

Pseudo Relevance Feedback Based on Iterative Probabilistic One-Class SVMs in Web Image Retrieval*

Jingrui He¹, Mingjing Li², Zhiwei Li², Hong-Jiang Zhang², Hanghang Tong¹,
and Changshui Zhang³

¹ Automation Department, Tsinghua University, Beijing 100084, P.R.China
{hejingrui98, walkstar98}@mails.tsinghua.edu.cn

² Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, P.R.China
{mjli, i-zli, hjzhang}@microsoft.com

³ Automation Department, Tsinghua University, Beijing 100084, P.R.China
zcs@tsinghua.edu.cn

Abstract. To improve the precision of top-ranked images returned by a web image search engine, we propose in this paper a novel pseudo relevance feedback method named iterative probabilistic one-class SVMs to re-rank the retrieved images. By assuming that most top-ranked images are relevant to the query, we iteratively train one-class SVMs, and convert the outputs to probabilities so as to combine the decision from different image representation. The effectiveness of our method is validated by systematic experiments even if the assumption is not well satisfied.

1 Introduction

With the ever-growing volume of digital images on the World Wide Web, the problem of how to effectively manage and index this huge image resource constantly draws people's research attention, and various web images search engines have been developed to address this issue, such as Google Image Search¹, AltaVista Image Search², AllTheWeb Picture Search³, etc.

When performing image retrieval using these text-based search engines, we can often find that some top-ranked images are irrelevant to the user's query concept. The problem may be attributed to the following reasons: the multiple meanings of words or phrases used to characterize the content of an image, misplacement of images in a totally irrelevant environment, etc. The removal of those top-ranked irrelevant images is highly desirable from the users' perspective.

One solution to this problem is to introduce relevance feedback into the retrieval process, which asks the user to mark the relevance of some retrieved images. However, in real applications, the user might be reluctant to provide

* This work was performed at Microsoft Research Asia.

¹ <http://www.google.com/imghp>

² <http://uk.altavista.com/image/>

³ <http://www.alltheweb.com>

any feedback information. Another possible solution is to re-rank the retrieved images before presenting them to the user, trying to put the relevant images in the head of the list while the irrelevant ones in the tail. It seems to be a promising way of solving the problem since no user intervention is required.

In the field of information retrieval, document re-ranking has long been studied to refine the retrieval result or to better organize the retrieved images [2][3][4]. Most of the techniques used in document re-ranking can be applied to web image retrieval in parallel. For example, Park et al [6] propose a hierarchical agglomerative clustering method to analyze the retrieved images; Yan et al [12] train SVMs whose positive training data are from the query examples, while negative training data are from negative pseudo relevance feedback; and Lin et al [5] propose a relevance model to calculate the relevance of each image, which is a probabilistic model that evaluates the relevance of the HTML document linking to the image. However, all of these re-ranking methods have drawbacks. For clustering based re-ranking methods, the number of clusters is hard to determine and it is quite doubtful that the constructed image clusters will be meaningful in all cases. For classification based approaches, positive training data is hard to obtain in the scenario of query by keyword. And the relevance model [5] depends on the documents returned by a text web search engine, which may be irrelevant with the retrieved images.

In this paper, we propose a novel pseudo relevance feedback method named iterative probabilistic one-class SVMs (IPOCS) to re-rank the retrieved images. Based on the assumption that most top-ranked images are relevant, given the images retrieved by a search engine, we iteratively train one-class SVMs (OCS) [9] for each image feature, convert their outputs to probabilities so as to combine the decision of various OCS, and re-rank the retrieved images according to their probabilities of being relevant. Systematic experiments demonstrate the effectiveness of our method, even if the assumption is not well satisfied.

The rest of the paper is organized as follows. In Sect.2, we present IPOCS in detail. To evaluate the proposed re-ranking method, we compare its performance with that of a recently developed manifold ranking algorithm [14] by systematic experiments in Sect.3. Finally, we conclude the paper in Sect.4.

2 Iterative Probabilistic One-Class SVMs

To better explain the method of IPOCS, in this section, we will first introduce OCS, followed by some discussion of its application in IPOCS; then we will discuss probabilistic outputs for OCS in order to combine the decision from different kinds of image features; finally, we will give the flowchart of the algorithm.

2.1 One-Class SVMs

The underlying principle of OCS is to estimate the minimum volume in the feature space that contains a constant fraction of the total probability of data distribution while keeping a large margin [8]. Among the several interpretations

of OCS [8][9], we present the most straight-forward one here. Consider training data $x_1, \dots, x_l \in X$, where $l \in \mathbb{N}$ is the number of observations, and X is the input space. Let Φ be a mapping $X \rightarrow F$ such that the dot product in the feature space F can be easily calculated by some kernel function: $k(x, y) = (\Phi(x) \cdot \Phi(y))$.

The basic idea of OCS is to find the smallest hyper-sphere to enclose most of the training data in the feature space F , which can be expressed as the following quadratic program:

$$\min_{R \in \mathbb{R}, \xi \in \mathbb{R}^l, c \in F} R^2 + \left(\sum_i \xi_i \right) / (vl) \quad (1)$$

$$\text{subject to } \|\Phi(x_i) - c\|^2 \leq R^2 + \xi_i, \xi_i \geq 0 \text{ for } i \in \{1, \dots, l\}$$

and the dual form:

$$\min_a \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) - \sum_i \alpha_i k(x_i, x_i) \quad (2)$$

$$\text{subject to } 0 \leq \alpha_i \leq 1/(vl), \sum_i \alpha_i = 1$$

with $c = \sum_i \Phi(x_i)$. The decision function takes the form:

$$f(x) = \text{sgn}\left(R^2 - \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) + 2 \sum_i \alpha_i k(x_i, x) - k(x, x)\right) \quad (3)$$

The parameter $v \in [0, 1]$ controls the tradeoff between the radius of the hyper-sphere and the number of observations that is enclosed in it. Furthermore, it has been proved that the following two statements hold [9]:

1. v is at least as large as the fraction of outliers;
2. v is no larger than the fraction of SVs.

Therefore, a large v ensures a large fraction of SVs and probably a small hyper-sphere; while a small v guarantees a small number of outliers and thus many observations within the hyper-sphere.

OCS has already been applied in content-based image retrieval [1], and the training data is provided by positive relevance feedback. In the context of web image retrieval, where no feedback information is available, we make a reasonable assumption that most top-ranked images are relevant, and use them as the training data. To remedy the possible errors incurred by this assumption, we design the following algorithm: given a certain kind of image representation, we first keep the top images for re-ranking. Then the first n ($n < m$) images are selected as pseudo positive examples, on which OCS is trained. Finally we re-rank the m images according to the outputs of OCS. The above operations are iterated until a certain criterion is satisfied.

The algorithm is justified as follows: the selection of the top m images for re-ranking is to ensure enough relevant images are included, and thus a high recall. On the other hand, the reason why only the first n images are used for training

OCS is that if many images are included in the training process, the presumed assumption might be violated, and the performance of OCS will deteriorate. The purpose for iterating between training and re-ranking is to progressively accumulate the relevant images in the head of the ranking list and the irrelevant ones in the tail.

In the training process of OCS, we take a large v . Thus many training observations will be support vectors, and their information will be fully utilized. The stopping criterion can be that the ranking order does not change after two consecutive re-ranking processes, or that the maximum iteration number is reached. In our implementation, the second criterion is taken to ensure the processing time is under control.

2.2 Probabilistic Outputs for OCS

In IPOCS, we construct OCS for each kind of image representation, the outputs of which must be combined for the overall decision. However, like two-class SVMs, OCS outputs uncalibrated values, thus we need to convert them to probabilities for the purpose of combination.

Inspired by the work presented in [7], after OCS is obtained, we train the parameters of an additional sigmoid function to map the outputs to probabilities. However, we use unlabeled data to train the sigmoid function, while in [7], the training data belongs to two classes. In the original algorithm [7], after SVMs is trained on observations of two classes, the posterior is modeled as follows:

$$P(y = 1|f) = 1/(1 + \exp(Af + B)) \quad (4)$$

where $f = f(x)$ is the uncalibrated output of SVMs for the observation x , $y \in \{-1, 1\}$ is the class label, and $P(y = 1|f)$ is the posterior probability that is a positive example given the output of SVMs. The parameters A and B are adapted to give the best probability outputs. As long as $A < 0$, (4) is monotonic, and large SVMs outputs correspond to large posterior probabilities.

The determination of A and B is based on maximum likelihood estimation, which can be transformed into the following optimization problem. Given a training set $\{(f_i, y_i)\}$, where f_i and y_i are the SVMs output and the class label of x_i , define $t_i = (y_i + 1)/2$ as the probability of x_i being a positive example. Note that since $y_i \in \{-1, 1\}$, $t_i \in \{0, 1\}$. Thus the optimization problem:

$$\min\left\{-\sum_i (t_i \log(p_i) + (1 - t_i) \log(1 - p_i))\right\} \quad (5)$$

where $p_i = 1/(1 + \exp(Af_i + B))$

To fit the sigmoid function, the algorithm in [7] uses a simple out-of-sample model: each observation in the training set is assigned with a small probability of opposite label in the out-of-sample data, i.e., $t_i \in [0, 1]$ instead of $t_i \in \{0, 1\}$. In our algorithm, this scheme is generalized to deal with unlabeled data. To speak

concretely, we establish a model for estimating the probability that an unlabeled image is a relevant one based on the original ranking result, i.e.

$$t_i = g(r_{x_i}) \quad (6)$$

where r_{x_i} is the ranking order of x_i , and $g(\cdot)$ is a function that maps each retrieved image to a real number in $[0, 1]$. In other words, the probability of an image being relevant is determined by its position in the original ranking list.

Generally speaking, $g(\cdot)$ should be a decreasing function, i.e., top-ranked images should have a large t_i , while bottom-ranked images should have a small one. In our current implementation, we take a simple form of $g(\cdot)$:

$$t_i = g(r_{x_i}) = 1/r_{x_i}^\beta \quad (7)$$

where β is a positive parameter that controls that decreasing rate of t_i as r_{x_i} increases. Presently, we set it to 1 for simplicity.

Once we obtain the posterior probabilities from the outputs of OCS constructed using each kind of image representation, we must combine the results to give the overall decision, in which several schemes may be taken. For example, we may average the probabilities, or select the largest one instead. In the context of web image retrieval, we prefer the second scheme to the first one, since we want to preserve the most confident decision of each OCS. Suppose that there are T kinds of image representation, let p^j denote the probabilistic output of OCS constructed using the j th image representation. The combination scheme can be expressed as follows:

$$p = \max\{p^1, \dots, p^T\} \quad (8)$$

2.3 The Flowchart of IPOCS

Based on the above discussion, we summarize the algorithm of IPOCS in Fig.1.

1. Perform similarity ranking, keeping the first m images for re-ranking;
2. Iterate for N_p times:
 - (a) Select the first n images as pseudo positive examples;
 - (b) for each kind of image representation:
 - Train OCS based on the pseudo positive examples;
 - Fit a sigmoid function to get the probabilistic outputs.
 - (c) Combine the outputs to get the overall probabilities, using (8);
 - (d) Re-rank the m retrieved images according to p .

Fig. 1. The flowchart of IPOCS

3 Experimental Results

In this section, we perform experiments to evaluate the performance of IPOCS, and compare it with manifold ranking (MR) [14], a recently developed algorithm for ranking data points which considers their global structure instead of pair-wise distances. We design two kinds of experiments in our evaluation: one is based on Corel images, and the other is based on the images returned by a prototype web image search engine *iFind* [13].

For both IPOCS and MR, we use the top $m = 100$ images for re-ranking, which are represented using color histogram [10] and wavelet features [11]. In IPOCS, we use these two kinds of image features to get two sets of probabilities, and fuse them to get the overall decision. The adopted kernel function in OCS is the RBF kernel, i.e., $k(x_i, x_j) = \exp[-\|x_i - x_j\|^2 / (2\sigma_p^2)]$. Thus there are four parameters needed to be set: n , v , σ_p and N_p . We conservatively set $n = 10$ to ensure that OCS will not be misled by many irrelevant images. Based on the discussion in subsection 3.1, we experimentally set $v = 0.99$, which achieves the best result among all the choices. The value of σ_p is empirically set to be 0.1. And N_p is set to 20 as a tradeoff between processing time and performance. In MR, the first n images are initially assigned with score 1, and the three parameters are set as follows: $\alpha = 0.99$, which is consistent with the experiments performed in [14]; $\sigma_m = \sigma_p$; and the iteration number N_m is set to 50, since we observe no improvement in performance with more iterations.

3.1 Experiments with Corel Images

We first form a general-purpose image database from which the initial retrieved images are to be selected. The database consists of 5,000 Corel images, which are made up of 50 image categories, each having 100 images of essentially the same topic. To simulate the top m images retrieved by a web image search engine, we first designate a certain category to contain all the relevant images, fix the ratio ra_m of relevant images in the m images, and randomly select images from the database according to ra_m . Then we vary the ratio ra_n of relevant images in the first n images to compare the two methods in different circumstances.

The adopted performance measure is precision. In this experiment, each of the categories is taken as the target, and the precision is averaged over all categories. The comparison results are illustrated in Fig.2.

From Fig.2, we can see that IPOCS can always significantly improve the retrieval result and outperform MR, no matter what value is taken for ra_m and ra_n . For example, when $ra_m = ra_n = 0.5$, P10 (precision within the top 10 images) is 82.5% using IPOCS, which improves the original precision (0.5) by 65.0%. While P10 is only 60.7% using MR, which improves the original precision by 21.4%. When $ra_m = 0.2$ and $ra_n = 0.5$, P10 is 76.9% using IPOCS, which improves the original precision by 53.8%. While P10 is 45.2% using MR, which even brings degradation to the original result. Note that when ra_n is small, the presumed assumption is not well satisfied, while IPOCS still significantly improves the ranking result.

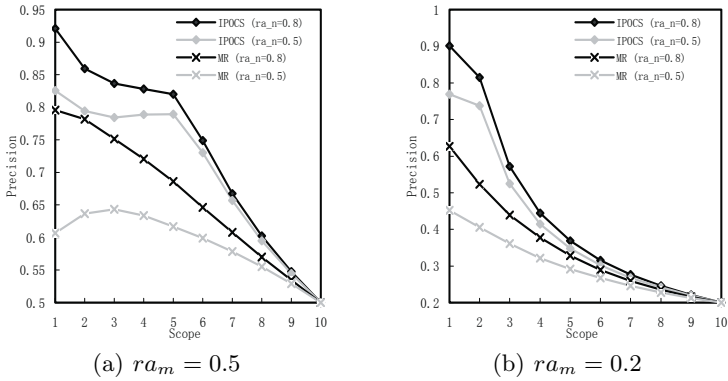


Fig. 2. Comparison of re-ranking results

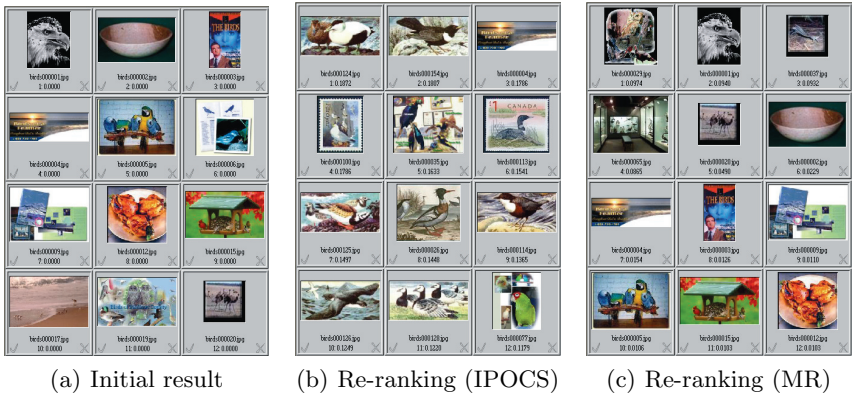


Fig. 3. Ranked images for the query "bird"

3.2 Experiments with Retrieved Images from *iFind*

iFind is a prototype web image search engine developed at Microsoft Research Asia. It is able to search 12 million web images based on text description of an image obtained from the web page where it is located. In this experiment, we resort to *iFind* to produce the initial retrieval result, given the query keywords, and make use of both IPOCS and MR to re-rank the images. In Fig.3, we compare the top 12 images using the two algorithms given the query keyword "bird".

Obviously, the initial retrieval result Fig.3(a) produced by similarity ranking is far from satisfactory, since there are many irrelevant images due to inappropriate text description. Comparing both Fig.3(b) and 3(c) with the initial result, we can see that IPOCS greatly improves the performance, with the top 12 images all closely related to the query; while the improvement using MR is hard to tell.

4 Conclusion

In this paper, we have proposed a novel pseudo relevance feedback method based on IPOCS, which is used to re-rank images retrieved by a web image search engine. In the context where feedback information is not available, we make a reasonable assumption that most of the top-ranked images are relevant, and train OCS using these pseudo positive examples. To remedy the possible errors incurred by this assumption, we design an iterative algorithm to progressively refine the ranking result; furthermore, to combine the decision from different kinds of image representation, we train an additional sigmoid function which maps the outputs of OCS to probabilities. Experimental results demonstrate the effectiveness of the proposed method.

Acknowledgements. This work was supported by National High Technology Research and Development Program of China (863 Program) under contract No.2001AA114190.

References

- [1] Chen, Y., et al: One-class SVM for learning in image retrieval. Proc. ICIP (1999) 440-447
- [2] Hearst, M.A., et al: Reexamining the cluster hypothesis: Scatter/gather on retrieval results. Proc. SIGIR (1996) 76-84
- [3] Lee, K., et al: Document re-ranking model using clusters. KORTERM-TR-99-03 (1999)
- [4] Leuski, A.: Evaluating document clustering for interactive information retrieval. Proc. CIKM (2001)
- [5] Lin, W., et al: Web image retrieval re-ranking with relevance model. Proc. IEEE/WIC Int. Conf. on Web Intelligence (2003) 242-248
- [6] Park, G., et al: A ranking algorithm using dynamic clustering for content-based image retrieval. Proc. CIVR (2002) 328-337
- [7] Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, MIT Press (1999)
- [8] Ratsch, G., et al: Constructing boosting algorithm from SVMs: An application to one-class classification. *IEEE Trans. on PAMI* **24** (2002) 1184-1199
- [9] Scholkopf, B., et al: Estimating the support of a high-dimensional distribution. *Neural computation* **13** (2001) 1443-1471
- [10] Swain, M., et al: Color indexing. *Int. Journal of Computer Vision* **7** (1991) 11-32
- [11] Wang, J.Z., et al: Content-based image indexing and searching using Daubechies' wavelets. *Int. Journal of Digital Libraries*, **1** (1998) 311-328
- [12] Yan, R., et al: Multimedia search with pseudo-relevance feedback. Proc. Int. Conf. on Image and Video Retrieval (2003) 238-247
- [13] Zheng, C., et al: iFind: A web image search engine. SIGIR Demo (2001)
- [14] Zhou, D., et al: Ranking on data manifolds. 18th Annual Conf. on Neural Information Processing System (2003)