
Random Tag Forest

Weilong Yang

School of Computing Science
Simon Fraser University

Abstract

We consider the multi-label classification problem in this paper. We propose a randomized ensemble learning algorithm, *random tag forest*, which is an ensemble of random tag trees. Each tree is built by randomly defining a hierarchical tree structure over a subset of tag vocabulary. Each node in the tree corresponds to a tag in the vocabulary. During testing, a testing example will pass through each tree in the random tag forest. The tags along its path will be output as the prediction to this example. The final classification prediction is made by aggregating output across all trees in the random tag forest. The proposed approach is evaluated on a benchmark of image annotation. The experiments show that our approach achieves better results than the baseline.

1 Introduction

Multi-label classification is one of the most fundamental problems in machine learning. Different to *single-label* classification in which each example is only associated with a single label (e.g. 0 or 1 label for binary classification), each example in *multi-label* classification is associated with a set of labels. Multi-label classification has various applications in both data mining and computer vision. For example, one of its typical applications in computer vision is content-based image tagging [1]. We are given a training set $\mathcal{D} = \{(x^n, Y^n)\}_{n=1}^N$ in which each training image x is associated with a list of tags $Y = (y_1, \dots, y_l) \in \{0, 1\}^L$, where $y_i = 1$ indicates the i -th attribute (e.g. red, round, eatable, etc.) or object (e.g. tree, dog, grass, etc.) exists in the image. The task of multi-label learning is to learn a model from this training set so that we can automatically predict a list of tags for any given testing image. Compared with single label classification, it usually has a much larger label space which poses a challenge to the learning algorithm, though it may also provide more information regarding the labels, i.e. label correlation and label co-occurrence.

In the literature, a lot of progress has been made in multi-label classification. One of the most straightforward approach is to transform multi-label problem to a set of single-label problems by considering the learning of each label y_i as an independent learning task. However, this approach ignores the correlations among the labels. To overcome this shortage, label powerset [2] is proposed and it considers all the subsets of labels as a single-label classification task. Then, the original multi-label problem is transformed to a large number of single-label classification tasks. Since label powerset considers all the subsets, it is very inefficient to tackle the problem with a large label space. To improve the computing efficiency of the label powerset approach, Tsoumakas *et al.* [3] propose a random k -labelset approach by only considering the label subset of size k , and the subsets are sampled randomly. The philosophy behind those approaches is *problem transformation*, transforming the multi-label problem into a set of small single-label problems. Besides this line of work, conditional random field (CRF) [4] exploits the correlations among labels by adding a pairwise term in the potential function. The CRF model usually assumes the label space forms as a tree structure and it involves the inference on the tree structure, so it might be both memory and time consuming depending on the size of tree structure. In the literature of computer vision, there are several methods [1] [5] which are based on the simple k Nearest Neighbors (k NN) approach. A

Algorithm 1 Learning a Random Tag Tree

input : A set of training data $S = \{(x^n, Y^n)\}_{n=1}^N$ and a pre-selected label k

- Split S into positive and negative sets according to label k , i.e. $S_0^+ = \{(x^i : y_k^i = 1)\}$ $S_0^- = \{(x^i : y_k^i = 0)\}$
- Discard S_0^- and only work on S_0^+
- $i \leftarrow 0$

repeat

- if** S_i^+ meets the splitting criteria **then**
 - Randomly select a label j from S_i^+
 - Split S_i^+ according to label j and train a LR accordingly
- end if**
- if** $i \neq 0$ AND S_i^- meets the splitting criteria **then**
 - Randomly select a label l from S_i^-
 - Split S_i^- according to label l and train a LR accordingly
- end if**
- $i \leftarrow i + 1$

until Both S_i^+ , and S_i^- do not meet the splitting criteria

In our experiments, the number of positive examples for each tag varies a lot. Some tags have much more positive examples than others. This imbalance problem will become more severe on the rightmost subtree. Therefore, on the rightmost subtree, the tag with more positive samples are more likely to be selected than others. In our experiments, we have observed that in the learned random tag forest, there are a few tags which often appear on the right-most path of most of trees. This is undesirable since it would lead to more false positives for these tags. To avoid this problem, we simply discard the right child path of the root tag. In our experiment, this trick works well and improves the overall performance.

In the random tag forest, each tree is trained independently which makes it very easy to parallel the training algorithm. During testing, we will pass the example x through each of the random tag tree. Each tree will output a list of tags T which corresponds the nodes on the path of the testing example going through, as shown in Fig. 1, and a list of scores S . x will be passed to the left child if its decision score on current node $f_v(x)$ is bigger than the threshold λ , and we will also push this decision score to S . However, if the decision score is less than λ , we will then push a penalty score -0.5 into S , and the example will be passed to the right child. In the experiments, we show that λ controls the balance between precision and recall. After passing x through all the trees in the random forest, the output score is simply the average of the output from all the trees. The testing algorithm of a single random tag tree is illustrated in Algorithm 2.

3 Experiments

Dataset. We test our approach on a benchmark of image annotation, Corel5K [8]. It consists of about 5000 images. Following the same experimental setup in [1], we use 4500 images for training and the rest 499 images for testing. Each image has been labeled with 1 to 5 keywords. The total keyword vocabulary contains 260 words. However, in the training set, most of keywords have few positive samples. It is noted in [4] that discriminant approaches often suffer from the insufficient positive training samples. Therefore, in our experiment, we only consider the labels whose positive training samples is more or equal to 55. Then, the vocabulary is reduced to 71.

Features. In Corel5K dataset, we use image features provided by [1]. For each image, we use the following descriptors: color histograms in RGB, LAB, and HSV space respectively, GIST features [9], two SIFT features which are extracted densely and on the Harris-Laplacian interest points respectively. For the densely sampled SIFT feature, we also compute the histograms over three horizontal regions of the image, then these histograms are concatenated as a new feature. In total, we obtain seven features. We normalize each histogram-like feature and concatenate all the features together.

Algorithm 2 Prediction using a random tag tree

input : Test example x , a random tag tree \mathcal{T} , an empty array for the tag output $T = []$, an empty array used to save the score of the output tags $S = []$, a threshold set by user λ .

$v \leftarrow \text{root_node}$

if $f_v(x) < \lambda$ **then**

 Push v to T , push -0.5 to S

 Return T and S

else

 Push v to T , push $f_v(x)$ to S

 Traverse to left child, $v = \text{left_child}(v; \mathcal{T})$

repeat

if $f_v(x) > \lambda$ **then**

 push v to T , push $f_v(x)$ to S

 Traverse to left child, $v = \text{left_child}(v; \mathcal{T})$

else

 push v to T , push -0.5 to S

 Traverse to right child $v = \text{right_child}(v; \mathcal{T})$

end if

until v is a leaf node of \mathcal{T}

end if

Return T and S

	$\lambda = 0.1$		$\lambda = 0.2$		$\lambda = 0.3$		$\lambda = 0.4$	
	RTF	LR	RTF	LR	RTF	LR	RTF	LR
mean P	0.291	0.280	0.448	0.426	0.485	0.481	0.525	0.529
mean R	0.573	0.545	0.423	0.395	0.320	0.297	0.252	0.230
mean F1	0.386	0.370	0.435	0.409	0.385	0.367	0.340	0.320
N+	66	64	61	58	55	53	52	50
Avg. length	5.77	5.72	3.05	2.97	1.96	1.85	1.32	1.21

Table 1: Comparison of random tag forest (RTF) and the baseline (LR) on the Corel5K dataset. The best performance (in terms of mean F1 measure) is achieved by our approach at $\lambda = 0.2$. Note that we use Avg. length to denote the average length of predicted tag list for testing images.

Baseline. We compare our approach to the standard L2-regularized logistics regression (LR). For each keyword in the vocabulary, we train a binary LR. Similar to our approach, we use the Liblinear package for the LR implementation. We set the same value of parameter C in the LR as our approach $C = 100$ for all the keywords.

Measures. As previous works [4] [1], we use mean precision, mean recall, mean F1 measures over the keywords to measure the performance of our algorithm. The mean F1 measure is computed as $\text{mean F1} = \frac{2 \times \text{meanRecall} \times \text{meanPrecision}}{\text{meanRecall} + \text{meanPrecision}}$. We use N+ to denote the number of keywords with non-zero recall value.

Results. The results of our approach and baseline on the Corel5K dataset are shown in Table 1. We changed the value of the thresholding parameter λ from 0.2 to 0.4. The best performance in terms of the F1 measure is achieved by our approach at $\lambda = 0.2$. As aforementioned, parameter λ controls the balance between precision and recall. Along the increasing of λ , the average length of the predicted tag list is decreasing. Although mean precision is increasing with λ , the mean recall is decreasing. In Fig. 2, we plot the mean F1 result as a function of the number of trees in the ensemble. We can see that mean F1 is increasing with adding more trees to the ensemble.

4 Conclusion

We have presented random tag forest, ensembles of random tag tree, for dealing with multi-label classification problem. Although random tag forest achieves better results on the Corel5K dataset than the baseline, the improvement is not significant. This algorithm still suffers from the problem

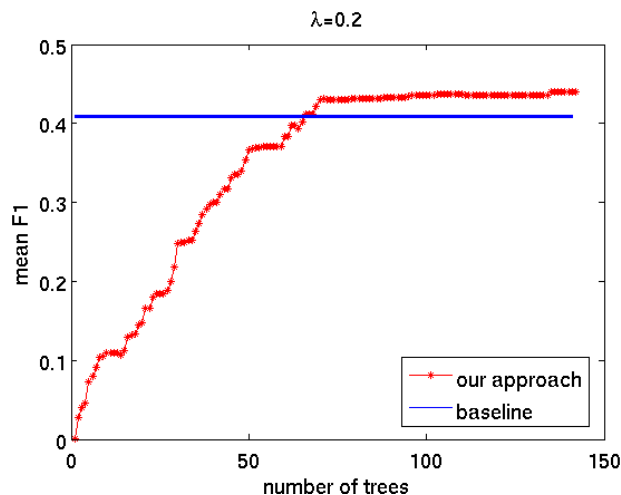


Figure 2: The plot of mean F1 results as a function of the number of trees in the ensemble.

that some tags have too few positive examples. Furthermore, it is still not very clear for us that how well the random tag tree can model the structure among the tags. In the random tag forest, we define the hierarchical tree structure over the tag vocabulary. However, in the Core15K dataset, the tags are usually the objects in the image, so that their structure is more like co-occurrence rather than hierarchy. So it might be more interesting to explore other types of structures in this randomized ensemble learning framework.

References

- [1] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, “Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation”, in *ICCV*. Ieee, 2009, pp. 309–316.
- [2] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown, “Learning multi-label scene classification”, *Pattern Recognition*, 2004.
- [3] G. Tsoumakas and I. Vlahavas, “Random k-labelsets: An ensemble method for multilabel classification”, in *ECML*, 2007.
- [4] Y. Xiang, X. Zhou, Z. Liu, T.S. Chua, and C.W. Ngo, “Semantic context modeling with maximal margin conditional random fields for automatic image annotation”, in *CVPR*. IEEE, 2010.
- [5] A. Makadia, V. Pavlovic, and S. Kumar, “A new baseline for image annotation”, in *ECCV*, 2008, pp. 316–329.
- [6] L. Breiman, “Random forests”, *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [7] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, “LIBLINEAR: A library for large linear classification”, *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.
- [8] P. Duygulu, K. Barnard, J. De Freitas, and D. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary”, *ECCV*, 2002.
- [9] A. Oliva and A. Torralba, “Building the gist of a scene: The role of global image features in recognition”, *Progress in brain research*, vol. 155, pp. 23–36, 2006.