Introduction to Machine Learning and Evolutionary Robotics: chosen-project-Leaf identification

Isac Pasianotto[1] and Davide Rossi[2]

[1] problem statement, solution design, solution development, data gathering, writing
[2] problem statement, solution design, solution development, data gathering, writing

Course of AA 2022-2023 - Data Science and Scientific Computing

1 Problem statement

Our goal is to propose a method for leaf identification based on the provided leaf attributes. More formally, we want to build a function $f_{predict} : X \to Y$ where: $Y = \{y : y \text{ is a leaf species}\}$ and $X = \{x : x \text{ is a leaf}\}$. To do that we dispose of a dataset $\mathcal{D} = \{x'^{(i)}, y^{(i)}\}_{i=1}^{n_x}$. Remark: we are using $x'^{(i)} \in X' \subseteq \mathbb{R}^p$. That is due to the fact that $x \in X$, being a leaf, is not processable by a machine. In our case the needed pre-processing phase $f_{preproc} : X \to X'$ has already been done.

2 Assessment and performance indexes

To assess our classification problem we will focus mostly on effectiveness rather than efficiency.

To assess the learning technique we have chosen the LOOCV since it has the best effectiveness and (as we will see later) the dataset we will work on is small enough to avoid worrying about the poor efficiency.

At each iteration of the LOOCV we will compute a performance index to assess how the model learned by f_{learn} fits the data. The available indexes we know for classification problems are: accuracy, error and weighted accuracy, but due to the fact that we're using LOOCV, the weighted accuracy can not be computed since, at each iteration, all the class accuracies but one will be $\frac{0}{0}$ (remembering that wAcc $= \frac{1}{|Y|} \sum_{y \in Y} \left(\frac{\sum_i \mathbf{1}\left(y^{(i)}=y \land y^{(i)}=\widehat{y}^{(i)}\right)}{\sum_i \mathbf{1}\left(y^{(i)}=y\right)} \right)$), hence we will only consider the accuracy performance index.

3 Proposed solution

To achieve the given goal, we will try two different techniques: the first one, according to Occam's razor, is a single classification tree, used to compute a baseline accuracy. The other one, which we expect to perform better, is the Random Forest learning technique.

To evaluate these techniques, as we said in the previous chapter, we will use the LOOCV method and the accuracy performance index: [1]

```
function single_accuracy (f_predict, m, {x'^(i), y^(i)} ):
    if f_predict(x'^(i), m) = y^(i): return 1
    else: return 0
function assess_LOOCV (D):                              //D ↔ {(x', y)}
    for j ∈ 1, ..., n_x:
        D_test ← j^th row of D
        D_learn ← D \ D_test
        m_tree ← learn_tree(D_learn)
        m_forest ← learn_forest(D_learn)
        eff_j^tree ← single_accuracy(predict_tree, m_tree, D_test)
        eff_j^forest ← single_accuracy(predict_forest, m_forest, D_test)
    eff_μ^tree ← (1/n_x) Σ_j eff_j^tree
    eff_σ^tree ← sqrt( (1/n_x) Σ_j (eff_j^tree − eff_μ^tree)^2 )
    eff_μ^forest ← (1/n_x) Σ_j eff_j^forest
    eff_σ^forest ← sqrt( (1/n_x) Σ_j (eff_j^forest − eff_μ^forest)^2 )
    return eff_μ^tree, eff_σ^tree, eff_μ^forest, eff_σ^forest
```

where learn_ψ and predict_ψ are the learning and predicting functions for a learning technique $\psi \in \{\text{tree}, \text{Random Forest}\}$.

If Random Forest will be assessed enough appropriate we will keep it, otherwise we'll consider other learning techniques.

4 Experimental evaluation

4.1 Data

The dataset we will use is the `leaf` dataset, that is publicly available [1] and comes with an exhaustive data description.

Note that the dataset is well balanced: the frequency is more or less the same for all the species:

[1] Note that the function `single_accuracy(···)` is defined in this way since, using LOOCV, the number of rows in $\mathcal{D}_{test}$ is 1, hence the accuracy at each iteration is a binary value.

Species	1	2	3	4	5	6	7	8	9	10
Frequency	12	10	10	8	12	8	10	11	14	13
Species	11	12	13	14	15	22	23	24	25	26
Frequency	16	12	13	12	10	12	11	13	9	12
Species	27	28	29	30	31	32	33	34	35	36
Frequency	11	12	12	12	11	11	11	11	11	10

4.2 Procedure

We are going to use the R software and the `tree` and `randomForest` packages to learn our models.

Regarding trees, as parameter for the size of a single node we will first use the `tree` package default value $\mathrm{n_{min}}$=10, to have a first basic idea of the accuracy, then we'll use $\mathrm{n_{min}}$=1 , in order to perfectly fit the data (in both cases, we will set the parameter mindev=0).

For the number of trees and variables for Random Forest we will use the commonly accepted as default values: $n_{tree} = 500$ and $n_{vars} = \lceil \sqrt{p} \rceil$, where p is the number of available independent variables. For each learned tree we will use $n_{min} = 1$, which is the default value for classification trees in `randomForest` package.

For the assessment part with the LOOCV we are going to repeat the learn and predict phase for $n_x = 340$ times and saving the accuracies of each iteration in two vectors of boolean values that will be used to compute the average accuracies $\mathrm{Eff}_\mu^{tree}, \mathrm{Eff}_\mu^{forest}$ and their standard errors $\mathrm{Eff}_\sigma^{tree}, \mathrm{Eff}_\sigma^{forest}$.

If one the proposed learning techniques proves to be sufficiently satisfying, we will use it to try a "real" prediction by splitting the dataset $\mathcal{D}$ into $\mathcal{D}_{learn}$ and $\mathcal{D}_{test}$ with $|\mathcal{D}_{learn}| \approx 70\%|\mathcal{D}|, \mathcal{D}_{test} = \mathcal{D} \setminus \mathcal{D}_{learn}$, and use the learned model to predict the leaves species in $\mathcal{D}_{test}$. Eventually we will check the quality of prediction by constructing the confusion matrix (done with the `caret` package) and computing the values of accuracy, error and (if possible) weighted accuracy.

4.3 Results and discussion

By applying the leave one out cross validation we got the following results:

Technique ψ	Eff_μ^ψ	Eff_σ^ψ
Classification tree ($\mathrm{n_{min}}$=10)	0.6045	0.4896
Classification tree ($\mathrm{n_{min}}$=1)	0.6500	0.4777
Random Forest	0.7765	0.4177

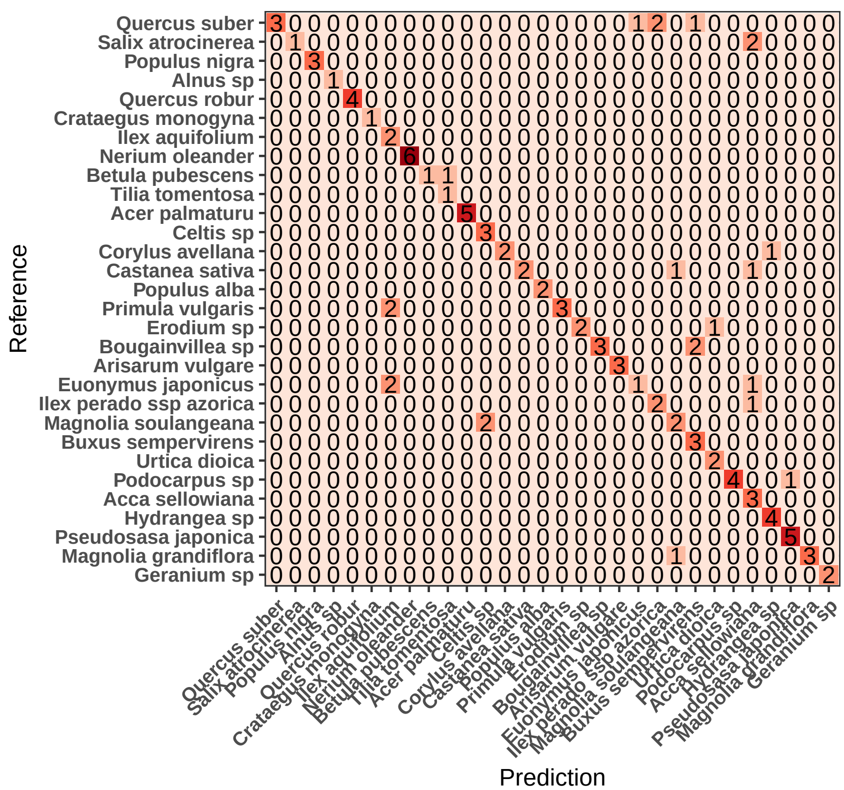
As expected (remember the Wisdom of the Crowds principle), Random Forest outperforms classification trees.

We were quite satisfied with the results obtained with Random Forest, so we tried to compute a random forest model and checked the quality of the outcomes in the prediction phase. We randomly split the dataset and applied the learning technique we just built to predict some values, then we computed some performance indexes, and the results were the following:

index	value
Accuracy	0.7745
Error	0.2255
Weighted accuracy	0.8043

These are quite satisfying values for the indexes, considering that the dataset is very small: in fact, it consists of just 340 observations, so we have used about 240 of them for the learning phase. Notice that, since the dataset is quite well balanced, the values for accuracy and weighted accuracy are quite close.

Let's give a look at the confusion matrix:



As we can see, most of the values are located along the matrix diagonal, meaning they are correct predictions.

Note that, because of the small size of $\mathcal{D}$, there's quite high variability between results for a single model (in our repeated trials the prediction accuracy went from $\approx 68\%$ to $\approx 82\%$, depending on how the dataset was randomly split).

References

[1] Pedro F. B. Silva, André R. S. Marçal, and Rubim Almeida da Silva. leaf dataset. https://archive.ics.uci.edu/ml/machine-learning-databases/00288/.