Introduction to Machine Learning and Evolutionary Robotics project: Leaf identification

Mauro Farina

Course of AA 2021-2022 - Ingegneria Elettronica e Informatica

1 Problem statement

The goal is to propose a method for leaf identification. Specifically, we want to train a model to identify the plant species based on an input of 14 real-valued attributes derived from the plant's leaf: 8 shape-related features and 6 texture-related features.

2 Assessment and performance indexes

To assess the models we use the prediction accuracy, defined as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \in [0, 1] \tag{1}$$

and evaluated on a subset of observations that are unseen by the model.

In order to gather more insight into how challenging it is to predict each class, we also assess the class-specific prediction accuracy using the confusion matrix:

$$\text{Accuracy}_i = \frac{\text{Confusion}_{i,i}}{\sum_j \text{Confusion}_{i,j}} \in [0, 1] \tag{2}$$

where the confusion matrix is structured with the true classes represented in the rows and the predicted classes in the columns.

3 Proposed solution

In pursuit of our goal, we first learn a simple Decision Tree (DT) model as a baseline. We then train further models based on two of the most popular general-purpose supervised Machine Learning techniques [1]: Random Forest (RF) and Support Vector Machine (SVM).

In order to properly evaluate the performance achieved by a learning technique, we first need to fine-tune the hyperparameters (if any), then train a model using the optimal hyperparameters (the ones that maximize the average effectiveness index) and measure its accuracy. The usual approach is to apply a nested cross-validation procedure, which simultaneously returns the optimal hyperparameters and the average accuracy. However, we use a single k-Fold Cross-Validation (k-CV) procedure since there is strong evidence that, for most practical purposes, it does not incur in worse selections [3].

Random Forest has three parameters: n_{trees} (the number of trees), n_{vars} (the number of independent variables randomly chosen when learning a single tree), and *nodesize* (minimum size of terminal nodes). Since it has been shown experimentally that neither n_{trees} nor n_{vars} relate to overfitting, we use their "default" values.

On the other hand, the SVM technique has a few parameters that need to be fine-tuned:

- kernel: linear, polynomial, gaussian
- cost (c): the misclassification tolerance budget on learning data
- gamma (γ): determines how fast $e^{-\gamma||x-x'||^2}$ goes to 0 with distance
- degree (d): degree of the polynomial kernel
- multiclass strategy: method of prediction in multiclass problems (one-vs-one or one-vs-all)

4 Experimental evaluation

4.1 Data

The dataset used for training and assessment of the models is the 'Leaf' dataset [2], a collection of shape and texture features extracted from digital images of leaf specimens publicly available under the CC BY 4.0 license. It is composed of 430 instances, each characterized by a class (an integer associated with the plant species name), a specimen number (an integer that, combined with the class, identifies the specific image from which the features were extracted), and 14 real-valued features. The dataset comes with a detailed description of how each feature is measured and the number of specimen available for each class.

4.2 Procedure

Before learning any model, we preprocessed the data: specifically, we removed the feature Specimen_Number which is not informative as it only refers to the particular specimen used to extract the 14 features. Furthermore, we randomly shuffled the data (according to the $seed = 386$, randomly picked in the interval

$[1, 1000]$) so that each fold of the k-CV procedure provides a representative sample for model training and evaluation. We used $K = 5$ for all k-CV procedures for two reasons:

- the values $K = 5$ and $K = 10$ are widely recognized as popular choices
- the dataset contains 340 observations distributed across 30 different classes: with a value $K = 10$ we would get folds of 34 observations, which is not ideal when testing for a 30-class classification model

In order to measure the class-specific accuracy, we used a running confusion matrix, obtained by summing the confusion matrixes obtained at each fold.

As mentioned before, the model based on Random Forest was trained with "default" classification-problem values for $n_{\text{trees}} = 500$ and $n_{\text{vars}} = \lceil \sqrt{n_{\text{features}}} \rceil = \lceil \sqrt{14} \rceil = 4$, while the hyperparameter-tuning procedure was performed for parameter *nodesize* on the grid $P = (1, 2, 4, 8)$

The grid of hyperparameters used to fine-tune the SVM-based models depends on the kernel:

- linear kernel (SVM-L): $c = (1, 10, 100)$
- polynomial kernel (SVM-P): $c = (1, 10, 100) \times d = (1, 2, 3)$
- gaussian kernel (SVM-G): $c = (1, 10, 100) \times \gamma = (0.01, 0.05, 0.1)$

Regardless of the kernel, all SVM models were trained using the one-vs-one multiclass strategy.

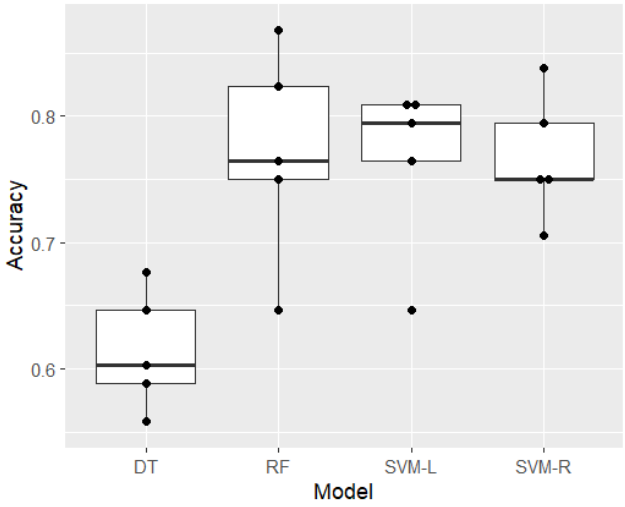

Figure 1: Spread of accuracies achieved during the optimal 5-CV

4.3 Results and discussion

We present the results obtained by implementing the procedure in RStudio using the *randomForest* and *e1071* libraries for model training and prediction assessment.

The optimal computed hyperparameters are $nodesize = 4$ for the RF model, $c = 100$ for all SVM models with $d = 1$ and $\gamma = 0.01$ for the SVM-P and SVM-G models respectively. Since a SVM-P model with $d = 1$ yields the same results as a SVM-L model, we omit it in the discussion.

As expected, the DT model ($\mu_{\text{acc,DT}} = 0.61$) performs worse than the RF ($\mu_{\text{acc,RF}} = 0.77$) and SVM-based models ($\mu_{\text{acc,SVM-L}} = 0.76$, $\mu_{\text{acc,SVM-G}} = 0.77$).

Since the RF model and the SVM models achieve similar average performance, we visualize the five accuracies achieved during the 5-CV with the boxplot in Figure 1, from which we can see that despite achieving a higher average accuracy, the RF model exhibits the largest spread in the accuracy values. Furthermore, it is the SVM-L model that achieves the highest median value, which is a less sensitive measure to outliers than the mean.

From Table 1 we learn that all models struggle to predict correctly the classes '4' and '26', suggesting that the features used to train the models may not be sufficiently representative to distinguish these classes.

Given these results, we conclude that RF and SVM-based models are equally good choices.

Table 1: Class-specific accuracies obtained from the running confusion matrix of the fine-tuned models

Class	RF	SVM-L	SVM-G	Class	RF	SVM-L	SVM-G
1	0.67	0.67	0.83	22	0.67	0.83	0.92
2	0.50	0.60	0.80	23	0.91	0.82	0.73
3	0.60	0.70	0.60	24	0.69	0.77	0.85
4	0.13	0.25	0.38	25	0.56	0.78	0.67
5	1.00	1.00	0.92	26	0.17	0.42	0.42
6	0.88	0.88	0.88	27	0.55	0.64	0.55
7	0.60	0.80	0.80	28	0.58	0.50	0.75
8	1.00	1.00	1.00	29	1.00	0.92	0.92
9	0.71	0.79	0.71	30	0.92	0.92	0.92
10	0.85	0.77	0.69	31	0.64	1.00	0.73
11	1.00	1.00	1.00	32	0.18	0.55	0.55
12	0.67	0.67	0.83	33	0.55	0.64	0.82
13	0.69	0.69	0.77	34	0.91	1.00	0.91
14	0.67	0.58	0.50	35	0.45	0.73	0.55
15	1.00	1.00	1.00	36	0.80	0.90	0.90

References

[1] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.*, 15(1):3133–3181, jan 2014.

[2] Pedro Silva and Andr Maral. Leaf. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C53C78.

[3] Jacques Wainer and Gavin Cawley. Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*, 182:115222, 2021.