

Meta-tag Propagation by Co-training an Ensemble Classifier for Improving Image Search Relevance

[‡]Aayush Sharma* [‡]Gang Hua, ^bZicheng Liu and ^bZhengyou Zhang
[‡]India Institute of Technology [‡]Microsoft Live Labs Research & ^bMicrosoft Research
Roorkee, India One Microsoft Way, Redmond, WA
s.aayush@gmail.com {ganhua, zliu, zhang}@microsoft.com

Abstract

The ever-increasing gigantic amount of images over the web necessitates automatic schemes for meta-tagging content descriptions such as object categories. These meta-tags are essential to text-based image search engines to improve their search relevance. Traditional supervised scheme is not suitable for this task because it needs too much manual labelling efforts and yet is hard to scale to a large number of object categories. Notice that in the search scenarios, the meta-tagging does not need to be perfect to help improve relevance because the available text tags and user click-through logs can partially rectify the inaccurate information. A weakly supervised scheme would be ideal when only sporadic labelled examples are exploited in spite of the expected loss in tagging accuracy. In this paper, we develop a novel weakly semi-supervised ensemble classifier trained based on a co-training framework for this tagging task. In essence the meta-tags are recursively propagated from the sparsely tagged examples to the un-tagged ones. Preliminary experiments on benchmark database such as Graz02 demonstrate the efficacy of the proposed approach.

1. Introduction

Since the start of the internet era, a gigantic amount (e.g., at the magnitude of billions) of images have been accumulated over the web. Although the research on content-based image retrieval [14] has been there for decades, it seems that we are still far from any practical systems which can be commercialized. Nevertheless, text-based image search systems have been successfully deployed by main stream search engines such as Google, Yahoo, as well as Microsoft.

All these commercial internet image search engines start by crawling and indexing the web images based on the *static rank* of the web-pages the images are associated with.

The surrounding texts of the web images are also extracted and stored during the crawling time. When a user types in a text query, the text information, along with some other additional information sources, will be used to calculate the relevance of the images with respect to the query the user typed in. This is called the *dynamic rank* of the images. The features used to calculate the dynamic rank, either from the surrounding texts or other information sources, are called *ranking features*.

To the best of our knowledge, the majority of the ranking features used by mainstream image search engines nowadays are from the surrounding texts and the image click-through information from the query logs. This largely ignores the relevance of the real image content to the queries the users typed in, and makes the current dynamic ranking system vulnerable to *web stuffing* attacks from malicious content providers such as adult sites. Web stuffing is a trick where malicious content providers intentionally and repeatedly embed texts highly related to one specific query to a web-page or the surroundings of web images, and thus fool the dynamic ranking system to falsely boost the ranking of these images. This will seriously hurt the search relevance when evaluated, for example, by the normalized discounted cumulative gain (NDCG) score [8].

It is obvious that automatic analysis of the image content and adding meta-tags such as the category information to describe the image content would be an effective way of countering such kind of web-stuffing attacks. Previous supervised image categorization methods are not suitable for this task because they need a lot of labelled examples, and thus are difficult to scale to a large number of object categories. When only sporadic labelled examples are used, a weakly supervised scheme would be ideal for this task although the tagging accuracy may not be as good as a fully supervised method. Notice that in the space of web images (at the magnitude of billions), scalability is the first priority. Moreover, we do not need to achieve perfect tagging results before we can help improve the search relevance, because

*Work performed as an intern at Microsoft Research, Redmond

surrounding texts and click-through logs would be able to counter some of the inaccurate meta-tagging information.

We explore the ability of an ensemble of decision trees induced from very weak supervision under a co-training framework [1], where new training examples with pseudo-labels for each decision tree are bootstrapped from all the other decision trees in the ensemble. The framework essentially behaves like that the meta-tag is propagated recursively from the tagged training examples to un-tagged new examples, which we name *meta-tag propagation*. Our preliminary evaluation on benchmark data-set such as Graz02 demonstrates the efficacy of the proposed method.

Related work are summarized in Sec. 2. The proposed method for co-training an ensemble of decision trees is presented in Sec. 3. Experimental results are reported and discussed in Sec. 4. Finally we conclude in Sec. 5.

2. Related work

Related work spans two different fields in machine learning and computer vision, and more specifically, weakly supervised or semi-supervised learning, and visual object categorization.

Semi-supervised learning exploits both labelled and unlabelled data for classification, clustering, and regression. In this paper, we focus on semi-supervised classification. Semi-supervised learning has been extensively studied (see [18] for a survey). Different semi-supervised learning methods employ different properties of the structure of the data to leverage the unlabelled examples. For example, semi-supervised EM [11] exploits the cluster structure of the data. While spectral-graph based method [17] relies on the general assumption that two examples which are close in the feature space should be in the same class.

A seminal work of semi-supervised learning is the co-training framework proposed by Blum and Mitchell [1]. Co-training assumes that the feature space can be naturally partitioned into two. Then two classifiers are trained on these two feature subsets. In the co-training process, each of the classifiers will expand its set of training examples by those unlabelled examples which are classified with high confidence by the other classifier. This process will be iterated until there is no more unlabelled data examples. For example, Levin et al [6] applied it to co-train two boosting classifiers for pedestrian detection from surveillance videos.

Zhou and Li [3] extend the co-training framework to a tri-training method where three classifiers are co-trained simultaneously. They used it for text classification. In this paper, we go one step further and exploit the co-training framework to train an ensemble classifier which can contain any number of classifiers (e.g., > 3). Besides we apply the proposed method to perform image categorization.

For any image categorization task, a good image representation is essential to achieve reasonable results for this

challenging task. We adopt the popular bag-of-feature representation [10, 13, 4, 16, 5] in our meta-tagging propagation task. To construct the bag-of-features, two steps are needed: first, sample local image patches and their descriptors are extracted; second, all these local descriptors are aggregated to be a set, which we call bag-of-features. Note that we do not quantize the local descriptors for simplicity and better understanding of the behavior of the co-trained ensemble classifier. Previous work has performed such kind of quantization [4, 16, 2] to obtain the so-called bag-of-visual-words model. It could easily be embedded in as an additional step in the proposed method, though.

There are various options in this two-step process. For example, different sampling strategies [12] can be employed in the first step, such as using random sampled patches, or using patches generated from any interest point detectors such as Harris, DOG [7] or Harris-Laplace [9]. Different local image descriptors can also be exploited such as color [10], texture, or more advanced SIFT descriptors [7].

Since any co-training schemes assume that different views of the data are used in the different co-trained classifiers, we leverage the different options presented above to achieve that. The details will be presented in the following section.

3. Co-training an ensemble of decision trees

Figure 1 is an overview of our algorithm which consists of an ensemble of six decision trees. We use D^L to denote the labelled example set and D^U to denote the unlabelled example set.

The basic idea of our algorithm is to start with a small set of labelled images, and gradually propagate the labels to the rest of the images which do not have labels. As more images have labels (predicted labels), we add these new images to the training set to improve the decision trees. The improved decision trees are used to propagate labels to more images, and the process continues.

In order to ensure the decision trees are sufficiently independent for each other, the trees are built with different local image descriptors and different sampling strategies. We use two different local image descriptors: SIFT descriptor and wavelet-based color descriptor, and three different sampling strategies: uniform sampling, SIFT feature detector, and center-focused sampling. In center-focused sampling, the likelihood of drawing a subwindow at a certain position depends on how close the position is to the image center. The intuition is that for the majority of images, the object of interest is usually located near the image center. There are in total six different combinations resulting in six decision trees.

In Figure 1, we use t to denote the iteration counter, and j to denote the index of the six decision trees. We use $D_{t,j}^L$ to

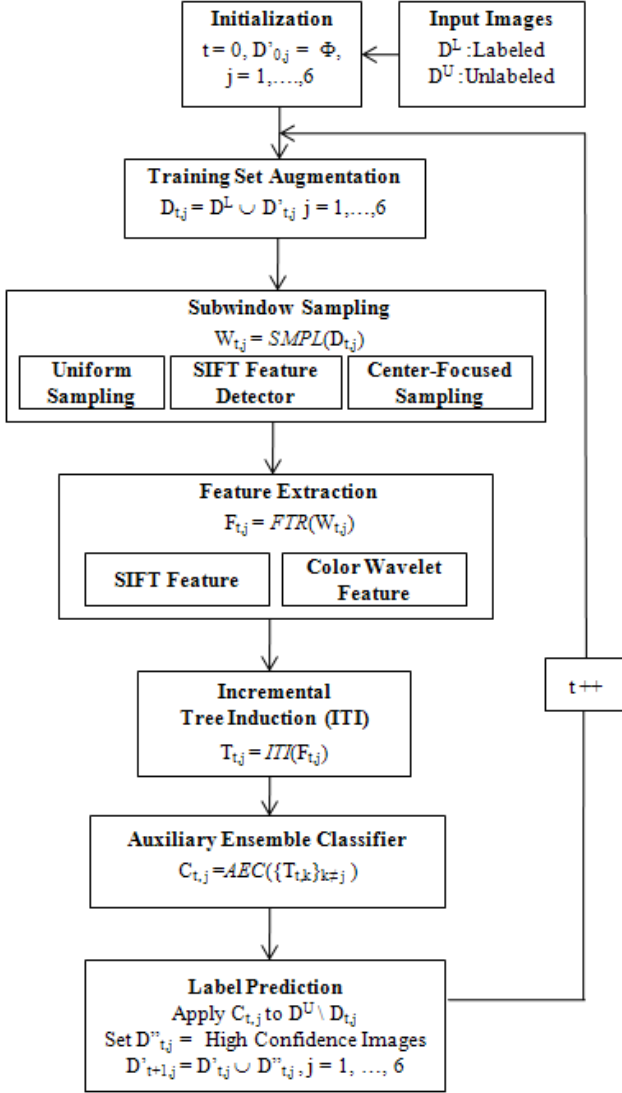


Figure 1. The decision tree ensemble co-training algorithm

denote the unlabelled images which have been augmented to the training set for tree j prior to t th iteration. The union of the original labelled image set D^L and $D^U_{t,j}$, which is denoted as $D_{t,j}$, is used to train the j th decision tree $T_{t,j}$. To avoid building a tree from scratch at each iteration, we use an incremental tree induction algorithm [15] which adjusts a decision tree locally based on newly added data.

For each tree $T_{t,j}$, $j = 1, \dots, 6$, we aggregate the other five trees $\{T_{t,k}\}_{k \neq j}$ to form an auxiliary ensemble classifier which is denoted as $C_{t,j}$. The reason we do not use $T_{t,j}$ itself is because we would like to avoid the potential problem of self-training. We use a method similar to what was used in [10] to construct an ensemble classifier from multiple decision trees. Basically, for each visual feature vector, we pass it through each tree and obtain the leaf node index. For each tree, the votes for each leaf node index are accumu-



Figure 2. Sample images of the three object categories in the Graz02 data set. Each column presents three sample images of each category.

lated into a histogram which are used for labelling the leaf node. For each tree and each image, the leaf node labels corresponding to all of its feature vectors are accumulated into a histogram which are used to classify the image. For each image, the results of the five individual tree classifiers are combined by majority voting to determine the final label of the image.

For each image in $D^U \setminus D_{t,j}$, we use $C_{t,j}$ to predict its label. For each class label l , to be conservative, we select two images from $D^U \setminus D_{t,j}$ whose confidences are the highest among all the images with predicted labels equal to l . The two selected images together with their predicted labels are added to the augmented image set $D^U_{t+1,j}$ which will be used at the next iteration to improve the j th tree $T_{t+1,j}$.

To evaluate the performance of our algorithm, we use a separate set of images, which is disjoint from D^L and D^U , to measure the classification performance of the decision tree ensemble at each iteration. The classification results provide an objective measurement on the quality of the propagated labels.

4. Experiments

We evaluate the efficacy of our approach by conducting experiments on publicly available Graz02 database. The database consists of three object categories - Bike (365 images), Car (420 images) and Person (311 images). Some sample images of each category are showing in Fig. 2.

The database is challenging in the sense of variable illumination, large background clutter, variable objects scales and perspectives, as well as occlusions. High variation with

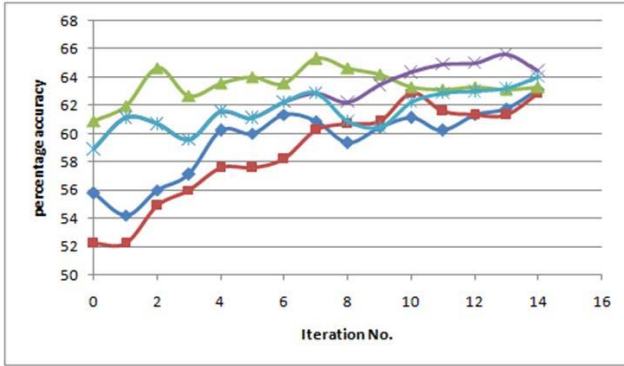


Figure 3. The incremental accuracy changes of 5 different runs of co-training the ensemble classifier. Each curve shows the accuracy of the ensemble classifier on the testing data set at each iteration of the co-training process.

respect to background makes it difficult to detect objects on the basis of context alone. For the purpose of experimentation, 900 images (300 per class) are randomly selected. These images are then randomly partitioned into two equal sized sets (150 per class) for training and testing purposes. The results are reported in the form of classification accuracy on the testing set.

For co-training, initial labelled set is randomly selected from the training set with 5 images per class. Rest of the training images form the unlabelled set which are used to augment the labelled training set as the co-training progresses. A total of 14 iterations of co-training are conducted, after which almost all the training examples will have been propagated with an label. During each iteration for each tree, 6 images (2 per class) with highest confidence of labelling from its auxiliary ensemble classifier are added to its training set. Each tree is then incrementally updated based on the newly added training examples.

4.1. Accuracy of the ensemble classifier

The incremental accuracy changes of the co-trained ensemble classifiers over 5 different runs are shown in Fig. 3. Note the classification accuracy reported in Fig. 3 is evaluated on the testing data-set. The performance on the unlabelled training data-set is indeed better than the performance value reported here. That is expected so we omit the results here to focus on demonstrating the generalization ability of the co-trained ensemble classifier on new data.

As we can clearly observe from Fig. 3, the initial performance of the ensemble classifier can indeed vary a lot based on the different initial training examples chosen. For example, the lowest initial accuracy in these 5 runs is nearly 52.0%, and the highest initial accuracy is 61%, there is an absolute accuracy gap of 9%. As we can clearly observe from Fig. 3, although the co-training process can not guarantee to absolutely increase the testing accuracy at each iter-

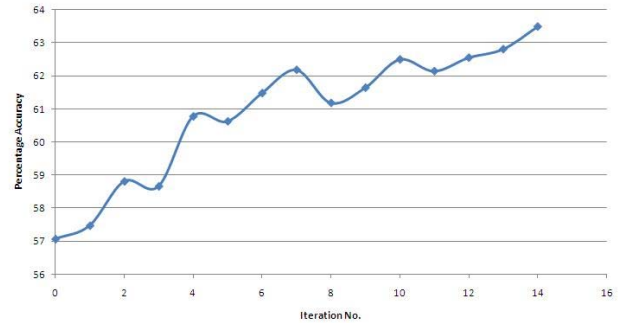


Figure 4. The average incremental accuracy change of the 5 different runs.

Accuracy before boosting: 62.89 %

	Total	Bikes	Cars	Persons
Bikes	150	99	36	15
Cars	150	9	137	4
Persons	150	52	51	47

Accuracy after boosting: 68.44 %

	Total	Bikes	Cars	Persons
Bikes	150	77	24	49
Cars	150	8	133	9
Persons	150	24	27	99

Table 1. The classification confusion matrix for another run of the co-training scheme with random initialization.

ation, overall it is able to improve the classification accuracy as the co-training progresses.

Fig. 4 presents the average incremental accuracy change over the 5 different runs. On average, we can obtain an absolute accuracy gain of 6.41% after 14 co-training iterations. Table 1 presents the confusion matrices of one run of the co-training of the ensemble classifiers. The upper part presents the confusion matrix from the initial training. As we can see, the recognition accuracy of person is more or less like random. After the co-training process, the recognition accuracy of person is significantly improved. It is at the expense of a small degradation of the recognition accuracies of the bike and car category, though.

In Fig. 5, we present some sample images which are classified incorrectly from each of the three categories. There are some information we may be able to infer by observing these mis-classified examples. For example, the objects of the target category inside most of these images are fairly small and the bag-of-feature representation is just overwhelmed by the background clutters. What is worse, in some of these misclassified images, only small portion of the target objects are shown in the scene. They are either occluded by some other objects, or are partially out of the im-



Figure 5. Some sample images which are erroneously classified after the co-training process. The first, second and third columns show the miss-classified images of person, car and bike, respectively. All the images in the first and second columns are mis-classified as bikes. The top two images in the third column are mis-classified as cars, and the last one is mis-classified as person.

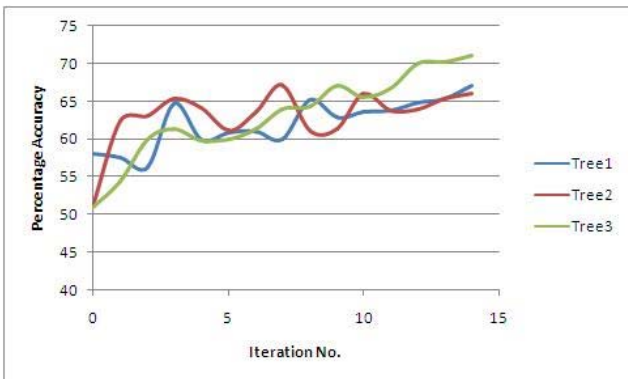


Figure 6. The progressive performance change of the three decision trees using SIFT features during one run of the co-training.

age view. Nevertheless, it may be just OK to mis-categorize these images, especially in the internet image search scenario. Users may not be interested in these images anyway because the objects they are interested in are not fully shown in these images.

4.2. Accuracy of individual trees

Fig. 6 and Fig. 7 present the progressive performance change of each of the individual decision trees during one run of the co-training process. In Fig. 6 the progressive performance changes of the three decision trees using the SIFT features are presented, while in Fig. 7 the progressive performance changes of the other three decision trees using the color Haar wavelet features are presented. Notice each of

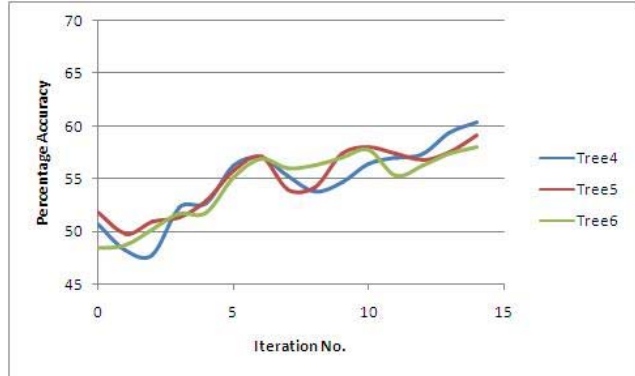


Figure 7. The progressive performance change of the three decision trees using color Haar wavelet features during one run of the co-training.



Figure 8. The testing accuracy of fully supervised trained ensemble classifier with different number of training examples.

these decision trees employed a different combination of the sampling strategies and the feature descriptors. As we can clearly observe, the performance of any of the individual decision trees in the ensemble is progressively improved in the co-training process without any exception. This demonstrates the efficacy of the co-training process from another view regarding the augmentation of the labelled training examples. What we also observe is that the decision trees utilizing the SIFT feature overall obtain better accuracy than those using the color wavelet features. This indicates that the SIFT features may be more powerful for the object categorization task, at least on the Graz02 data set.

4.3. Comparison with fully supervised training

One may wonder what would be the recognition performance if we perform fully supervised learning of the ensemble classifiers. Fig. 8 presents the recognition performance of a fully trained ensemble decision tree classifiers with different number of training examples. As we can clearly observe, the testing accuracy is almost the same after using 30 training examples per category (i.e., 90 in total). This may

on one hand reflect the difficulty of this data set. On the other hand, it also reflects the limited classification power of the decision tree classifier we employed to form the ensemble. In our future work, we will also investigate whether or not using other stronger classifier such as SVM or Boosting to form the ensemble could help improve the performance.

5. Conclusion and future work

We have presented a meta-tag propagation algorithm by co-training an ensemble of decision trees. To ensure the independence among the different decision trees in the ensemble, we use different combination of feature descriptors and sampling strategies for each of the individual decision trees. Our preliminary experiments show that we are able to use an extremely small set of labelled images to achieve the classification performance which cannot be obtained by a regular decision tree ensemble unless using a much larger set of labelled images.

The proposed approach targets at the task of meta-tagging image categories for mainstream text-based image search engines. This type of image meta-tagging will greatly help image search engines to mitigate malicious web-stuffing attacks and thus improve image search relevance. Our future work includes larger scale evaluation on images from mainstream image search engines and further investigate how much it can improve image search relevance.

References

- [1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of the Workshop on Computational Learning Theory*, pages 92–100, 1998. 2
- [2] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google’s image search. In *Proc. the Tenth IEEE International Conference on Computer Vision*, pages 1816–1823, 2005. 2
- [3] Z. hua Zhou and M. Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. on Knowledge and Data Engineering*, 17(11):1529–1541, November 2005. 2
- [4] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. the Tenth IEEE International Conference on Computer Vision*, pages 604–610, 2005. 2
- [5] S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proc. the Ninth IEEE International Conference on Computer Vision*, pages 649–655, 2003. 2
- [6] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *Proc. of the Ninth IEEE International Conference on Computer Vision*, pages 626–633, 2003. 2
- [7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2
- [8] I. Matveeva, C. Burges, T. Burkard, A. Laucius, and L. Wong. High accuracy retrieval with multiple nested ranker. In *Proc. the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 437–444, 2006. 1
- [9] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005. 2
- [10] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *Proc. of Advances in Neural Information Processing Systems 19*, pages 985–992, 1997. 2, 3
- [11] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2-3):103–134, 2000. 2
- [12] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proc. European Conference on Computer Vision*, pages 490–503, 2006. 2
- [13] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. the Ninth IEEE International Conference on Computer Vision*, pages 1470–1477, 2003. 2
- [14] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000. 1
- [15] P. E. Utgoff, N. C. Berkman, and J. A. Clouse. Decision tree induction based on efficient tree restructuring. *Machine Learning*, 29(1):5–44, 1997. 3
- [16] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proc. the Tenth IEEE International Conference on Computer Vision*, pages 1800–1807, 2005. 2
- [17] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proc. the 22nd international conference on Machine learning*, pages 1036–1043, New York, NY, USA, 2005. 2
- [18] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. 2