BOOSTING WEB IMAGE SEARCH BY CO-RANKING*

Jingrui He[1], Changshui Zhang[2], Nanyuan Zhao[2], Hanghang Tong[3]

[1,2]Department of Automation, Tsinghua University, Beijing 100084, China
[1]{hejingrui98, walkstar98}@mails.tsinghua.edu.cn, [2]{zcs, zhaony}@tsinghua.edu.cn

ABSTRACT

To maximally improve the precision among top-ranked images returned by a web image search engine without putting extra burden on the user, we propose in this paper a novel co-ranking framework which will re-rank the retrieved images to move the irrelevant ones to the tail of the list. The characteristic of the proposed framework can be summarized as follows: (1) making use of the decisions from multi-view of images to boost retrieval performance; (2) generalizing present multi-view algorithms which need labeled data for initialization to the unsupervised case so that no extra interaction is required. To implement the framework, we use one-class support vector machines to train the basic learner, and propose different schemes for combination. Experimental results demonstrate the effectiveness of the proposed framework.

1. INTRODUCTION

To browse through the huge image resource available on the World Wide Web both effectively and efficiently, people have designed several web image search engines, such as Google Image Search, AltaVista Image Search, AllTheWeb Picture Search, Lycos Multimedia Search, etc. Generally speaking, these search engines are all text-based, i.e., the images are described by filename, caption, surrounding text, and text in the HTML document that displays the images, etc. Compared with content-based image search engines, their performance in terms of precision is relatively satisfactory. However, it is often observed that some top-ranked images are actually irrelevant to the user's query concept. This problem may be attributed to the following reasons: (1) the multiple meanings of words or phrases used to characterize the content of an image; (2) misplacement of images in a totally irrelevant environment; etc. The removal of these top-ranked irrelevant images will further boost the performance of web image search engines, and is highly desirable from the users' perspective.

The above problem can be reformulated as follows: given a list of images retrieved by a web image search engine, how to re-rank them in order to move the irrelevant ones to the tail of the list, and further improve the precision among top-ranked images accordingly. In this paper, we take on this problem as a multi-view problem, and propose a novel co-ranking framework which is based on the fact that low level image features can be partitioned into disjoint subsets (views) which roughly satisfy the assumptions of compatibility and un-correlation. On the other hand, since in real applications, the user might be reluctant to provide relevance feedback, the framework is implemented in an unsupervised manner without extra interaction with the user, in contrast to present multi-view learning algorithms [1, 3, 7], which make use of labeled data for initialization.

Furthermore, we adopt One-Class SVMs (OCS) [2] to train the basic learner in each view since it is expected to perform well in a finite sample size setting [2]. To combine the decisions from different learners, the outputs of OCS are converted to probabilities using the method in [5]. Finally, we propose different combination schemes which are compared via experimental results.

The rest of the paper is organized as follows. In Sect.2, we briefly review related work in web image re-ranking and multi-view learning. The proposed co-ranking framework is presented in Sect.3, with the implementation issues discussed in Sect.4. Experimental results are provided in Sect.5, which demonstrate the effectiveness of the framework from various aspects. Finally, we conclude the paper in Sect.6.

2. RELATED WORK

To solve the problem of web image re-ranking, researchers have proposed different methods. For example, Yan et al [9] train SVMs whose positive training data are from the query examples, while negative training data are from negative pseudo relevance feedback; however, in the scenario of query by keyword, positive training data is hard to obtain. And Lin et al [10] propose a relevance model to calculate the relevance of each

* This work is supported by the project (60475001) of the National Natural Science Foundation of China.

image, which evaluates the relevance of the HTML document linking to the image; however, this model depends on the documents returned by a text web search engine, which may be totally irrelevant with the retrieved images.

In a multi-view problem, the features of the domain can be partitioned into disjoint subsets (views) that are sufficient to learn the target concept [3]. Several algorithms have been proposed to deal with this problem. For example, based on the assumptions of compatibility and un-correlation, Blum et al [1] propose co-training which will gradually add self-labeled examples to the training set; Nigam et al [7] propose co-EM which differs from co-training in that unlabeled data obtain probabilistic labels instead of absolute ones and these labels are updated in each iteration; furthermore, Muslea et al [3] propose co-EMT, which combines semi-supervised and active learning. It is worth noticing that all of the above algorithms need a labeled set to train the initial basic classifiers. However, in the context of web image retrieval, the labeled set must be provided by the user, which will inevitably put extra burden on the user.

3. THE PROPOSED CO-RANKING FRAMEWORK

Suppose that in the problem domain, we have two views $V1$ and $V2$, thus any example x can be described as $[x_1, x_2]$, where x_1 and x_1 belong to the two views respectively. Thus our co-ranking framework comes in parallel with co-training, and is summarized in Fig.1.

In essence, co-ranking is proposed to deal with the kind of problems that given a rough ranking of examples, how to boost the ranking result such that the examples are sorted in descending order of a certain criterion. Different from co-training, co-EM, and co-EMT, the framework is implemented in an unsupervised manner. To be specific, firstly, the input examples in D do not have labels, and they only have an initial ranking order; secondly, L is an unsupervised learning algorithm. Furthermore, in each iteration, the examples in D will be re-ranked according to the present combined learner l, which is different from co-training, co-EM, and co-EMT. Although presently we only adopt two views, the co-ranking framework can be easily extended to the situation of more than two views.

Like in the co-training algorithm, two assumptions should be approximately satisfied for the co-ranking algorithm to perform well: (1) the two views should be compatible, i.e., the optimal ranking result can be obtained from either of the two views; (2) the two views should be independent in order to refine the ranking result from different perspectives. The analysis on the convergence of our co-ranking framework based on the above assumptions should be analogous to that of co-training, and we leave the rigorous analysis to future work.

The co-ranking framework is well suited for the problem of web image re-ranking. Firstly, the ranked set D of unlabeled examples is provided by a web image search engine based on text description. Secondly, the criterion is the relevance between each image and the query concept. Thirdly, by making use of low level image features to form the views, the above assumptions can be considerably satisfied, which guarantees a good performance from theoretical perspective. To be specific, present low level features can be categorized into color, texture, shape, etc. It is reasonable to assume that features belonging to different categories are independent, since they describe image contents from different perspectives. On the other hand, although features from different categories may not be totally compatible, we can still assume this assumption since the relevant images returned by a web image search engine are generally similar enough to be identified by features from any of the categories, and the number of irrelevant ones is often small and can be considered as random noise.

- **Inputs:**
  - a learning problem with two views $V1$ and $V2$
  - an unsupervised learning algorithm L
  - the ranked set D of m unlabeled examples
  - the number k of iterations to be performed
- **Loop** for k iterations
  - use L and $V1(D)$ to create a basic learner l_1 that will assign a ranking score to each example x based on present ranking order
  - use L and $V2(D)$ to create a basic learner l_2 that will assign a ranking score to each example x based on present ranking order
  - combine the ranking scores of l_1 and l_2 to obtain the combined learner l
  - re-rank the examples in D according to l
- **Outputs:**
  - the combined learner l
  - a ranked list of the examples in D

Fig.1. Co-ranking framework

4. IMPLEMENTATION ISSUES

4.1. The Design of Different View

In our current implementation, we use color histogram feature [8] which belongs to the color category, and wavelet feature [6] which belongs to the texture category to form $V1$ and $V2$. Although the two views are constructed using low-level features, the co-ranking

method can still improve the retrieval performance, as is shown in the next section. One can certainly choose other low-level features and even textual description to form the views. However, the selection of the optimal views is beyond the scope of this paper, and we will further extend our work in this direction.

4.2. One-Class SVMs as the Basic Learner

A key issue of the co-ranking framework is the design of the unsupervised learning algorithm L. We adopt One-Class SVMs (OCS) [2] to train basic learners in each view, since it is expected to perform well in a finite sample size setting. When training data are corrupted by noise, the performance of OCS will greatly degrade. Given the present ranking order of the images in D, precision among top-ranked images are relatively higher than that among bottom-ranked ones. Therefore, we use n ($n < m$) top-ranked images as the training data for OCS, and the obtained basic learner will output ranking scores for all the images in D.

In the co-ranking framework, the outputs of weak learners must be integrated to get the combined learner l. However, OCS outputs uncalibrated values, and we need to convert them to probabilities. Therefore we adopt the method proposed in [5], which is different from [4] since the former uses unlabeled data while the latter uses labeled data. To be specific, we train the parameters of a sigmoid function to map the outputs to probabilities:

$$P(y=1|f) = 1/\left[1+\exp(Af+B)\right] \quad \textbf{(1)}$$

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 x is a positive example given the output of SVMs. The determination of A and B is based on the following optimization problem.

$$\begin{aligned} &\min\left\{-\sum_i \left(t_i \log(p_i) + (1-t_i)\log(1-p_i)\right)\right\} \\ &\text{where } p_i = 1/\left[1+\exp(Af_i+B)\right] \end{aligned} \quad \textbf{(2)}$$

where f_i is the SVMs output of the ith observation x_i, and t_i is the probability of x_i being a positive example, which is obtained via (3).

$$t_i = 1/r_{x_i}^{\beta} \quad \textbf{(3)}$$

where r_{x_i} is the ranking order of x_i, and β is a positive parameter that controls the decreasing rate of t_i as r_{x_i} increases. Presently, we set it to 1 for simplicity. Therefore, top-ranked images have a large t_i, while bottom-ranked images have a small one, which is consistent with our intuition.

4.3. Combination Scheme

To combine the posterior probabilities obtained from different views, we propose two schemes: (1) to average the probabilities, (2) to select the largest one. Let p^1 and p^2 denote the probabilistic outputs for $V1$ and $V2$, the combined learner l using the two schemes can be expressed as (4) and (5) respectively:

$$l(x) = \left[p^1(x) + p^2(x)\right]/2 \quad \textbf{(4)}$$

$$l(x) = \max\left\{p^1(x), p^2(x)\right\} \quad \textbf{(5)}$$

5. EXPERIMENTAL RESULTS

5.1. Parameter and Operation Setting

To test the performance of the proposed co-ranking framework in different circumstances, we first form a general-purpose image database from which the initial retrieved images are to be simulated. 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 dataset D of m ranked 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, which will be fed into OCS to train the basic learner. In all our experiments, the adopted performance measure is precision. Each of the categories is taken as the target, and the precision is averaged over all categories.

The parameter settings of the co-ranking framework are as follows. $k = 20$ iterations are performed as a tradeoff between processing time and performance. The dataset D consists of $m = 100$ images, and the first $n = 10$ images are used for training OCS. The adopted kernel function in OCS is the RBF kernel, i.e., $k(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / (2\sigma_p^2)\right)$. The value of σ_p is empirically set to be 0.1, which achieves the best result among all the choices.

5.2. Comparison of Combination Strategies

As is mentioned in section 4.3, two schemes are available for obtaining the combined learner l ((4) and (5)). In this subsection, we perform experiments to compare their performance. The results are listed in Fig.2.

From Fig.2, we can see that the first strategy, which averages the probabilities, performs better than the second one, which selects the larger probability as the final result. For example, when $ra_n = 0.5$, P10 is 93% using the first

scheme, and is 83% using the second one. Based on the experimental results, we will apply the average strategy in subsequent experiments.

5.3. Comparison with Single View Algorithm

One characteristic of the co-ranking framework is that it partitions image features into two subsets (views), and the two views will take the advantage of each other to iteratively improve the ranking result. One may naturally come up with the questions that will this partition be of any good? Will the performance of an algorithm that runs without feature partition be even better? To answer these questions, we design another algorithm named iterative One-Class SVMs (IOCS) for comparison: it differs from the co-ranking method in that in each iteration, only one OCS will be trained on all the features. Comparison results are presented in Fig.3.

Experimental results demonstrate that our co-ranking algorithm, which partitions the features into two views, outperforms its counterpart where only a single view is assumed. For example, when $ra_n = 0.8$, P10 using the co-ranking method is 97.4%, and is 80.1% using IOCS.

5.4. Experiments with Google Retrieved Images

We have also performed experiments with Google retrieved images. Given the query keyword "building", we first resort to Google to form the initial ranked set D, and the first ten images are shown in Fig.4(a). Obviously, the fourth image is totally irrelevant with the query, and it is retrieved due to improper text description. Then we apply the co-ranking framework to D, and the first ten re-ranked images are shown in Fig.4(b). From the result, we can see that all of the top-ranked images are closely related to the query concept, thus the performance of the web image search engine is improved.

6. CONCLUSION

In this paper, we propose a novel co-ranking framework to deal with the problem that some top-ranked images returned by a web image search engine are actually irrelevant to the user's query concept. By partitioning the image features into disjoint subsets (views), the framework can iteratively boost the ranking result, using the basic learners trained in each view. One major difference between this framework and other multi-view algorithms is that we do not need labeled data for initialization, while existing algorithms all depend on labeled data to construct basic learners. In our current implementation, we choose OCS as the learning algorithm and design different schemes for the sake of combination. The effectiveness of the proposed framework is validated by systematic experiments. Future work includes: 1) rigorous analysis on the convergence of the proposed framework; 2) investigate the optimal views for web image retrieval.

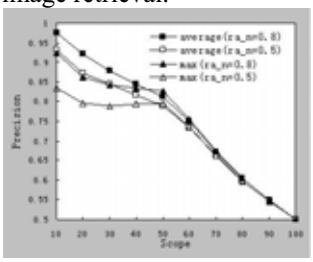


Fig.2. $ra_m = 0.5$

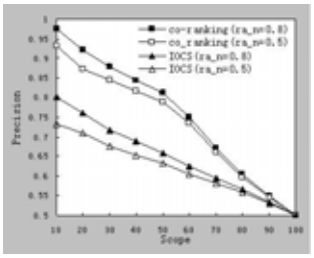
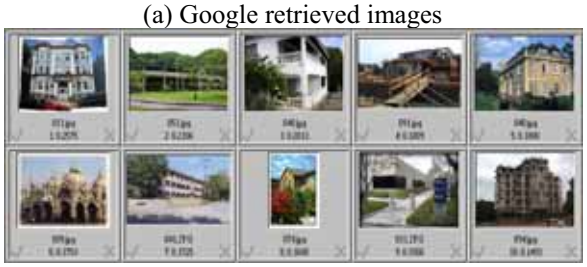


Fig.3. $ra_m = 0.5$

(a) Google retrieved images



(b) Re-ranked images

Fig.4. Experiments with Google retrieved images

7. REFERENCES

[1] A. Blum, et al, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. COLT*, pp. 92-100, 1998.

[2] B. Scholkopf, et al, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, pp. 1443-1471, 2001.

[3] I. Muslea, et al, "Active + Semi-Supervised Learning = Robust Multi-View Learning," *Proc. ICML*, pp. 435-442, 2002.

[4] J.C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," In *Advances in Large Margin Classifiers*, MIT Press, 1999.

[5] J.R. He, et al, "Probabilistic One-Class SVMs in Web Image Retrieval," *Proc. PCM*, 2004.

[6] J.Z. Wang, et al, "Content-Based Image Indexing and Searching Using Daubechies' Wavelets," *Int. Journal of Digital Libraries*, vol. 1, pp. 311-328, 1998.

[7] K. Nigam, et al, "Analyzing the Effectiveness and Applicability of Co-Training," *Proc. CIKM*, pp. 86-93, 2000.

[8] M. Swain, et al, "Color Indexing," *IJCV*, pp. 11-32, 1991.

[9] R. Yan, et al, "Multimedia Search with Pseudo-Relevance Feedback," *Int. Conf. on Image and Video l*, pp. 238-247, 2003.

[10] W. Lin, et al, "Web Image Retrieval Re-Ranking with Relevance Model," *Proc. IEEE/WIC Int. Conf. on Web Intelligence*, pp. 242-248, 2003.