Machine Learning and Data Mining project: leaf identification

Walter Nadalin

Course of AA 2021-2022 - Data Science and Scientific Computing

1 Problem statement

The purpose of this project is to implement an automatic method to classify some leaves based on some of their attributes. The leaves and attributes considered are contained in a publicly available dataset. In other words, the goal is to build a model that can predict a leaf's species given its characteristics. So the input of the model is a numeric vector $x \in \mathbb{R}^p$, with $p = 14$ number of attributes, and its output is a label related to one of the species in the dataset.

2 Proposed solution

The proposed solution is to grow a Random Forest (RF) from the given data. RFs are constructed using the same principles as decision trees [2] and bagging [5]. In addition they help reduce tree correlation by injecting more randomness into the tree growth process. More specifically, during the growth of a tree, the search for the splitting variable is limited to a number $m < p$ of features. The algorithm is implemented in `R` using `setseed(20)` for reproducibility.

3 Assessment and performance indexes

The main performance index used is the fraction of miss-classified samples and some estimates of it, like the Out-Of-Bag (OOB) error estimate.

In a RF there are various parameters to tune:

- number of trees: it needs to be sufficiently large to stabilize the error, a good starting point is 10 times the number of features;
- m: helps to balance low tree correlation with reasonable predictive strength, a good starting point for classification is $\sqrt{p}$;
- tree complexity: node size is a common parameter chosen to control tree complexity and often the default value is of 1 [3], [1] for classification;

a more thorough discussion about them is provided by [6]. A search for the optimal parameter's values is therefore necessary. The values reported above are used to build a first model. Then they are changed and a new RF is grown on all the data. This is done many times, each time the OOB estimate is extracted.

To provide an understanding of the final model's generalizability and to ensure that the error obtained is reliable, the following steps are performed:

1. the dataset is divided into two subsets, one for training (80% of the data) and one for testing. Two models with the chosen values for the parameters are trained on the first set: the first with an external Cross-Validation (CV) and the second without any re-sampling method (obviously there's still the bootstrapping of the RF). The models are then used to make a prediction on the second set. In this step the errors are also compared with the ones obtained with a single tree and with a bag of trees;

2. after verifying that the previous step produced a good match between the CV error of the first model, the OOB estimate of the second and the measured errors on unseen data, a final model is learned on all available data without CV and the obtained OOB estimate is reported;

4 Experimental evaluation

Many modern implementations of RFs exist, in this work Leo Breiman's algorithm [4] is used. For the implementation the `ranger` [9] package is exploited. It provides an implementation of the said algorithm in the programming language `R`. Other relevant packages used are the `caret` package for re-sampling, the `ipred` package to create the bag of trees and the `rpart` the package to grow the single tree.

4.1 Data

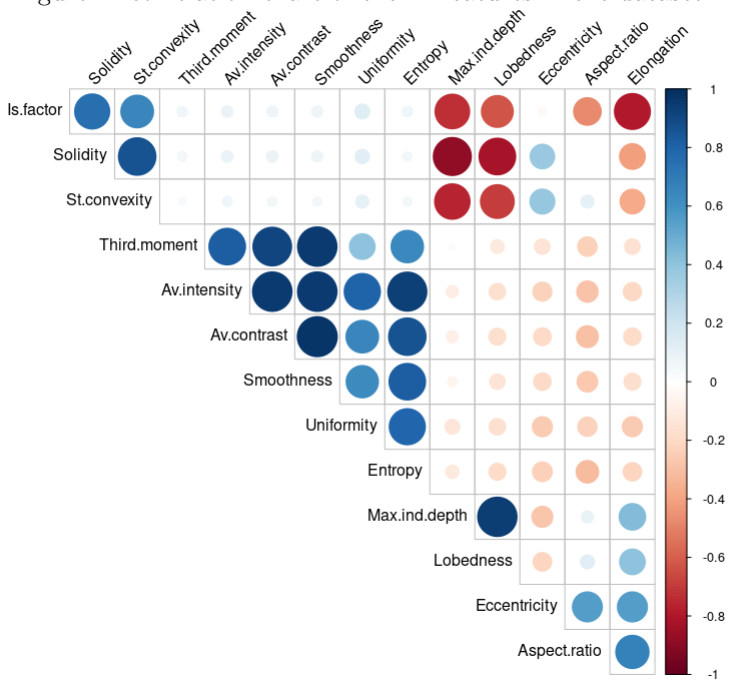
A description of the dataset is available and states that it includes 40 different plant species. However only 30 are present in it. For each species, measurements of the features of about 10 samples are given, for a total of 339 samples. There are 14 attributes per sample: 7 related to shape and 7 to the structure.

By plotting the correlation chart (1), considering the Pearson correlation coefficient [7], we can observe some highly correlated features. Moreover, by building the first RF (with 140 trees, $m = 3$ and node size 1) we get an idea of the importance (impurity-based and permutation-based) of the features. In both cases smoothness and av. contrast are among the 4 less important features.

Given these results, an improvement is obtained by removing the following features when learning the first model on all the data: smoothness and average contrast. Indeed, the overall OOB error goes from 0.25 (with all the features) to 0.22[1]. Therefore only the 12 of the 14 attributes are considered.

[1] This result was also tested with other seeds, from 1 to 1000, and in all but 248 cases the error estimate considering all features was worse and the training with all was always slower.

Figure 1: correlation chart of the 14 features in the dataset.



4.2 Procedure

First, the minimum number of trees after which the error stabilizes is obtained by going from 10 to 2000 in steps of 10. In this procedure, all RFs are obtained by considering all data with $m = 3$ and node size equal to 1. The graph (2) suggests that it is possible to consider a value around 1500.

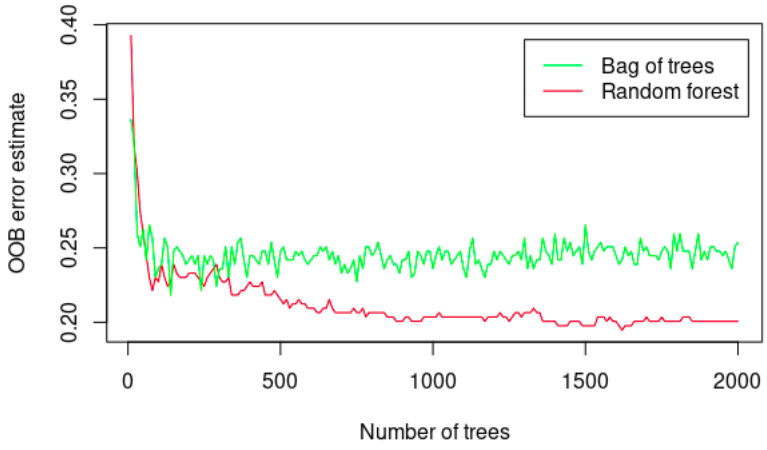
Now the other parameters are tuned. Values from 1 to 10 for node size and m are considered. Then 100 models are learned. The comparison suggests that values 6 for node size and 2 for m are the best.

To compare the models the estimated and real errors are considered. They are obtained training various models only on the learning set with or without an external 5-fold CV repeated 10 times. When this re-sampling technique is not applied the estimates of the error are the OOB estimates[2]. The results are reported in (1). Notice that 200 unpruned trees are learned for the bag of trees and for the optiman single tree is found by pruning a fully grown one using a cost complexity parameter. Its value is chosen among 20 values using CV.

As last step the best serial random forest is learned on all the data and the OOB estimate obtained is provided.

[2]The OOB estimate isn't available for a single tree so the case without re-sampling wasn't considered for it.

Figure 2: OOB error estimate for different number of trees (for the bag of trees unpruned trees are used).



4.3 Results and discussion

The table (1) shows the errors obtained in the first of the two steps discussed in 3. It can be observed that the RF outperforms the other models. In addition, an OOB estimate of 0.20 is obtained in the second step. However, the table suggests that this is likely an underestimate of the error that could be obtained on future unseen data. The model learned on all data is still expected to perform better than the models trained on a portion of them though.

Table 1: comparison of the estimated and real errors between the models. (A) and (B) refer respectively to the usage or not of CV.

Error →	Estimated (A)	Real (A)	Estimated (B)	Real (B)
Single tree	0.57	0.57	/	/
Bag of trees	0.24	0.29	0.23	0.29
RF	0.23	0.25	0.23	0.25

As a comparison, the same dataset was used in [8] with the same objective in mind. In that work, the best results were achieved with a CV error of 0.22 and an error of 0.16 in the test set. In this work the same indices are equal to 0.23 and 0.25. Which are close but not as good. They also use less data for training. This indicates a worse performance all around.

Overall, the solution found works well but not extremely well, as it is expected to fail in 2 or 3 out of 10 cases. Moreover, changing the seed can result in better or worse performance by looking at the same indices. There are also cases where the RF seems to perform worse than the bag.

One possible way to try to improve this work would be to generate more samples from the available data using some suitable technique.

References

[1] Eric C. Polley Benjamin A. Goldstein and Farren B. S. Briggs. Random forests for genetic association studies. *Statistical Applications in Genetics and Molecular Biology*, 10(1), 2011.

[2] Olshen R.A. Breiman L., Friedman J.H. and Stone C.J. *Classification And Regression Trees*. Routledge, 1 edition, 1984.

[3] Ernst D. Geurts P. and Wehenkel L. Extremely randomized trees. *Machine Learning*, 63:3–42, March 2006.

[4] Breiman Leo. Random forests. *Machine Learning*, 45:123–140, August 1996.

[5] Breiman Leo. Bagging predictors. *Machine Learning*, 45:5–32, October 2001.

[6] Philipp Probst, Marvin N. Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3), 2019.

[7] Boer C. Schober P. and Schwarte L. Correlation coefficients: Appropriate use and interpretation. *Anesthesia and Analgesia*, 126:1, february 2018.

[8] Pedro Filipe Barros Silva. Development of a system for automatic plant species recognition. Master's thesis, Faculdade de Ciências da Universidade do Porto, 2013.

[9] Marvin N. Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data. *Journal of Statistical Software*, 77:1–17, 2017.