Introduction to Machine Learning and Evolutionary Robotics Project: Leaf identification

Stefano Simonetto[1], Lorenzo Bonin[2], and Ionuţ Alexandru Pascariu[3]

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

Course of AA 2020-2021

1 Problem statement

Before the development of machine learning techniques, leaf identification has always been done by experts of this specific field, capable of deciding the species of a given leaf on the grounds of their knowledge. Nowadays, they can be supported in this task by automatic leaf recognition systems, which provide extremely useful tools also for non-professionals. The aim of this project is to find the best among some classifiers which, given some features related to the shape and the texture of a leaf displayed in an image, are able to return the name of the species of the leaf. Hence, the idea is to compare different models according to some performance indexes and decide which one suits the leaf identification problem best.

2 Assessment and performance indexes

It was decided to assess the performances of the different classifiers by means of effectiveness, by using the following indexes: Accuracy, False Positive Rate (FPR) and False Negative Rate (FNR). In order to dispose of a sufficient amount of unseen data to compute these values it was decided to split the dataset in two parts, as described in the next sections.

The techniques to be firstly selected are the ones with the best accuracy, considering both mean and standard deviation. In a second step they are compared by means of the FNR and FPR, which are calculated for each class, setting the threshold value to 0,5. Therefore, ”posive” means ”the leaf belongs to that specific class”, while ”negative” means ”the leaf doesn’t belong to that specific class”.

3 Proposed solution

The solution has been implemented in R. The learning techniques that have been compared are the following: decision tree, random forest and support vector machines (SVM) with polynomial and radial kernel. Regarding SVM, since leaf identification is a multiclass classification problem, it was decided to use the default approach of the *svm* function of the *e1071* R package, which is simply the one-versus-one classification. Our workflow has been the following:

- Firstly it was decided to split the data in 2 parts, the training set and the validation set.

- For each learning technique we obtained the top K_{out} parameter settings, by adapting the **nested K-fold cross validation** procedure illustrated by the algorithm 1 in [1], as expressed by the following pseudocode:

```
1. Define the parameters that have to be optimized, and, for each of them, specify the set of values to be checked in the optimization procedure
2. Divide the training data into $K_{out}$ folds randomly
3. (outer loop) for each fold $k_{out-i}$ in the $K_{out}$ folds:
   (a) Let fold $k_{out-i}$ be the $test_{out}$ set and the remaining $K_{out}$-1 folds be the $train_{out}$ set
   (b) Divide the $train_{out}$ set into $K_{in}$ folds randomly
   (c) (inner loop) for each $k_{in-j}$ in the $K_{in}$ folds
       i. Let the $k_{in-j}$ fold be the $test_{in}$ set and the remaining $K_{in}$-1 folds be the $train_{in}$ set
       ii. For each parameter setting, train on $train_{in}$ and evaluate the model on $test_{in}$; store the accuracy of each parameter setting
   (d) For each parameter setting, calculate the mean accuracy over the $K_{in}$ folds, and choose the best one. Store it for fold $k_{out-i}$.
4. For each of the top $K_{out}$ parameter settings, train the model on the whole training data, predict on the validation, compute the confusion matrix and the accuracy. Then select the best model in terms of accuracy and store its confusion matrix.
5. Compute the mean and the standard deviation of the accuracy values
```

- Once that the best model has been selected for each technique under consideration, we proceeded by computing the FNR and FPR values of each class for each of them. Finally, the best model, hence the technique, is selected as described in the paragraph 2.

4 Experimental evaluation

4.1 Data

The data set used is made up of 340 leaf observations of the 16 attributes which are exhaustively described in [2]. The Class attribute is the one which identifies the leaf species, so it was decided it to be the dependent variable. The remaining attributes are the independent variables. The data set contains 30 different classes, while [2] presents a total of 40 classes. Hence, the classes which are excluded from the actual data set (the ones numbered from 16 to 21 and from 37 to 40) cannot be predicted. It was decided to remove the *Specimen Number* attribute as it has been found to be unnecessary for our purpose.

4.2 Procedure

We implemented the procedure described in paragraph 3, deciding to split the data set in this way: 80% as training set and 20% as validation set. The values chosen for the outer and the inner loops of the nested K-fold cross validation procedure are K_{out}=5 and K_{in}=3. Exaggeratedly large values would prevent the folds from considering at least one instance for each of the 30 classes. These values have proven to be quite good for our specific problem. The parameters that have been optimized are the following:

- Regarding the Decision Tree, it was decided to tune the Complexity Parameter (**cp**), which has been evaluated on a set of 20 values from 0.01 to 0.2.
- Concerning Random Forest, we opted for the **number of variables randomly sampled as candidates at each split (mtry)** $\in \{2, 3, 4, 5, 6\}$ (values selected in order to avoid to be too far from the value $\sqrt{14}$) and the **number of trees (ntree)** $\in \{250, 500, 750, 1000, 1250, 1500\}$.
- Regarding SVM, we optimized γ, the **cost** and the **degree** (for polynomial kernel only). In our specific case, the default value for γ was 1/14, where 14 represents the number of independent variables, so it was decided to multiply it by an arbitrarily chosen set of **x** values. The intervals are the following:
  - **x** $\in \{ 2^{-4}, 2^{-3}, ..., 2^{4}, 2^{5}\}$
  - **cost** $\in \{ 2^{-8}, 2^{-7}, ..., 2^{7}, 2^{8} \}$
  - **degree** $\in \{1, 2, ..., 8\}$

All the values to be checked have been arbitrarily chosen, being careful to include all the default values of the R packages' functions that have been used. Other combinations were also implemented and the values were chosen differently, but the results obtained were not too different from those that are presented in the next paragraph.

4.3 Results and discussion

In table 1, it was decided to show -for each model- the mean and the standard deviation of the accuracy that was obtained from the point 5 of the procedure described in the section 3, the best parameters and the accuracy to which they led. Taking into account the table 1, it seems that SVM with radial kernel is better, as it has an higher average accuracy and a lower standard deviation than the other techniques. By the way, there is not such a huge difference with the SVM with the polynomial kernel. Decision tree has a way lower accuracy, so it has not been considered in the following operations. Therefore FPR and FNR have been computed only for the other three models.

ML Algorithm	$\overline{acc}_{final}$	σ_{final}	cp$_{best}$	ntree$_{best}$	mtry$_{best}$	γ_{best}	degree$_{best}$	cost$_{best}$	acc$_{best}$
Decision Tree	0.55	0	0.02	/	/	/	/	/	0.55
Random Forest	0.79	0.0084	/	250	4	/	/	/	0.8
SVM radial	0.90	0.013	/	/	/	0.018	/	256	0.91
SVM polynomial	0.88	0.020	/	/	/	0.0089	1	256	0.91

Table 1: Mean and Standard deviation of the accuracy and the best parameters

Considering table 2, it can be noticed that Random Forest has 16 classes which have pairs (FPR,FNR) equal to (0,0), while both SVM models have 20. Moreover, Random Forest has two classes with FNR rate equal to one. Thanks to these information it was chosen to consider only SVM with radial kernel and SVM with polynomial kernel. According to the table 2, it can be seen that SVM with polynomial kernel has two classes with a greater FPR rate and only one with a greater FNR rate than the model with radial kernel. On the other hand, SVM with radial kernel has three classes with greater FPR rate and one class with FNR rate bigger than the model with polynomial kernel. Therefore, according with the table 2, it seems that SVM with polynomial kernel is slightly better than SVM with radial kernel, since the latter model has an extra class with a higher FPR rate. In conclusion, we can state that the two SVM models are almost equivalent, therefore they can both be considered the best solution for the leaf identification problem.

Class	R. Forest		SVM-radial		SVM-poly.	
	FPR	FNR	FPR	FNR	FPR	FNR
1	0	0	0	0	0	0
2	0.038	0.5	0.0172	0	0.0172	0
3	0	0	0	0	0	0
4	0	1	0	0	0	0
5	0.020	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	1	0.0172	0.33	0.0172	0.33
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0.020	0	0	0	0	0
13	0.020	0.33	0.0175	0	0.0172	0.33
14	0	0	0.0172	0	0	0
15	0	0	0	0	0	0

Class	R. Forest		SVM-radial		SVM-poly.	
	FPR	FNR	FPR	FNR	FPR	FNR
22	0	1	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0.33	0	0
25	0	0	0	0.5	0	0.5
26	0.020	0	0	0	0	0
27	0	0.5	0	0.5	0	0.5
28	0	0.5	0	0.5	0	0.5
29	0	0	0	0	0	0
30	0.074	0	0.034	0	0.0172	0
31	0	0	0	0	0	0
32	0.019	1	0	0.5	0	0.5
33	0.020	0	0	0	0.0172	0
34	0	0	0	0	0	0
35	0.020	0	0	0	0.0172	0
36	0	0	0	0	0	0

Table 2: FPR and FNR for each model except for Decision Tree

References

[1] CASPER HANSEN. *Nested Cross-Validation Python Code.* URL: `https://mlfromscratch.com/nested-cross-validation-python-code/#/`.

[2] Pedro Filipe Barros Silva. *Development of a System for Automatic Plant Species Recognition.* URL: `https://repositorio-aberto.up.pt/handle/10216/67734`.