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 certain 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 the species of a leaf given its attributes. Thus, the input of the model is a numerical 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 provided data. This choice is dictated by the fact that an RF is a common algorithm implemented in many different packages and libraries and generally performs well. That said, a RF should only be a starting point and more solutions should be explored before being satisfied. On the other hand, the method one should use to solve a problem also depends on the available resources: in this case, time was not one of them for me.

3 Assessment and performance indexes

RFs tend to generally provide a very good performance. Although they have several parameters that can be tuned, defaults usually produce good results. Furthermore, [7] illustrates that among other popular Machine Learning (ML) algorithms, RFs have the least variability in their accuracy when tuning. Despite this, one should tune the parameters when training a model. The parameter tuned in this work are:

- number of trees: it needs to be sufficiently large to stabilize the error;
- m: helps to balance low tree correlation with reasonable predictive strength, a good starting point for classification is $\sqrt{p}$;

- node size: the default value is of 1 [2], [1]. However, [5] showed that if the data has many noisy predictors and higher node size values are performing best, then performance may improve by increasing node size;

- sampling scheme: the default is bootstrapping where all of the observations are sampled with replacement. It is possible to adjust both the sample size and whether to sample with or without replacement. The sample size determines how many observations are drawn for the training of each tree: decreasing it leads to lower tree correlation which can have a positive effect. Moreover, sampling with replacement can lead to biased variable split selection [10].

a more thorough discussion about them is provided by [6].

After the tuning, to provide an estimate of the model's future error and to ensure that the estimation is reliable, the dataset is divided into two subsets, one for training (70%[1] of the data) and one for testing. Two models with the optimal parameters are trained on the first set: one with an external Cross-Validation (CV) and the other without any external sampling method. The models are then used to make a prediction on the second set. Here the errors are also compared against those obtained with a single tree and a bag of trees.

At last, after verifying that the previous step produced a good match between the estimated and measured errors, the 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` package [11] is exploited[2]. It provides an implementation of the said algorithm in the language `R`. Other relevant packages used are the `caret` for re-sampling, `ipred` for the bag of trees, `rpart` for the single tree and `tuneRanger` [6] for tuning of the RF. As a final note `set.seed(4)` is used for reproducibility in `R`.

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.

Selecting the right features in the data can help to improve the performance of the model. By building an RF with the default values for the parameters and 500 trees it is possible to obtain the importance (impurity-based

[1] The choice of why 70% is given by the fact that [9] used the same percentage: in this way we compare directly the results obtained.

[2] By default, this implementation sets the m parameter to $\lfloor\sqrt{p}\rfloor$ and node size equal to 1.

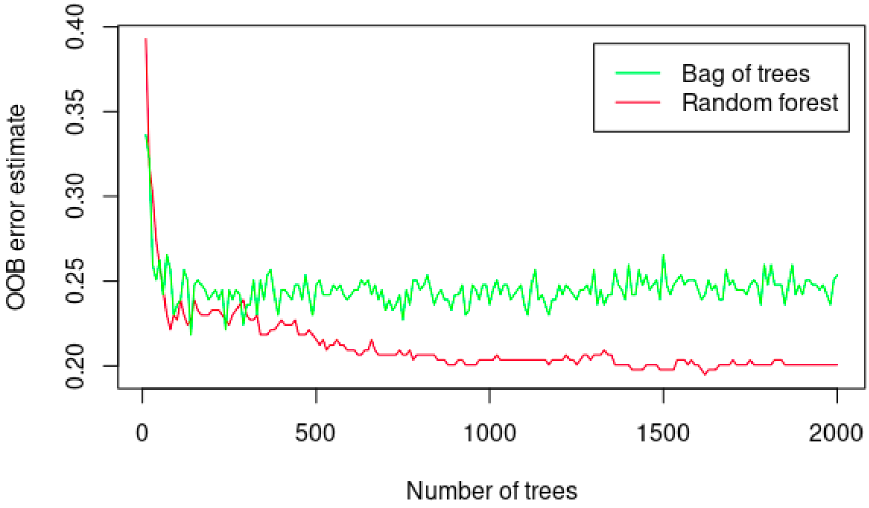
and permutation-based) of the features. In both cases smoothness and average contrast are the 2 less important features. In addition, by calculating the Pearson correlation coefficient [8] one can recognize the presence of some highly correlated features.

However, in presence of correlated predictors in high-dimensions classification frameworks variable selection is difficult. [3] motivates the use of the Recursive Feature Elimination (RFE) algorithm for variable selection in this context. RFE is a popular automatic method for feature selection and is also provided by the `caret` package. It works with a RF algorithm which is configured to explore all possible subsets of the attributes. Its application suggests to drop the smoothness, which agrees with what found before, therefore it is discarded.

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 the data and the default paramaters. The plot (1) suggests that it is possible to consider a value around 1500.

Figure 1: OOB error estimate for different number of trees considering a bag of trees with unpruned trees and a RF with $m = 3$, node size equal to 1, sample size equal to the total number of observation and sampling with replacement.



Now the other parameters are tuned. Model based optimization is used as tuning strategy and 200 combination of the parameters are tried. OOB predictions are used for evaluation, which makes the procedure much faster than other tuning strategies that use, for example, 5-fold CV. The tuning results show that the best values are 11 for m, 2 for node size and 1 for sample fraction. Moreover better results are obtained considering samples with repetitions.

To compare the models the estimated and real errors are considered. They are obtained training two models only on the learning set: one with and the other without an external 5-fold CV. When this re-sampling technique is not applied the OOB estimate of the error is provided. The results are reported in (1) and compared with the ones obtained by other models[3].

As last step,a RF is learned with the optimal parameters deduced from the tuning on all the data and the OOB estimate obtained is provided.

4.3 Results and discussion

The table (1) shows the errors obtained[4] in the first of the two steps discussed in 3. It can be observed, also by looking at (1), that the RF outperforms the other models. In addition, an OOB estimate of 0.22 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. This said, the model learned on all the data should still perform better than models trained on only a portion of the data.

Table 1: comparison of the estimated and real errors between the models. (A) and (B) are obtained respectively with and without CV.

Error	Estimated (A)	Real (A)	Estimated (B)	Real (B)
Single tree	0.61	0.61	-	-
Bag of trees	0.27	0.28	0.26	0.28
RF	0.26	0.26	0.24	0.26

Overall, the solution found works well but not extremely well, since it is expected to fail about 1 out of 4 times. Moreover, changing the seed can result in a better or worse performance by looking at the same indices.

As a comparison, the same dataset was used in [9] 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 both equal to 0.26. Which are not as good. As a positive note, in this work the CV estimate of the error and the actual error measured on the test set are equal: this gives greater confidence about the result obtained. This means that there is greater confidence that the result will be similar on new unseen data.

One possible way to try to improve this work by sticking to the same ML technique could be to generate more samples from the available data using some suitable technique. In this case one should be careful not to introduce undesirable correlations between features.

[3] 500 unpruned trees are learned for the bag of trees and an optimal single tree is found by pruning a fully grown one using a cost complexity parameter.

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

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

[3] Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27, 05 2017.

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

[5] Segal Mark. Machine learning benchmarks and random forest regression. *UCSF: Center for Bioinformatics and Molecular Biostatistics.*, 2004.

[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] Bischl B. Probst P. and Boulesteix A. Tunability: Importance of hyperparameters of machine learning algorithms, 2018.

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

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

[10] Boulesteix A.. Zeileis A. Strobl C. and Hothorn T. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8, January 2007.

[11] 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.