i}\beta_{il}\mathbf{k}_{\mathcal{I}(\mathbf{x}_{il})}, \tag{13}$$

where

$$\beta_{il} = \begin{cases} 0, & \text{if } \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha} \neq \max_{r=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{ir})}\boldsymbol{\alpha}), \\ 1/n_a, & \text{otherwise.} \end{cases} \tag{14}$$

n_a is the number of $\mathbf{x}_{il}$ for which $\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha} = \max_{r=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{ir})}\boldsymbol{\alpha})$. At the tth iteration, denote the current estimation of $\boldsymbol{\alpha}$ and the corresponding β_{il} by $\boldsymbol{\alpha}^{(t)}$ and $\beta_{il}^{(t)}$ respectively. CCCP replaces $\max_{l=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha})$ in the constraint of Eq.(10) by

$$\begin{aligned} \max_{l=1,\ldots,n_i}(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}^{(t)}) + \sum_{l=1}^{n_i}\beta_{il}^{(t)}\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(t)}) \\ = \sum_{l=1}^{n_i}\beta_{il}^{(t)}\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}. \end{aligned} \tag{15}$$

The optimization problem of Eq.(12) in our case is then

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} + \gamma \sum_{i,j} \xi_{ij} \qquad (16) \\ \text{s.t.} \quad & \forall\, B_i \succ B_j : \\ & \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{jr})}\boldsymbol{\alpha} - \sum_{l=1}^{n_i} \beta_{il}^{(t)} \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha} \leq \xi_{ij} - 1,\ r = 1, \ldots, n_j, \\ & \forall\, i, j :\ \xi_{ij} \geq 0, \end{aligned}$$

which is a standard QP problem. Following CCCP, the solution $\boldsymbol{\alpha}$ of the QP is then used as $\boldsymbol{\alpha}^{(t+1)}$, from which we can get $\beta_{il}^{(t+1)}$, and the iteration continues until convergence.

4.3. Using the Softmax of the Instances' Scores

In the first case, the member instances contribute equally to the ranking score of the bag. While in the second case, only the instance with maximal score is responsible for the score given to the bag. In the third case, we adopt a scheme which is a compromise between these two methods. We let all instances in a bag explicitly contribute to the ranking score of the bag. However, their contributions are different. Instances with higher scores are weighted heavier than instances with lower scores. More specifically, we consider the following *log-sum-exp* function:

$$g(B_i) = \log\Big(\frac{1}{n_i} \sum_{l=1}^{n_i} e^{\eta(\mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha} + b)}\Big). \qquad (17)$$

Note that the *log-sum-exp* function is convex. Therefore, if we substitute Eq.(17) into Eq.(6), we will again get an optimization problem with quadratic objective function and concave-convex constraints. Although we can also use the CCCP to solve this problem, the relaxed optimization problem of Eq.(12) will no longer be a QP problem in this case. Consider the terms that depend on $\boldsymbol{\alpha}$ in the first-order Taylor expansion of Eq.(17) at $\boldsymbol{\alpha}^{(t)}$, *i.e.*

$$\frac{\sum_{l=1}^{n_i} e^{\eta \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}^{(t)}} \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}}{\sum_{l=1}^{n_i} e^{\eta \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}^{(t)}}}. \qquad (18)$$

It can be regarded as a weighted average of the scores of the member instances, where the weights of the instances depend on their scores under $\boldsymbol{\alpha}^{(t)}$. According to this property, we use Eq.(18) to approximate the bag's ranking score in Eq.(17) and design an iterative algorithm to solve the problem. At the tth iteration, denote the current estimation of $\boldsymbol{\alpha}$ by $\boldsymbol{\alpha}^{(t)}$. We solve the optimization problem

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} + \gamma \sum_{i,j} \xi_{ij} \qquad (19) \\ \text{s.t.} \quad & \forall\, B_i \succ B_j : \\ & \frac{\sum_{l=1}^{n_i} e^{\eta \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}^{(t)}} \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}}{\sum_{l=1}^{n_i} e^{\eta \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{il})}\boldsymbol{\alpha}^{(t)}}} \\ & \geq \frac{\sum_{l=1}^{n_j} e^{\eta \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{jl})}\boldsymbol{\alpha}^{(t)}} \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{jl})}\boldsymbol{\alpha}}{\sum_{l=1}^{n_j} e^{\eta \mathbf{k}'_{\mathcal{I}(\mathbf{x}_{jl})}\boldsymbol{\alpha}^{(t)}}} + 1 - \xi_{ij}, \\ & \forall\, i, j :\ \xi_{ij} \geq 0, \end{aligned}$$

which is a standard QP problem. And the obtained $\boldsymbol{\alpha}$ from this QP is then used as $\boldsymbol{\alpha}^{(t+1)}$. For initialization, we equally weight the instances in a bag, which is equivalent to Eq.(7).

We can interpret this iterative algorithm from the perspective of EM algorithm, which has been exploited by other MI learning algorithms, such as MI regression [19], EM-DD [24] and MI-SVM [2]. In the *E*-step, we use the current estimation of $\boldsymbol{\alpha}$ to compute the weights of the member instances, which reflect how much they contribute to the score of the whole image. Then in the *M*-step, we refine the estimation of $\boldsymbol{\alpha}$ by solving the QP problem in Eq.(19), which is based on the weights obtained in the *E*-step. By conducting this iterative process, we not only deduce the ranking scores of the instances, but also estimate how important the instances are for the ranking of the image.

5. Experiments

5.1. Dataset

We collect images from Flickr, a popular personal photo sharing service. Users are encouraged to tag images on Flickr, not only those of their own, but also other users'. As a result, it is possible to search images tagged with a specific word. However, images with a common tag do not necessarily meet user's interest to the same extent. And we should rank them according to how well they meet a specific information need.

We send 10 animal queries to Flickr. For each of them, we download 200 images that are tagged with it. Animals are demonstrably among the most difficult classes to recognize. And there has been much fascination with searching animals on the web recently [3]. We set our information need as searching for images that well depict the desired animals. As the ground truth, we manually assign one of the three following labels to the images: (1) *Good* image: This is a good example of the animal category. The animal is the dominant object in the image and is well delineated. (2) *Intermediate* image: The image contains the desired animal. However, the animal is not the dominant object or the

image has extensive occlusion. (3) *Junk* image: It doesn't contain the real animal. The three images in Fig.1 are examples of good, intermediate and junk images respectively. Based on these judgements, we can construct pairwise preference relations between images with different labels for the learning of the ranking functions.

5.2. Performance Measures

To evaluate the quality of a ranking for a set of test images, we use two performance measures commonly used for information retrieval. The first measure is the Normalized Discounted Cumulative Gain (NDCG) [16]. For a list of images sorted in descending order of the scores output by a learned ranking model, the NDCG score at the mth image is computed as

$$N_m = C_m \sum_{j=1}^{m} \frac{2^{r(j)} - 1}{\log(1+j)}, \tag{20}$$

where $r(j)$ is the rating of the jth image and C_m is the normalization constant [4, 5]. We set the rating of the good images as 3, the intermediate images as 2 and the junk images as 1. According to Eq.(20), the gain of a retrieved image is discounted by a ranked position based factor $\log(1+j)$. This is based on the observation that the greater the ranked position of a relevant image, the less valuable it is for the user, because it is less likely that the user will ever examine the image due to time, effort, and cumulated information from images already seen. C_m is chosen so that a perfect ordering gets NDCG scores 1. The perfect ordering is obtained by ranking the images in descending order of their ground truth ratings.

We also use Average Precision (AP) to evaluate the ranking methods. It is the average of precision after each positive (relevant) image is retrieved. We regard good images as positive, intermediate and junk images as negative images. The AP is calculated as

$$\mathrm{AP} = \frac{1}{N_{\mathrm{pos}}} \sum_{j=1}^{N} P(j) \times \mathrm{rel}(j), \tag{21}$$

where N_{pos} is the number of positive images, N is the number of total retrieved images, $\mathrm{rel}(j)$ is a binary function indicating whether the jth image is relevant, and $P(j)$ is the precision at j.

5.3. Experimental Setup

For the experiments, the images are first segmented using Normalized cuts [22]. Then for each region, a set of low level features are extracted, including color correlograms, color moments and Gabor textures. We use the Gaussian kernel for the kernel matrix $\mathbf{K}$.

Images corresponding to each query are randomly partitioned in half to form a training set and a test set. The learning algorithms learn a ranking function for each query using the training images and then rank the test images using the learned model. The parameter γ and σ^2 (Gaussian kernel radius) are selected according to a two-fold cross-validation on the training set. The parameter η for the softmax scheme is set to be 4. Each experiment is repeated five times for five random splits, and the average results are reported.

We compare the rankings obtained by the multiple-instance ranking algorithms with the original Flickr ranking which we compute over the same test images based on their orders in the Flickr search result. We let Flickr rank the images in descending order of their "interestingness", which is supposed to be determined according to the clickthroughs, comments and some other factors. This is the best ranking result in Flickr. Among the three multiple-instance ranking algorithms, the Average method treats the constituent regions equally, which does not fully embody the spirit of multiple-instance assumption. Moreover, through taking average, it actually converts the multiple-instance rank learning problem to a single-instance rank learning problem. Therefore, by comparing the other two methods with it, we can tell, to some extent, how multiple-instance ranking performs compared with single-instance ranking.

5.4. Results

In Table 1, we report the AP for each query using different algorithms. We can see that the multiple-instance ranking algorithms significantly exceed the Flickr ranking. Among the three learning algorithms, the Max scheme achieved comparable overall performance with respect to that of the Average scheme and Softmax consistently outperforms the other two methods. The slight inferiority of the Max scheme in comparison to the Average scheme indicates that for image retrieval application, it is likely that there are several regions in the image which together really explain why the image is desirable. However, treating the regions equally and combining their scores by taking the average is not good enough. We can achieve better performance by weighting more important regions heavier.

We compare the NDCG values at the fifth, tenth and twentieth images in Fig.3. Unlike AP, which reflects the overall ranking quality of the entire list, NDCG reflects the quality of the top ranked images. According to Fig.3, the Softmax scheme outperforms the Average method in NDCG measure just like in AP. And the Max scheme, which does not perform very well according to AP, shows quite pleasant performance this time. This indicates that the Max method is good at discovering some relevant images which have a distinctly relevant region in it. This property makes the Max method quite desirable because users are usually more interested in top results when conducting search.

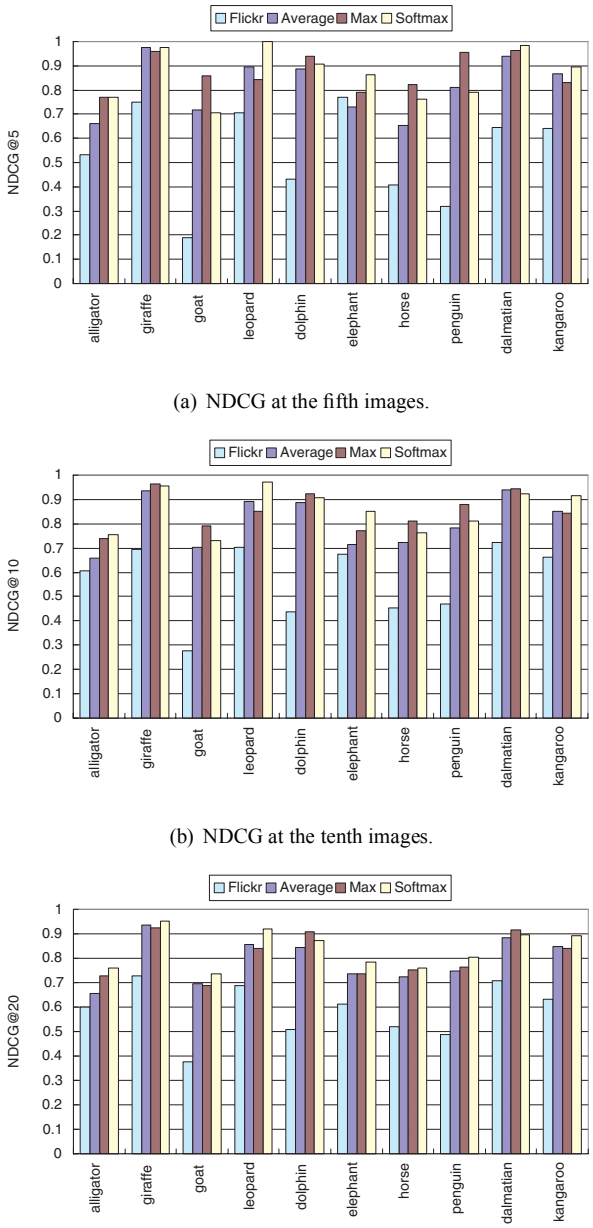
Query	Flickr	Average	Max	Softmax
alligator	0.445	0.503	0.522	**0.575**
giraffe	0.613	0.810	0.781	**0.857**
goat	0.284	0.521	0.482	**0.577**
leopard	0.584	0.721	0.698	**0.798**
dolphin	0.477	0.730	0.741	**0.772**
elephant	0.422	0.570	0.565	**0.622**
horse	0.442	0.571	0.582	**0.596**
penguin	0.388	0.630	0.619	**0.646**
dalmatian	0.594	0.767	0.745	**0.789**
kangaroo	0.580	0.728	0.688	**0.762**

Table 1. Average Precisions for each query. The mean value of the APs for these four algorithms are 0.483, 0.655, 0.642 and 0.699 respectively.

The multiple-instance ranking algorithm can not only rank images but also indicate the relevance of the regions. We show some examples for the Softmax scheme in Fig.4. For each image, we first normalize the scores of the regions to the range $[0, 1]$, *i.e.* the minimal and the maximal region score in the image is 0 and 1 respectively. We then linearly map the scores to gray values. So the brighter the regions in Fig.4 the more relevant they are. Note that since image segmentation is still a hard problem, the segmentation results are not perfect. However, the learning algorithm works even using inaccurate segmentations.

6. Conclusion

In this work we exploit the problem of learning to rank images. Unlike previous work on image retrieval, which usually treat it as a binary classification problem and learn classifiers for it, we directly learn ranking functions, using image pairs with preference relationships between them. We consider images represented by sets of regions and propose multiple-instance ranking based on the max margin framework. Three different schemes are designed to encode the multiple-instance assumption. The algorithms achieved pleasant performance on real-world images collected from Flickr. For future study, we would like to exploit other methods to associate image's score with regions' scores and validate the algorithms by more extensive experiments.



(a) NDCG at the fifth images.

(b) NDCG at the tenth images.

(c) NDCG at the twentieth images.

Figure 3. NDCG values evaluated at the fifth, tenth and twentieth images respectively. They reflect the quality of the top ranked images.

References

[1] Flickr photo sharing service, http://www.flickr.com/.

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002.

[3] T. L. Berg and D. A. Forsyth. Animals on the web. In *CVPR*, 2006.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.

[5] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *SIGIR*, 2006.

[6] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):394–410, 2007.

[7] L. Chen, L. Zhang, F. Jing, K.-F. Deng, and W.-Y. Ma. Ranking web objects from multiple communities. In *CIKM*, 2006.

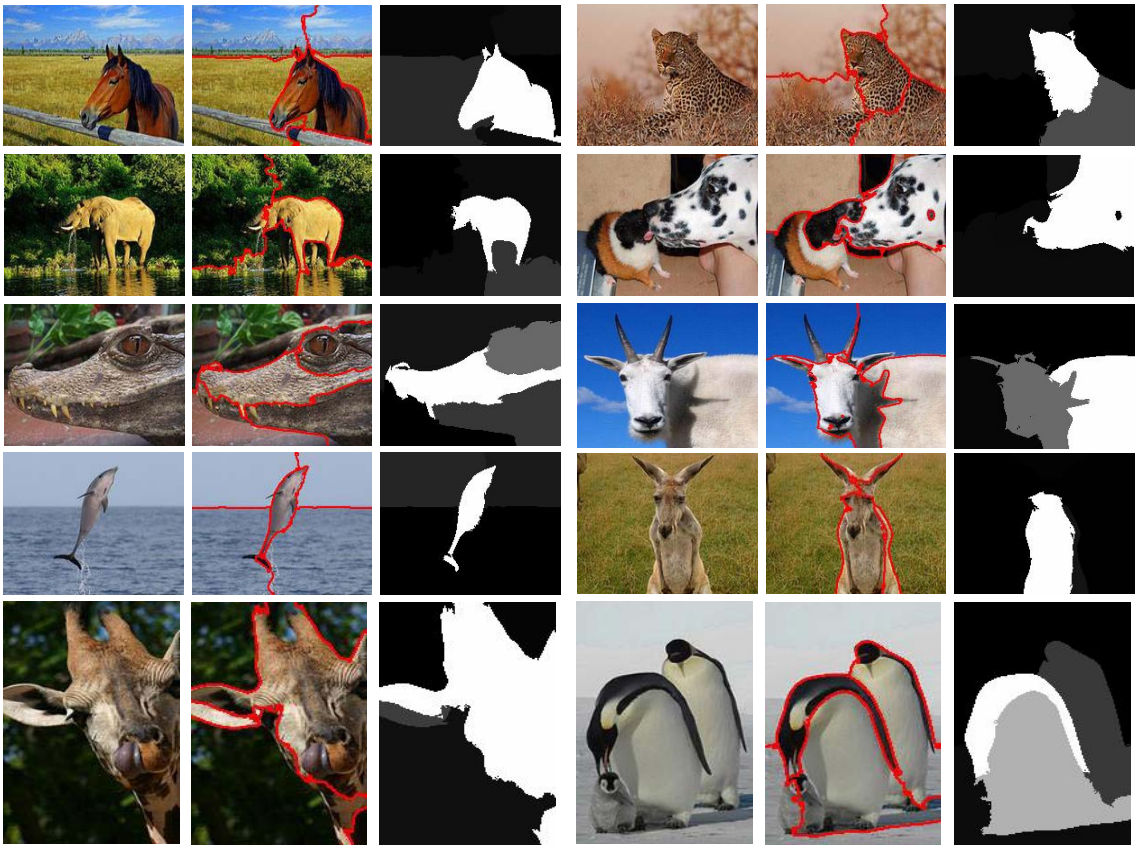


Figure 4. Examplar images show how the multiple-instance ranking algorithms can predict region relevance. The first and the fourth columns show the original images. The second and the fifth columns show the segmented regions. The third and the sixth columns show the relevance of the regions. The brighter the regions the more relevant they are. The Softmax scheme is adopted here.

[8] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939, 2004.

[9] P.-M. Cheung and J. T. Kwok. A regularization framework for multiple-instance learning. In *ICML*, 2006.

[10] K. Crammer and Y. Singer. Pranking with ranking. In *NIPS*, 2001.

[11] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

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

[13] A. Frome, Y. Singer, and J. Malik. Image retrieval and recognition using local distance functions. In *NIPS*, 2006.

[14] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007.

[15] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *NIPS*, 2000.

[16] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[17] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, 2002.

[18] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *NIPS*, 1998.

[19] S. Ray and D. Page. Multiple instance regression. In *ICML*, 2001.

[20] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2004.

[21] A. Shashua and A. Levin. Ranking with large margin principle: Two approaches. In *NIPS*, 2003.

[22] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[23] A. Smola, S. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proc. of International Workshop on Artificial Intelligence and Statistics*, 2005.

[24] Q. Zhang and S. A. Goldman. EM-DD: An improved multiple-instance learning technique. In *NIPS*, 2001.

[25] Q. Zhang, S. A. Goldman, W. Yu, and J. Fritts. Content-based image retrieval using multiple-instance learning. In *ICML*, 2002.