Evaluation of Random Forests on large-scale classification problems using a Bag-of-Visual-Words representation

Xavier SOLÉ [a,1] Arnau RAMISA [a] and Carme TORRAS [a]

[a] *Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas 4-6, 08028 Barcelona, Spain*

Abstract. Random Forest is a very efficient classification method that has shown success in tasks like image segmentation or object detection, but has not been applied yet in large-scale image classification scenarios using a Bag-of-Visual-Words representation. In this work we evaluate the performance of Random Forest on the ImageNet dataset, and compare it to standard approaches in the state-of-the-art.

Keywords. large-scale image classification, classifier forest, random forests

Introduction

In recent years, "big data" has emerged as a trove of enormous potential to tackle problems typically too hard for artificial intelligence: large datasets can be used to learn very accurate predictors, able to match or even surpass expert human performance in tasks such as automatic translation, medical diagnostics or legal counseling. These predictors can then be used to automate processes, and provide service to citizens at a level never imagined before.

However, in such scenario, it is as important to learn good classifiers as it is to have methods with a very low computational footprint at test time, since then a system can be scaled to dimensions that can truly serve millions of users simultaneously. A good example of this are web services: more than a half billion search queries have to be served each day, and thousands of pictures are uploaded to photo-sharing sites every minute. Services such as picture search by example or automatic tagging of new photos have, therefore, to be based on methods very efficient at test time to be useful in practice.

In this context, the Random Forest method, proposed by Breiman et al. [1], is a good candidate, as its computational cost at test time is very small. This machine learning approach, that combines discriminative and generative aspects, can be used for multiple tasks, like classification, regression or density estimation. However, even though Random Forests have been used in many different fields with successful results [2,3,4], to the best of our knowledge it has not yet been evaluated in a large-scale image classification scenario using a Bag-of-Visual-Words representation.

In order to see how Random Forest behaves on a large scale image classification context, and to study its error and computational complexity, we have evaluated their performance in the ImageNet [5] Large Scale Visual Recognition Challenge'10 (LSVRC'10) dataset. We also compare the performance obtained with Random Forest in this dataset to that of two other methods for multi-class image classification: the widely used One-

[1] Corresponding Author: Xavier Solé. E-mail: xavisn1@gmail.com

versus-Rest Support Vector Machines (OvR-SVM) approach, and the Ensembles of Class Balanced Nested Dichotomies (ECBND), both reported in an earlier work [6].

1. Random Forest

In this section we will briefly describe the Random Forest method by Breiman et al. [1]. Random Forests are sets of random decision trees constructed as follows: beginning in the root node, we separate the initial set of training images I using some split function into two disjoint sets; then this procedure is recursively repeated until a stopping criterion is met, and a leaf node is generated. Each leaf node has an associated probability distribution over classes $c_j \in C$, computed as the fraction of images labeled as c_j that reached the leaf node. A forest of random trees is generated by repeating the random tree creation process.

At test time, a new example traverses each random tree in the forest to determine its corresponding leaf node. Then, the probability distribution over classes for this particular example is computed as the average of the distributions of the leaf nodes reached in every tree.

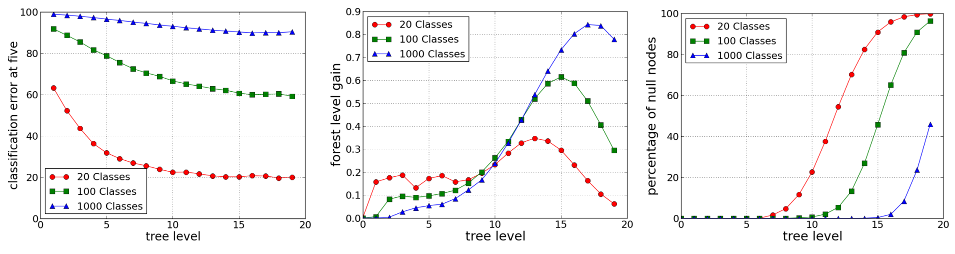
The objective function that is optimized during the creation of a random tree is the information gain, measured as the Shannon entropy between the labels of the images in a node, and the combination with those of its descendants. The split function associated with the internal nodes is computed by randomly generating a number of random axis-aligned divisions of the set of vectors associated with a node and then choosing the one that induces the highest gain. The creation of division candidates is controlled by two parameters: *number of candidate features* and *number of candidate thresholds per feature*.

Computational cost. The computational complexity at test time for a Random Forest of size T and maximum depth D (excluding the root) is $O(T \cdot D)$. However, the computational cost can be lower if trees are not balanced. It must be noted that unification costs of ensemble methods, left out in the theoretical cost computation as they are often negligible, in the regime where Random Forest operates become quite significant and dominate the total cost. Another important cost to be considered in our method is memory space, exponential in the depth of the tree: $O(2^D)$.

2. Experimental Results

We have evaluated the Random Forest method on the ImageNet Large Scale Visual Recognition Challenge'10 (LSVRC'10) dataset. This data set is formed by 1000 classes, approximately one million training images and 150k testing images, with categories as diverse as “lemon”, “cress”, “web site”, “church” or “Indian elephant”. Since it is common to find images with more than one category of objects, the recommended evaluation criterion is error at five, i.e. five class predictions are allowed for each image without penalization. To facilitate comparison with related work, we used the demonstration Bag-of-Visual-Words (BoVW) features for the LSVRC'10 dataset[2]. Our implementation of the Random Forest method is based on the Sherwood C++ library [2].

[2]`http://www.image-net.org/challenges/LSVRC/2010/download-public`



(a) (b) (c)

Figure 1. (a) Error at five results for Random Forest, $T = 60$ in all experiments. (b) Gain (objective function) for each tree level. (c) Average percentage of "*null nodes*", i.e. branches terminated at a previous level.

Small-scale experiments. The first step in our experiments consisted in evaluating the performance of Random Forest with a small number of classes (20 and 100 classes), before moving to the large-scale case. First, we adjusted the randomness parameters, fixing D and T. The optimal parameters found were used in the rest of the experiments.

Next, we conducted experiments to see how the size of the forest T and the maximum depth D affect the results, interleaving the optimization of the two parameters. In practice, we fixed a depth, and then increased the size of the forest until the error saturated (i.e. adding trees does not result in more accurate predictions). Figure 1a shows how the tree depth affects the error at five and also that, although results with a small number of classes are good, the error increased dramatically as more classes were considered.

Experiments with full dataset. Finally we conducted experiments with all 1000 classes varying the size of the forest (T and D parameters). Figure 2 shows the experimental results as well as computational and spatial complexity of Random Forests of increasing sizes. We have chosen operating points for the parameters D and T that, at a higher cost, improve the error at five of the final result. This way we explore the most optimistic possibilities of the Random Forest method. As can be seen in Figure 2a, the cost grows fast with respect to the error at five, but the real limiting factor is the exponential spatial cost, as can be seen in Figure 2b. We observe only a moderate decrease of the error with an exponential increase in computational requirements.

To estimate the improvement we can expect by increasing D we use the evolution of entropy between consecutive levels. Therefore, in order to find out if we are close to the optimal D parameter, we compute the *forest level gain*: $G(F,l) = E(F,l-1) - E(F,l)$, that tells us how increasing D translates in gain towards our objective function. $E(F,l)$ is the average forest F entropy at level l. In particular, it allows us to model how the forest evolves when increasing D to values greater that 17, where it is impractical to operate due to space constraints.

Extrapolating the bell-shaped curves of the small-scale experiments to the full-scale one, we can conclude that at depth 17 we are close to the maximum gain per level, and that increasing the size of the tree would yield diminishing returns. This conclusion is supported by Figure 1a, where we can see that the error at five does not decrease significantly when D increases above depth 13 for the 20 classes experiment and above depth 15 for the 100 classes one.

Finally, Figure 1c suggests that approximating the computational cost with the worst case scenario (i.e. a balanced tree) is accurate, since at the second to last level used in our experiments, trees only have 1.89% null nodes on average, and 8.35% at the last.

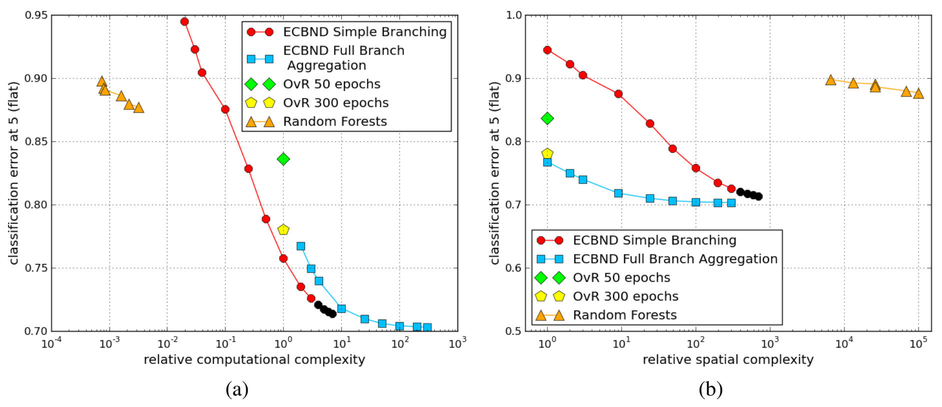


Figure 2. Error at five of the evaluated methods on the LSVRC'10 dataset at different complexity points.

Comparison with other methods. We compare the performance of the Random Forest classifier in the LSVRC'10 dataset with those of the Ensembles of Class-Balanced Nested Dichotomies (ECBND) and One-versus-Rest Support Vector Machines (OvR-SVM) classifiers used in [6]. In Figure 2a we can see that Random Forest obtains worse results overall than ECBND and OvR-SVM, but with a much lower computational cost at the same error at five. However, as discussed earlier, the most pressing limitation for the Random Forest method is the memory requirements. In Figure 2b we can see the relation between memory complexity and error at five for the evaluated methods.

3. Conclusions

In this work we have seen that Random Forest is unpractical for large-scale multi-class image classification problems using BoVW because of the high spatial cost and low accuracy. Other hierarchical classification methods such as ECBND are much more discriminative at each node, thus requiring less levels to reach the same error. On the other hand, Random Forest attains the lowest computational complexity at testing time, and for certain problems they can be the best choice [3].

Acknowledgements

This research is partially funded by the Spanish Ministry of Science and Innovation under project DPI2011-27510, by the CSIC project CINNOVA (201150E088) and by the ERA-Net CHISTERA project ViSen PCIN-2013-047. A. Ramisa worked under CSIC/FSE JAE-Doc grant.

References

[1] L. Breiman, "Random Forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] A. Criminisi and J. Shotton, *Decision Forests for Computer Vision and Medical Image Analysis*. Springer, 2013.

[3] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, pp. 1297–1304, 2011.

[4] A. Bosch, A. Zisserman, and X. Munoz, "Image Classification using Random Forests and Ferns," in *ICCV*, pp. 1–8, 2007.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, pp. 248–255, 2009.

[6] A. Ramisa and C. Torras, "Large-scale image classification using ensembles of nested dichotomies," in *Frontiers in Artificial Intelligence and Applications: Artificial Intelligence Research and Development*, vol. 256, pp. 87–90, IOS Press, 2013.