Introduction to Machine Learning project: Leaf identification

Lorenzo Basile[1], Roberto Corti[2], and Arianna Tasciotti[3]

[1,2,3] problem statement, solution design, solution development, writing

Course of AA 2019-2020

1 Problem statement

Leaf identification is the process of matching an unknown observed leaf to its proper scientific name. Modern automatic recognition systems can provide the experts of this field with a fast and efficient classification tool.

In this project our aim is to use several supervised machine learning techniques to develop a leaf classifier.
The input of this problem is given by 14 morphological and textural features of a picture of a leaf and the output is a number associated to its species. The classifiers that we present have been trained on an already preprocessed dataset provided by [1] that comprises 40 different plant species.

2 Assessment and performance indexes

In order to assess how well the classifier would work if applied to unseen data, we used a 5-fold cross validation (CV) making sure that each class is represented in each fold. We trained different classification models that have been compared using as performance indexes the Accuracy and, for each class, the False Positive Rate (FPR) and the False Negative Rate (FNR).

3 Proposed solution

The leaf identification problem is a multiclass classification problem with numerical predictors. We focused on two classes of methods: tree based methods and support vector machines (SVM), implemented using `R` language. Regarding tree based methods, we relied on a simple decision tree and then tried to improve its performance by using tree aggregation methods, such as random forest. This classifier aggregates independent views by making use of multiple

decision trees.
SVM are another very powerful machine learning tool, natively built for binary classification: to adapt them to our multiclass task we opted for the *one versus one* approach in which all classes are compared pairwise and the final prediction is the most frequently predicted class. We chose three different kernels (linear, radial and polynomial) to build our SVM and tuned their hyperparameters.
Once we trained these classifiers, we compared them by means of the performance indexes presented in Section 2.

4 Experimental evaluation

4.1 Data

The dataset used for solving this leaf identification problem is made up of 340 leaf observations of the following 16 features:

1. Class (Species)
2. Specimen Number
3. Eccentricity
4. Aspect Ratio
5. Elongation
6. Solidity
7. Stochastic Convexity
8. Isoperimetric Factor
9. Max. Indentation Depth
10. Lobedness
11. Average Intensity
12. Average Contrast
13. Smoothness
14. Third moment
15. Uniformity
16. Entropy

As explained in [1], the variable *Class* assigns each leaf of the dataset to one of 40 different species encoded as a number. However, we noticed that of the 40 Species found in the original dataset only 30 (the ones whose leaves are defined as *simple* in [1]) are present. Thus, our models will not be able to classify these missing species.
Concerning the other attributes, from 3 to 10 they indicate properties that characterize the shape of the leaf, while from 11 to 16 they refer to texture peculiarities of the image. A full and precise explanation of these variables can be found in [1]. The attribute *Specimen Number* represents the number of leaf specimens available by species. Hence we thought that this variable was not a useful predictor for our classifiers and we decided to discard it during the preprocessing phase.
In order to verify the degree of correlation among the predictors, we produced the correlation matrix and we observed that some variables are highly correlated. We compared the test performance of our classifiers with all predictors and without some redundant ones.

4.2 Procedure

In order to tune the hyperparameters and measure the test performance of the methods presented in Section 3, we performed two nested 5-fold CV loops.

In particular, we divided the dataset into 5 folds ensuring that each class is represented in each fold. Then at each iteration of the outer CV, we assigned one fold to test set and the others to training and we tuned the hyperparameters with an inner 5-fold CV on the corresponding training set. After that, we evaluated the performance indexes of our tuned models on the unseen test fold (20% of the original dataset). Finally, we averaged on the values of Accuracy, FPR and FNR collected at each iteration of the outer CV.
Regarding decision tree, we tuned its maximum depth, while for random forest we tuned the parameter m, which represents the number of features taken into account by each tree, keeping in mind the widely accepted golden standard of fixing it to $\sqrt{p}$ (as stated in [2]) and not allowing it to deviate much from this value.
When it comes to SVM, we tuned the regularization parameter for all kernels, the degree and scale factor for polynomial kernel and the scaling parameter σ for radial kernel.

4.3 Results and discussion

As stated in Section 4.1, we trained the classifiers using all predictors and then without 3 redundant variables: *Smoothness*, *Average Contrast* and *Maximal Indentation Depth*.
We present the results in terms of Accuracy of the models with all predictors in table 1.

	Accuracy
Decision Tree	0.51
Random Forest	0.78
SVM (radial kernel)	0.76
SVM (linear kernel)	0.76
SVM (polynomial kernel)	0.75

Table 1: Accuracy results with all predictors.

As we expected, decision tree has by far the worst Accuracy; tree aggregation methods like random forest produce instead a quite reliable classifier with 0.78 Accuracy.
Even though SVM takes a completely different approach from random forest to the classification problem, not based on decision trees but on separating hyperplanes, it yields similar results in terms of Accuracy. Moreover, we observed that the kernel choice does not impact too much on performance.
In table 2, we report the Accuracy obtained training the classifiers without the 3 predictors previously mentioned.

	Accuracy
Decision Tree	0.51
Random Forest	0.78
SVM (radial kernel)	0.74
SVM (linear kernel)	0.78
SVM (polynomial kernel)	0.74

Table 2: Accuracy results without 3 predictors.

As observed, removing these predictors does not affect the results significantly; however, since for the most accurate models (random forest and SVM with linear kernel) we noticed that the Accuracy increases or stays the same, we think that this design choice makes sense because it reduces the computational cost while ensuring at least the same Accuracy.
Finally, we present in tables 3 and 4 the FPR and the FNR of each class for random forest and SVM with linear kernel trained without the 3 predictors.

Species	FPR	FNR		Species	FPR	FNR
1	0.017	0.167		22	0.008	0.133
2	0.020	0.200		23	0.000	0.067
3	0.017	0.300		24	0.020	0.333
4	0.020	0.500		25	0.004	0.300
5	0.000	0.067		26	0.023	0.700
6	0.004	0.000		27	0.007	0.367
7	0.011	0.300		28	0.023	0.400
8	0.000	0.000		29	0.003	0.067
9	0.024	0.233		30	0.000	0.100
10	0.004	0.233		31	0.000	0.200
11	0.000	0.000		32	0.015	0.367
12	0.012	0.367		33	0.007	0.167
13	0.013	0.333		34	0.008	0.000
14	0.024	0.533		35	0.008	0.367
15	0.000	0.000		36	0.000	0.100

Table 3: FPR and FNR for each class obtained with SVM linear kernel classifier.

Species	FPR	FNR		Species	FPR	FNR
1	0.010	0.267		22	0.017	0.367
2	0.021	0.600		23	0.000	0.100
3	0.004	0.400		24	0.031	0.367
4	0.005	0.700		25	0.000	0.100
5	0.005	0.000		26	0.015	0.233
6	0.000	0.000		27	0.012	0.233
7	0.015	0.200		28	0.018	0.300
8	0.000	0.000		29	0.003	0.067
9	0.022	0.300		30	0.009	0.100
10	0.013	0.067		31	0.005	0.100
11	0.003	0.000		32	0.026	0.533
12	0.009	0.300		33	0.010	0.533
13	0.016	0.333		34	0.004	0.100
14	0.011	0.267		35	0.007	0.267
15	0.000	0.000		36	0.000	0.100

Table 4: FPR and FNR for each class obtained with random forest classifier.

We can see from FNR values that both classifiers do not perform well on class 4; this may be due to the fact that there are only 8 observations for this species. Moreover, we notice that there is some serious imbalance between the two classifiers regarding to FNR of some specific classes. SVM is more sensitive than random forest on classes 2 and 33. On the other hand, random forest is more sensitive than SVM on classes 14 and 26.
To conclude, random forest and SVM with linear kernel are the best solutions among the tested classifiers. There is no general reason to prefer one over the other since Accuracy is the same but the choice of the model may be guided by the necessity to classify some specific classes with high sensitivity.
Better results could be achieved by increasing the number of observed leaves in each class, especially for those that have very few observations.

References

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

[2] Jerome Friedman, Trevor Hastie, Robert Tibshirani. *The Elements of Statistical Learning*, chapter 15. Springer, 2009.