Learning Balanced Trees for Large Scale Image Classification

Tien-Dung Mai[1(✉)], Thanh Duc Ngo[1], Duy-Dinh Le[1,2], Duc Anh Duong[1], Kiem Hoang[1], and Shin'ichi Satoh[2]

[1] University of Information Technology, VNU-HCM, Ho Chi Minh City, Vietnam
{dungmt,thanhnd,duyld,ducda,kiemhv}@uit.edu.vn
[2] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
{ledduy,satoh}@nii.ac.jp

Abstract. The label tree is one of the popular approaches for the problem of large scale multi-class image classification in which the number of class labels is large, for example, several tens of thousands of labels. In learning stage, class labels are organized into a hierarchical tree, in which each node is associated with a subset of class labels and a classifier that determines which branch to follow; and each leaf node is associated with a single class label. In testing stage, the fact that a test example travels from the root of the tree to a leaf node reduces the test time significantly compared to the approach of using multiple binary one-versus-all classifiers. The balance of the learned tree structure is the key essential of the label tree approach. Previous methods for learning the tree structure use clustering techniques such as k-means or spectral clustering to group confused labels into clusters associated with the nodes. However, the output tree might not be balanced. We propose a method for learning effective and balanced tree structure by jointly optimizing the balance constraint and the confusion constraint. The experimental results on the datasets such as Caltech-256, SUN-397, and ImageNet-1K show that the classification accuracy of the proposed approach outperforms that of other state of the art methods.

1 Introduction

This paper considers the problem of multi-class image classification whose goal is to classify an image belongs to one of the different pre-defined classes. It is one of the essential problems in computer vision because of many potential applications such as object categorization, scene classification, and semantic image retrieval [5,6,13,14].

One approach to the multi-class classification problem is to use multiple binary one-versus-all classifiers [19]. However, this approach is not scalable to large-scale datasets (e.g., ImageNet [20] which includes 21,841 concepts with each of them associated with 1,000 images), because all classifiers have to be called at run-time for every image.

One popular approach to reduce the complexity is to use label tree [2,6,7,15]. In a label tree model, label of a test sample is assigned by traversing its tree.

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part II, LNCS 9280, pp. 3–13, 2015.
DOI: 10.1007/978-3-319-23234-8_1

At each node visited, a small number of classifiers are applied to compute scores to determine which branch to follow. This tree structure causes the classification complexity to grow logarithmically, rather than linearly, with the number of classes. The balance of the learned tree structure is therefore essential to the label tree approach.

Using label tree requires two tasks of learning the tree and learning node classifiers. Existing approaches [2,7,11,15] are either to separate or combine these two tasks in one optimization framework. Although the combined methods usually have higher classification performance, they are too costly because node classifiers are trained multiple times until the algorithm is converged. In this paper, we consider the methods that separate the two tasks as in [2] and focus to the first task of learning the tree.

Given a set of class labels at each node (the root node contains all class labels, and the leaf node contains a single class label) and the number of branches k, the problem is to split these labels into k groups. There are two constraints: (i) confused class labels should be in the same group and (ii) the number of class labels of the branches should be equal. The first constraint is to reduce the tree loss and to learn node classifiers with ease. The second constraint is to create the balanced tree.

The popular method is to use clustering methods such as k-means and spectral clustering [17]. For example, in [2], the confusion matrix is computed, and spectral clustering is used to recursively split class labels. Because the objective function of spectral clustering penalizes unbalanced partitions, it encourages balanced trees. However, this method is not reliable because it assumes the high correlation among the estimated confusion matrix and the real one. In practice, this assumption is not hold, especially when binary one-versus-all classifiers have poor accuracy due to small number of training samples and curse of dimensionality. Another method is to perform k-means clustering on training samples [16]. In this method, the mean of all feature vectors of the training samples of a class is used as representation for each class. This representation implicitly enforces the balanced constraint when using with k-means clustering. However, using the mean is not an effective way for classes with high variations.

We propose a method for learning effective and balanced trees by jointly optimize the balance constraint and confusion constraint. We avoid the unreliable situation when using confusion matrix and single feature vector for class representation described above by using all feature vectors of the training samples in each class. We formulate the learning process in an optimization framework in which the balance constraint is solved using integer linear programming and the confusion constraint is solved using k-means clustering. We tested the proposed method on several benchmarks datasets such as Caltech-256, SUN-397, and ImageNet-1K, and the result shows the superiority over other state of the methods.

The rest of the paper is organized as follows. In Sec 2, related works are presented. In Sec 3, the proposed method is describe. The experimental results are presented in Sec 4. Finally, Sec 5 concludes the paper.

2 Related Work

Learning tree structure is one of the main issues of a label tree-based approach. In [2], Bengio et al. proposed an approach to learn a tree structure base on spectral clustering. The approach utilizes confusion matrix generated by applying one-versus-all classifiers to a validation set as affinity measure to split classes into disjoint subsets. Each subset is corresponding to a child node of the tree. Such splitting procedure is repeated recursively until the whole tree is created. This approach has several limitations. Firstly, to obtain confusion matrix, multiple binary classifiers are learned with one-versus-all strategy. It therefore becomes costly when the number of classes increases. Secondly, since the spectral clustering approach does not guarantee equal partitions, the tree structure can be unbalanced, which leads to a sub-optimal test efficiency. Thirdly, the similarity between classes may not be reflected correctly via the affinity matrix due to low accuracy of the one-versus-all classifiers. As a result, classifiers of child nodes which are learned by using the set of class labels split by the above spectral clustering may give incorrect prediction.

In [7], Deng et al. proposed an approach which jointly performs class partitioning and learning a classifier for each child node. The one-versus-all training step is eliminated. Learning the classifier weights and determining the partitions are formulated as an optimization problem. It is then solved by two alternative optimization steps. However, by allowing overlapping of classes among child nodes to reduce false navigation, it at the same time increases the test cost thus cannot ensure a desired speedup.

Liu et al. in [15] proposed a probabilistic approach for learning tree structure. Each node of the probabilistic label tree is associated with a categorical probability distribution and a maximum likelihood classifier defined as a multinomial logistic regression model. Training process at each node is formulated as a maximum optimization of a log likelihood function which is then solved by using alternating convex optimization. Firstly, the maximum likelihood classifiers are learned based on the categorical distribution of each child node. Then, the categorical distribution associated with each child node is learned.

There are other solutions introduced for reducing the number of classifiers such as ECOC-based methods [1,4,8,9,18]. They mainly involve designing an optimal coding matrix which requires a small number of bits for efficiency, good row and column separation for robustness, and high accurate bit predictors. Sparse random codes and random codes described in [1,8] require a large number of bit predictors ($15.log(N)$ and $10.log(N)$ respectively where N is the number of classes) to achieve a reasonable accuracy. However, it is shown in [19], the accuracy of these methods is worse than that of the one-versus-all approach. Spectral ECOC [24] is based on spectral decomposition on the normalized Laplacians of the similarity graph of the classes. The resulting eigenvectors are used to define partitions. Because it uses one-versus-one classifiers to generate the similarity matrix, it is not scalable for classification problems with large number of classes. Recently, Sparse Output Coding (SpOC) [25] is a new encoding and decoding scheme that learns coding matrix and bit predictor separately but still has good balance between error-correcting ability and bit prediction accuracy. However, it uses a predefined class taxonomy to build a semantic relatedness matrix for the both stages.

3 Learning a Balanced Tree for Image Classification

3.1 Overview of Label Tree

Following the definition in [2,7], a set of class labels $L = \{1, ..., C\}$ are organized into a label tree $T = (V, E)$ with a set of nodes V and a set of edges E. Each node $v \in V$ is associated with a set of class labels $l(v) \subseteq L$ that indicates information about class belonging to node v and a set of children $\sigma(v) \subset V$. Note, a set of class labels of the root node contains all classes $l(v = root) = L$ and a set of class labels of a leaf node only contains one class $l(v = leaf) \subseteq L, |l(v = leaf)| = 1$. The edges connect each node $v \in V$ to a set of children $\sigma(v)$.

To make a branching decision at a node v, we train $|\sigma(v)|$ one-versus-all classifiers corresponding to its child nodes.

To classify the class of a test image x in the label tree, starting from the root node, classifiers are applied to the feature vector of x to determine response values. The child node which takes the largest value will be selected to go on. This process is then repeated until a leaf node is reached. The test image is classified into the class whose label associated with this leaf node. Since we only need to evaluate classifiers of nodes along the path from the root to a leaf node, the testing complexity is sub-linear. If the tree is balanced, the complexity is the logarithm of the number of classes.

Following the notation in [7], we use $T_{Q,H}$ to denote a label tree having Q children for each non-leaf node and maximum depth H. Depth of each node is defined as the maximum distance to the root (the root has the depth 0). These two parameters should be set so that the tree structure is balanced and Q^H approximate the number of classes

3.2 The Proposed Approach

In order to create a balanced label tree, the number of class labels in each child node, which have the same parent node, need to approximate each others. For example, if node v has N class labels and we want to split them into Q child nodes. Each child node has the maximum T_{max} class labels. The value of T_{max} can be calculated with the following formula:

$$T_{max} = Q^{H-1} \tag{1}$$

where, $H = \log_Q(N)$ is maximum level. For example, if the node v has $N = 1000$ class labels and $Q = 32$, we obtain $T_{max} = 32$.

Let matrix $S_{N\times Q}$ contains splitting information of N class labels as they are split into Q children nodes. The value of $S_{i,j}$ means:

$$S_{i,j} = \begin{cases} 1, & \text{if } i^{th}\text{class belong to} j^{th}\text{child node} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Since a class only belong to one child node, we have:

$$\sum_{i=1}^{Q} S_{i,j} = 1 \tag{3}$$

In addition, the constrain which each node has maximum T_{max} class labels can be represented as:

$$\sum_{j=1}^{N} S_{i,j} \leq T_{max} \tag{4}$$

We follow the main idea of k-means algorithm, let $F_{N \times Q}$ be a matrix with $F_{i,j}$ to be the average distance from all images of class i to the center of cluster j. Each cluster is corresponding to a child node. If class i belongs to cluster j, the value of $F_{i,j}$ is minimized. This implies that the sum of average distance of classes belongs to $\ell(j)$ must to be minimized.

$$\min_{\ell(j)} \sum_{i \in \ell(j)} F_{i,j} \tag{5}$$

In general, we find values $S_{i,j}$ so that the sum of average distances between all images of classes and its nearest cluster center is minimized.

$$\min_{S,F} \sum_{i=1}^{N} \sum_{j=1}^{Q} S_{i,j} \cdot F_{i,j} \tag{6}$$

subject to the constraints (3) and (4).

The problem (6) is a minimum optimization problem with two variables S and F. It can be solved by using two alternating convex optimizations. In the first step, F is fixed, the problem (6) can be regraded as an integer linear programming problem subject to the constraints (2), (3) and (4), where S represents the integer variable to be determined, F are coefficients. Next, S is fixed, we update the cluster centers of classes which correspond to the non-zero values in columns of S, then we can obtain F by calculating the average distance from all images to these centers. This optimization can be repeated with a fixed number of iterations t (in our implementation, we set $t = 5$) or repeated until the solution is converged.

We summarize the algorithm for splitting set of class labels at node in Algorithm 1.

4 Experiments

4.1 Datasets

We conduct experiments on several benchmark datasets including Caltech-256 [12], SUN-397 [23] and ImageNet-1K [20]. These datasets are widely used to evaluate both hierarchy-based and flat-based approaches for large-scale image classification.

Algorithm 1. Splitting set of class labels $\ell(v)$ into Q child nodes

Input: $X = \{(x_i, y_i)\}, \cup y_i = \ell(v), |\ell(v)| = N$: the set of training images
Q: the number of child nodes.
t: the fixed number of iterations

Output: The class label sets of Q child nodes.

1. Initialize: Compute the mean of all feature vectors of the training images of a class, namely, $\bar{X}$. And then, use k-means algorithm for clustering $\bar{X}$ into Q clusters with centers C_Q: $C_Q = k\text{-means}(X, Q)$.
2. For each class, compute averge distance from all feature vectors of the training images to centers C_Q. We obtain $F_{N\times Q} = ave\_distance(X, C_Q)$.
3. Fix F, solve (6) for S and update centers C_Q
4. Repeat step 2 until (6) convergence or a specified number of iterations t is reached.

- **Caltech-256** [12] dataset. This is a multi-class object recognition dataset with 29,780 images of 256 classes. Each class contains at least 80 images of varying size and quality. Most of classes are relatively independent of one another.

- **SUN-397** [23] dataset. This is a scene classification dataset. It contains 108,754 images of 397 classes well-sampled from 908 scene classes of the SUN dataset. There are at least 100 images per class.

- **ImageNet-1K** or ILSVRC2010 [20] is a subset of ImageNet. It provides images of 1,000 classes, separated into three parts. The first part is a set of 1,261,406 images for training (at least 668 images per class). The second part includes 50,000 images for validation (50 images per class). And, the third part contains 150,000 images for testing (150 images per class).

4.2 Experimental Setting

With Caltech-256 and SUN-397, we split the original dataset into three disjoint subsets as following: 50% of images are for training, 25% of images for validation, and the last 25% of images are for testing. With ImageNet-1K, we use the provided image sets for validation and testing. We randomly pick 100 images of each class for training.

For each image, we extract dense SIFT features using VLFeat toolbox [21]. These features are then encoded using LLC encoding approach [22] with two level spatial pyramid (1×1 and 2×2 grids) [13] for pooling. Using a codebook with 10,000 visual words, we obtain a 50,000 dimensional feature vector for each image. LIBLINEAR (version 1.96) library [10] is used for training linear SVM classifiers with one-versus-all strategy.

We re-implemented the approaches proposed by Liu et al. [16] and Bengio et al. [2] as a base-line for comparison. Specially, [2] is considered as the original label tree based learning approach. First, we train n classifiers independently with one-versus-all strategy. We then apply these classifiers on a validation set to obtain a confusion matrix C. For each node v of the tree, we obtain a matrix $A = \frac{1}{2}(\bar{C} + \bar{C}^T)$ with $\bar{C}_{i,j} = C_{\ell(v)_i,\ell(v)_j}$. Regarding A as the affinity matrix, a standard spectral clustering is then used to partition the label sets between classes. However, since the objective function of spectral clustering penalizes unbalanced partitions, it might generate an unbalanced tree. In our experiments, we used the constrained k-means [3] instead of the k-means in clustering step to obtain a better balanced tree.

4.3 Evaluation Measurement

We employ standard measurements, global accuracy and test speedup [7], for evaluating the proposed approach and other approaches for comparison.

Global Accuracy. The global classification accuracy (Acc) is defined following:

$$Acc = \frac{1}{m}\sum_{i=1}^{m} fi(\hat{y}_i = y_i) \tag{7}$$

where, m is the total number of testing images and $fi(\hat{y}_i = y_i)$ is an indicator function. $fi(\hat{y}_i = y_i) = 1$ if the predicted class $\hat{y}_i$ is similar to the assigned class y_i of the image x_i; otherwise, $fi(\hat{y}_i = y_i) = 0$.

Test Speedup. Test speedup (S_{te}) is measured as the test cost of one-versus-all based approach divided by the test cost of the label tree based approach. Test costs are computed as the average number of vector operations (dot-products) required for classifying a testing image. If linear classifiers are used, the values of S_{te} can be defined as following:

$$S_{te} = \frac{n * m}{N}, \tag{8}$$

where n is the number of classes, m is the total number of testing images, and N is the total number of vector operations performed for classifying m testing images. A higher value of S_{te} indicates more efficient approach in terms of computational cost. It also means less number of classifiers evaluated on a test image to give the final class decision.

4.4 Experimental Results

Experimental results are presented in Table 1, 2, 3 corresponding to dataset ImageNet-1K, SUN-379, and Caltech-256 respectively.

To obtain stable experimental results, we trained and evaluated the approaches with different subsets selected by randomly sampling images in classes. We then reported the average classification performance with the corresponding standard

Table 1. Comparison the performance of the evaluated approaches on ImageNet-1K

Approaches	Flat		T32,2		T10,3		T6,4		T4,5	
	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}
Bengio et al. [2]			7.22 ± 0.21	15.78 ± 0.03	5.20 ± 0.03	33.33 ± 0.00	4.61 ± 0.10	42.28 ± 0.20	4.05 ± 0.06	49.89 ± 0.27
Deng et al. [7]			11.90	10.3	8.92	18.20	5.62	31.3		
Liu et al. [16]			12.12 ± 0.03	15.64 ± 0.01	9.73 ± 0.15	33.33 ± 0.00	9.39 ± 0.33	40.93 ± 1.56	8.68 ± 0.34	49.94 ± 0.22
Our approach			**13.14** ± 0.04	15.77 ± 0.00	**10.74** ± 0.07	33.33 ± 0.00	**9.85** ± 0.09	42.24 ± 0.9	**9.61** ± 0.13	50.06 ± 0.02
One-versus-All	26.01	1								

deviation. Moreover, the accuracy of multi-class classification using one-versus-all classifiers trained with LIBLINEAR [10] are also reported for reference.

We compare our proposed approach with other tree-based approaches proposed Bengio et al. [2], Deng et al. [7], and Liu et al. [16]. Each row of the table presents performance of one approach. Meanwhile, columns are related to tree configurations with different numbers of branches at a node and tree levels. For example, $T_{32,2}$ in Table 1 indicates a 2-level tree with 32 branches at a node. Note that the flat-based approach can be considered a special case of a tree-based approach. Given the number of level equals to 1, a tree becomes flat. And, the performance of a tree-based approach is strongly affected by the changes of tree configuration.

Generally, as we increase the number of level of a tree, the path from the root to a leaf node i.e. an individual class is lengthened. However, since the number of branches at a node i.e. the number of classifier evaluated at a node is decreased, the test speed up is significantly improved. But, this also results in accuracy drop. Tree configuration therefore can be adaptively selected to balance accuracy and computational cost for a specific practical need.

The essential conclusion can be drawn from the experimental results is that our proposed approach outperform other tree-based approach. At the same accuracy level, our approach is usually more efficient (i.e. higher Ste) than the other tree based approaches. Meanwhile, at the same speed up level, we achieve higher accuracy in most of the cases.

For example, as shown in Table 1, the average classification accuracy is significantly higher for the trees learned using our approach. For the tree $T_{10,3}$, there are approximately $10 * 3$ classifiers evaluated for a test image, so we achieved $1000/30 \approx 33.33$ speedup with the accuracy $10.74 \pm 0.07\%$. Meanwhile, the average accuracy of the approaches proposed by Bengio et al. [2], Deng et al. [7], and Liu et al [16] are $5.20 \pm 0.03\%$, 8.92% and $9.73 \pm 0.15\%$ respectively. As shown in Figure 1, the our method achieves comparable or significantly better classification accuracy at the same test speedup. Note that as we evaluate the approach proposed by Deng et al. [7], since it allows overlapping among child

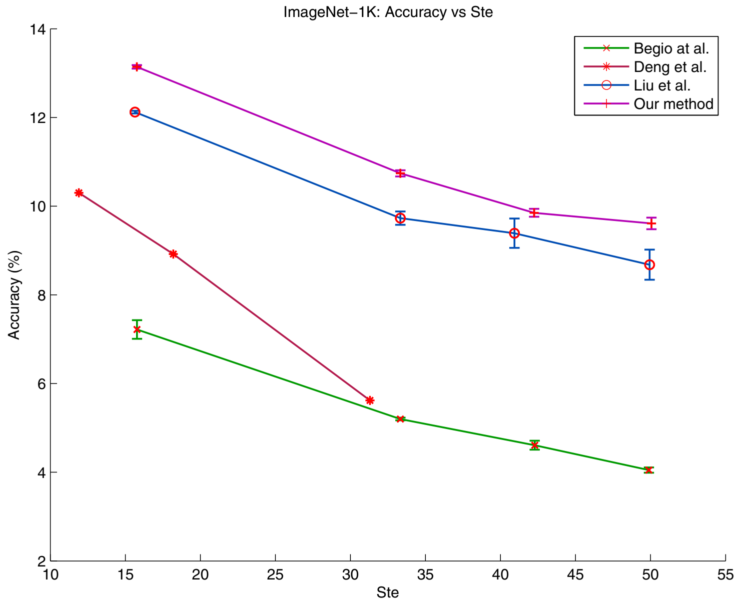

Fig. 1. Performance of the evaluated approaches on ImageNet-1K

nodes, it usually requires more evaluation cost at each level (i.e. smaller *Ste* in result).

Similar observation can be found in Table 2 and Table 3. The results in Table 2 show that for all types of tree configurations, our method achieves comparable or significantly better classification accuracy at the same test speedup on SUN-397 dataset. Table 3 shown the relationship between the average accuracy and the test speedup on Caltech-256 dataset. It shows that the better performance of our method compare to the others.

Table 2. Comparison the performance of the evaluated approaches on SUN-397

Approaches	Flat		T20,2		T8,3		T5,4		T2,9	
	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}
Bengio et al. [2]			30.86 ± 0.13	9.96 ± 0.02	25.76 ± 0.09	17.28 ± 0.06	22.83 ± 0.98	20.83 ± 0.14	15.91 ± 0.29	22.70 ± 0.10
Liu et al. [16]			37.34 ± 0.27	9.93 ± 0.00	35.48 ± 0.37	16.96 ± 0.08	33.55 ± 0.55	20.36 ± 0.41	28.37 ± 0.92	22.51 ± 0.02
Our approach			**38.32** ± 0.41	9.97 ± 0.01	**35.87** ± 0.57	17.16 ± 0.04	33.28 ± 0.28	20.90 ± 0.14	**29.46** ± 0.31	22.73 ± 0.07
One-versus-All	50.99	1								

Table 3. Comparison the performance of the evaluated approaches on Caltech-256. Our method achieves outperform accuracy than the others.

Approaches	Flat		T16,2		T7,3		T4,4		T2,8	
	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}	Acc%	S_{te}
Bengio et al. [2]			31.79 ± 0.69	8.00 ± 0.00	27.56 ± 0.47	12.55 ± 0.07	25.47 ± 0.22	16.00 ± 0.00	22.87 ± 0.29	16.00 ± 0.00
Liu et al. [16]			37.13 ± 0.60	8.00 ± 0.00	34.07 ± 0.94	12.40 ± 0.09	31.69 ± 0.14	16.00 ± 0.00	29.15 ± 0.48	16.00 ± 0.00
Our approach			**39.13** ± 0.21	8.00 ± 0.00	**35.07** ± 0.16	12.70 ± 0.04	**33.02** ± 0.43	16.00 ± 0.00	**29.68** ± 0.68	16.00 ± 0.00
One-versus-All	50.95	1								

5 Conclusion

The label tree approach is an efficient technique for the problem of large scale multi-class image classification. We have proposed a method for learning an effective and balanced tree that jointly optimize both the balance constraint and confusion constraint. We compared our proposed method with other state of the art methods in experiments on the large datasets such as Caltech-256, SUN-397, and ImageNet-1K. The results show that our proposed method achieves the best performance among the methods.

Acknowledgments. This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number B2015-26-01.

References

1. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multi-class to binary: a unifying approach for margin classifiers. J. Mach. Learn. Res. (2001)
2. Bengio, S., Weston, J., Grangier, D.: Label embedding trees for large multi-class tasks. In: Advances in Neural Information Processing Systems (NIPS), pp. 163–171 (2010)
3. Bradley, P., Bennett, K., Demiriz, A.: Constrained k-means clustering, pp. 1–8. Microsoft Research, Redmond (2000)
4. Crammer, K., Singer, Y.: On the learnability and design of output codes for multiclass problems. Machine Learning (2002)
5. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Comput. Surv. **40**(2), 5:1–5:60 (2008)
6. Deng, J., Berg, A., Fei-Fei, L.: Hierarchical semantic indexing for large scale image retrieval. In: The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, June 2011
7. Deng, J., Satheesh, S., Berg, A., Fei-Fei, L.: Fast and balanced: efficient label tree learning for large scale object recognition. In: Proceedings of the Neural Information Processing Systems (NIPS) (2011)
8. Dietterich, T.G., Bakiri, G.: Solving multi-class learning problems via error-correcting output codes. J. A.I. Res. (1995)

9. Escalera, S., Tax, M., Pujol, O., Radeva, P.: Subclass problem-dependent design for error-correcting output codes. PAMI (2008)
10. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
11. Gao, T., Koller, D.: Discriminative learning of relaxed hierarchy for large-scale visual recognition. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2072–2079, November 2011
12. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Tech. Rep. 7694, California Institute of Technology (2007). http://authors.library.caltech.edu/7694
13. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178. IEEE Computer Society (2006)
14. Li, L., Socher, R., Li, F.: Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), June 20–25, 2009, Miami, Florida, USA, pp. 2036–2043 (2009)
15. Liu, B., Sadeghi, F., Tappen, M., Shamir, O., Liu, C.: Probabilistic label trees for efficient large scale image classification. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 843–850. IEEE (2013)
16. Liu, S., Yi, H., Chia, L.T., Rajan, D.: Adaptive hierarchical multi-class svm classifier for texture-based image classification. In: IEEE International Conference on Multimedia and Expo. ICME 2005, pp. 1190–1193, July 2005
17. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 2, pp. 849–856 (2002)
18. Pujol, O., Radeva, P., Vitria, J.: Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. PAMI (2006)
19. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. **5**, 101–141 (2004)
20. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge (2014)
21. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008). http://www.vlfeat.org/
22. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y.: Locality-constrained linear coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3360–3367. IEEE (2010)
23. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: Computer Vision and Pattern Recognition (CVPR), pp. 3485–3492 (2010)
24. Zhang, X., Liang, L., Shum, H.: Spectral error correcting output codes for efficient multiclass recognition. In: ICCV (2009)
25. Zhao, B., Xing, E.P.: Sparse output coding for large-scale visual recognition. In: CVPR (2013)