Introduction to Machine Learning project: leaf identification

Eros Fabrici

Course of AA 2019-2020

1 Problem statement

Leaf identification, and more in general plant recognition, has traditionally been done by specialised professionals. The development of computer technologies has increased the process of automation, leading traditional methods to be substituted by novel methodologies. Therefore, a development of an automatic leaf recognition system could provide an inexpensive tool able to help in classifying leafs both for professionals and non-professionals.

The aim of this project is to compare some leaf classifiers, obtained by using different machine learning models, and propose the most effective in classifying different leaf species (see section 4.1 for further details about the data) according to some performance indexes, which will be described in the next section.

2 Assessment and performance indexes

In order to evaluate and compare the effectiveness of the classifiers, the following performance indexes will be taken into account:

- *Accuracy*: the ratio of correct predictions to the total number of predictions.
- *False Positive Rate* (**FPR**) and *False Negative Rate* (**FNR**). Usually these indexes are used in binary classification, but they can be also used for multi classification, where for each class a pair of indexes, FRP and FNR, are calculated.

The classifiers that achieve the best accuracy (both in terms of mean and standard deviation) will be selected on the first phase, then they will be chosen by confronting each class' FPR and FNR index. In this type of problem, FNR and FPR will have the same weight. Therefore the aim is to minimise both, rather than focusing on minimising a single index, such as in the problem of diagnosing a disease in which it should be better having a low FNR than FPR.

3 Proposed solution

The solution proposed experiments with the following machine learning models: *decision tree*, *random forest* and *support vector machines* with *polynomial* and *radial* kernel.

The general idea is to split at random the data set in two parts: one part (`main_training`) will be used for tuning and validating the parameters of each learning technique, while the aim of the other part (`final_validation`) is to validate the tuned parameters and to analyse the FPR and FNR of each class between the different models. In order to perform this, the solution is to use the *Nested K-fold Cross Validation* [1]. Following the algorithm.

1. For each learning technique do the following:
   (a) Divide `main_training` in K cross-validation folds at random.
   (b) For each $k = 1, 2, ..., K$ do:
      i. Let `train` be all the data excepts for the k-th fold.
      ii. Let `test` be the k-th fold.
      iii. Divide `train` into M folds.
      iv. For each $m = 1, 2, ..., M$ fold do:
         A. Let `val` be the m-th fold.
         B. Let `t` be all the data of the remaining $m - 1$ folds.
         C. Train with each parameter value (over a defined finite set of values) on `t` and evaluate it on `val`. Keep track of the mean accuracy for each parameter value.
      v. Learn a classifier on `train` with the parameter that achieved the best mean accuracy, test it on `test` and store the accuracy.
   (c) Calculate the mean and the standard deviation of the accuracy values over the K folds.
   (d) For each of the parameter values that achieved the highest mean accuracy do:
      i. Learn a classifier using on `main_train`.
      ii. Predict on `final_validation` and keep track of the accuracy and confusion matrices.
   (e) Compute mean and standard deviation of the accuracy values just obtained.
2. Select the models which performed better in terms of accuracy. For each of them, compute the FPR and FNR of each class.
3. Select the best learning technique according to the metrics described in section 2, together with its tuned parameters.

4 Experimental evaluation

4.1 Data

The data used contains 340 observations with 16 attributes, one representing the Class (leaf species), one the Specimen Number, while the remaining attributes, which are all numerical, describe the leaf shape and texture. In our case the dependent variable is the Class, which identifies a leaf species, while all the others are independent variables. The data set considers a total of 30 classes. For a full and detailed description of the data set see [2].

4.2 Procedure

The first phase consisted in cleaning the data. The attribute *Specimen Number* was not an useful predictor, as the name suggests. A test of the variable importance was made and confirmed the assumption. As a consequence, the attribute Specimen Number was discarded. After this the data set was split at random in two parts, labelled in section 3 as `main_training` and `final_validation`, the first composed by the 75% of the original data while the second by the remaining 25% observations. Such a ratio was chosen in order to guarantee that more than one observation of each class would appear in the `final_validation` set. Subsequently, the algorithm described in section 3 was applied with $K = 5$ and $L = 4$. The reason for choosing big folds was, as mentioned before, the high number of classes to be predicted. With a 10-fold CV it would not be possible to capture at least one observation for each class in a single fold with the data available.

For the *Decision Tree* the values of the *Complexity Parameter* (cp) tested goes from 0.01 to 0.1 (10 values). Higher values would for sure lead to overfitting.

As regards the `Random Forest`, the parameters tuned are the *number of trees* (B) and the *number of randomly selected predictors* (m), with $B \in \{500, 1000, 1500, 2000, 2500\}$ and $m \in \{1, 2, 3, ..., 14\}$ (note that when $m = 14$ the technique is actually *Tree Bagging*).

Finally, the *SVM* parameters chosen are:

- $\gamma \in \{(1/dimX)^a |$ radial: $a \in \{-5, -4, ..., 5\}$ polynomial: $a \in \{-2, -1, ..., 2\}\}$ where $dimX$ is the dimension of X, namely the number of independent variables (14).
- $degree \in \{1, 2, ..., 10\}$ (for polynomial kernel only).
- $cost \in \{10^{-2}, 10^{-1}, 1, 2, ..., 10\}$ (for radial kernel only).

The set of γ values for the polynomial kernel is smaller due to pure technical reasons, because R was halting the execution of the program as it was reaching the maximum number of iterations. Nonetheless, the interval of values of the parameters were chosen arbitrarily (however, they include the default values suggested by R documentation).

ML model	μ_{CV}	σ_{CV}	$\mu_{finalVal}$	$\sigma_{finalVal}$
Decision Tree	0.34	0.08	0.44	0
Random Forest	0.73	0.04	0.80	0.01
SVM poly. k.	0.70	0.06	0.80	0.05
SVM radial k.	0.72	0.03	0.79	0.01

Table 1: Mean and standard deviation of the accuracy obtained

Class	Random Forest		SVM randial kernel	
	FPR	FNR	FPR	FNR
1	0.015	0.333	0.015	0.333
2	0.028	0	0.044	0
3	0.014	0.5	0.015	0.5
4	0	0	0	0.5
5	0	0	0	0
6	0	0	0	0
7	0.029	0	0.015	0
8	0	0	0	0
9	0.029	0.5	0	0.5
10	0	0.333	0	0
11	0	0	0	0
12	0	0	0	0
13	0.029	0.333	0.044	0.333
14	0.015	0	0.015	0
15	0	0	0	0

Class	Random Forest		SVM randial kernel	
	FPR	FNR	FPR	FNR
22	0	0	0	0
23	0	0	0	0
24	0	0.333	0.015	0.333
25	0	0	0.015	0.5
26	0	1	0	1
27	0.015	0.333	0	0.333
28	0	0.333	0.015	0.333
29	0	0	0	0
30	0	0	0	0
31	0	0	0	0.333
32	0.029	0.667	0.029	0.667
33	0.015	0.333	0.015	0.333
34	0	0	0.015	0
35	0	0	0	0
36	0	0	0	0

Table 2: FPR and FNR: RF vs SVM radial k.

4.3 Results and discussion

Table 1 shows clearly that *Decision Tree* is not a suitable technique for this problem, thus it was excluded. Due to the fact that SVM with polynomial kernel had an higher standard deviation, it was excluded as well. Therefore, the final assessment was between *Random Forest* and *SVM* with radial kernel.

In the final validation, RF had the highest prediction accuracy (0.82) with $B = 500$ and $m = 3$, while SVM (0.80) with $\gamma = 1/14$ and $cost = 6$. Table 2 shows the FPR and FNR of of these two predictions.

It is possible to infer that SVM performed worst in both FPR and FNR. In fact, SVM had six classes (2, 3, 13, 24, 25, 28 and 34) with higher FPR than the corresponding ones in RF model, while the latter had only three classes (7, 9 and 27) with higher FPR. As regards FNR, SVM had three classes (3, 25, 31) with higher rate, while RF had only one class (10) with higher FNR. Finally, according to the performance indexes chosen and explained in section 2, the best model is the RF.

References

[1] Nested k-fold cross validation. `https://weina.me/nested-cross-validation/`.

[2] Pedro F. B. Silva. Development of a system for automatic plant species recognition. 2013.