Machine Learning and Artificial Intelligence
Homework 2

Debora Caldarola s263626

Prof. Barbara Caputo, A.Y. 2018/2019

1 Introduction

Support Vector Machines (**SVM**s) are a set of supervised learning methods usually applied to detect outliers, for regression and classification algorithms. Their name comes from the use of support vectors in the decision functions.
SVM is memory efficient and gives the possibility of choosing among different Kernel functions as decision functions.
The purpose of this experience is to apply SVMs on a given dataset to map its data points on the belonging category, using both a linear and an RBF (Radial Basis Function) kernel. Different values of regularization parameters will be introduced so that the changes on the accuracy of the model may be observed and the best parameters identified. Moreover, the decision boundaries of the classifier will be analyzed. At the end, K-fold Cross-Validation will be applied to improve the quality of the classification.

1.1 Homework Sub-tasks

1. Train the linear SVM on the training set, show how the accuracy on the validation test varies when changing the regularization parameter C and evaluate the model on the test set. Plot decision boundaries.
2. Repeat using a RBF kernel. Perform a grid search of the best parameters for a RBF kernel and show how they score on the validation test. Evaluate the best parameters on the test set.
3. Repeat the grid search performing a 5-fold Validation. Analyze the difference with the previous case.

2 Data Preparation

The presented results are computed using the ***Iris*** dataset. It consists of 50 samples from each of three species of Iris (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Each sample is described by four features: the length and the width of the sepals and petals, in centimeters. We will select only the first two dimensions, sepal lenght and width.
The data is randomly split into training, validation and test sets in proportion 5:2:3. Using the validation set can help to reduce overfitting on the test set. On the other hand, dividing the dataset the number of samples available for training the model is reduced and the accuracy of the prediction might become really dependant on the random choice of the training and validation sets.

2.1 Programming Environment Setup

The exercise will be solved using Python 3.7 programming language. Moreover, the following libraries and packages will be referenced:

- `numpy`, the fundamental package for scientific computing with Python;
- `matplotlib.pyplot`: `matplotlib` is a Python 2D plotting library; `pyplot` provides a MATLAB-like interface;
- `scikit-learn`, a free software machine learning library for Python. Specific imported packages: `sklearn.datasets`, `sklearn.svm`, `sklearn.model_selection`.

3 Linear SVM

Given two data points x and x', the linear kernel function k is defined as:

$$k(x, x') =< x, x' >$$

The linear kernel is useful if the original data is already high dimensional, like images, and if the original features are individually informative. In such a case, the decision boundary is likely to be representable as a linear combination of the original features so that another feature space is not required.
In our case, the dataset is made of images and so it is not non-linear, and the parameter **C** has to be introduced to regularize the amount of misclassified training points allowed in the model. For large values of C, a smaller-margin hyperplane will be chosen if that hyperplane gets all the training points classified correctly. Too high values of C might cause overfitting. Conversely, for a very small value of C, the optimizer will look for a larger-margin separating hyperplane, therefore for a simpler decision function, even if that choice brings to heavy misclassification.
In this case, the possible values of C lay in the range going from 10^{-3} to 10^3. Those values are used to train a linear SVM on the training set. The obtained decision boundaries are shown in Figure 1. There it is demonstrated how the smallest values of C bring to more fleeting boundaries (i.e., for C=0.001, all the training points belong to the same decision surface); the more C grows the more accurate the boundaries and the classification get, unless overfitting gets in the way.
The method is then evaluated on the validation set. Figure 2 shows how the accuracy varies depending on the value of C: here, the top value 0.85 is reached for C=1. If C is too low the classifier will underfit; at the same time, we can assume that for values of C higher than 10^3 it will overfit, considering how the accuracy starts decreasing.
The best value of C is now used to evaluate the model on the test set. The computed decision boundaries are reported in Figure 3. The obtained accuracy is equal to 0.73 and it is lower than the one got for the validation set.

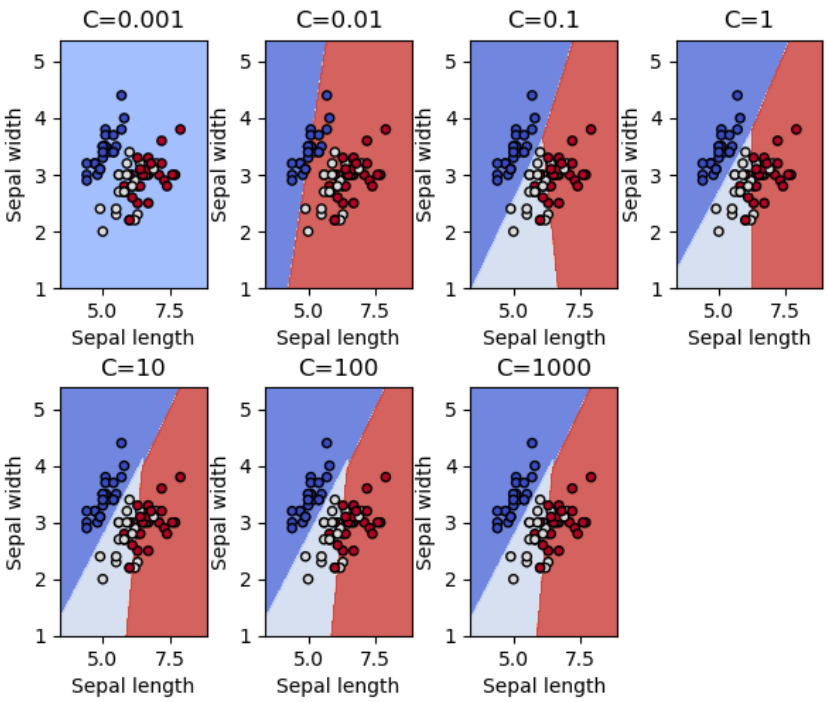

Figure 1: Decision boundaries of linear SVM for different values of C

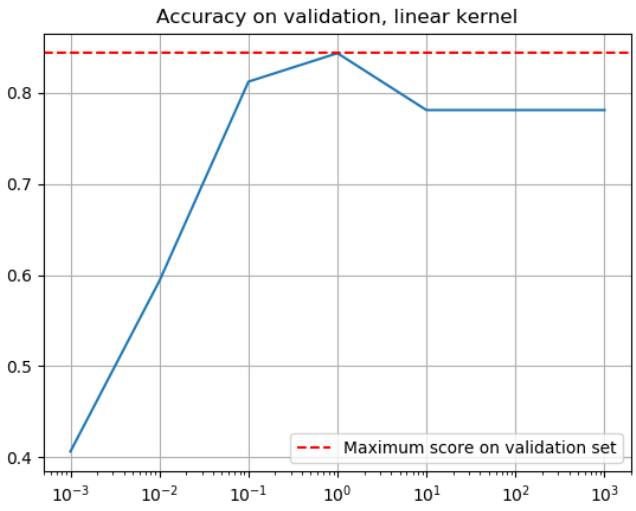

Figure 2: Accuracy on validation set as a function of C - Linear kernel

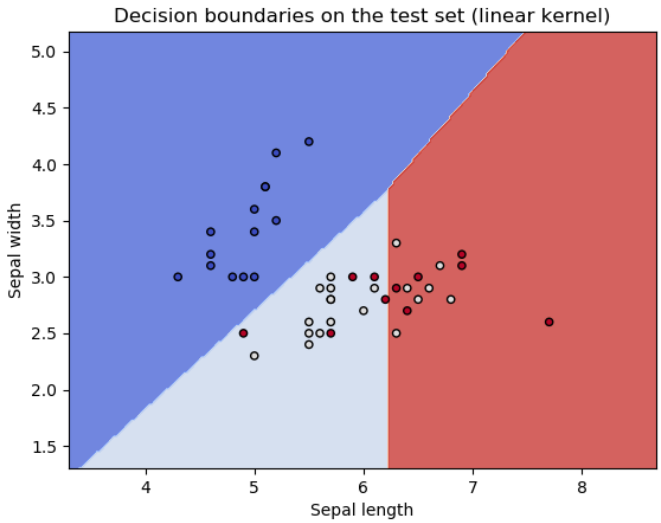

Figure 3: Decision boundaries on the test set - Linear Kernel

4 RBF Kernel

Given two data points x and x', the Radial Basis Function (RBF) kernel SVM k is defined as:

$$k(x, x') = \exp\left(-\gamma \|x - x'\|^2\right),$$

where γ must be greater than 0 and represents the inverse of the radius of the area of influence of samples selected by the model as support vectors, and $\|x - x'\|^2$ is the squared Euclidean distance between two data points **x** and **x'**.
C is the regularization parameter and it behaves as in the case of the linear kernel. The chosen range remains the same as before: $C \in [10^{-3}, 10^3]$.
Using a RBF kernel, the decision boundaries are computed on the traning set and Figure 4 shows the results obtained for different values of C. The value of the parameter γ is chosen automatically by the algorithm.

The main depictable difference between the boundaries shown in Figure 1 and Figure 4 is their shape: in the second case, the decision regions tend to cover the spread of the data, thanks to the use of a RBF kernel.
The best accuracy on the validation set is still 0.85 and it is reached for C=1 and C=10^3 (Figure 5). Moreover, for C=10^{-3} and C=10^{-2} the classifier underfits the dataset.

The accuracy on the test set improves compared to the one resulted using a linear kernel: its values is 0.76. In fact, in Figure 6 it is depictable that the number of misclassified test points is slightly lower than the ones showed in Figure 3 and the decision regions are more accurate.

4.1 Grid Search

In order to find the best parameters for an RBF kernel, a grid search is performed, analyzing different combinations of the values of C and γ.
The following ranges are chosen: $C \in [10^{-3}, 10^3]$, $\gamma \in [10^{-9}, 10]$.
The model is trained on the training set and scored on the validation set. The resulting best parameters are C=100 and γ=0.1 with a score of 0.78. It appears evident as the accuracy has improved compared to the one obtained using an automatic value of γ. Figure 7 shows how those parameters score on the validation set.

The behaviour of the model is really sensible to the value of γ. In fact, if γ is too large, the region of influence will only include the support vector itself and overfitting will be unavoidable (decision

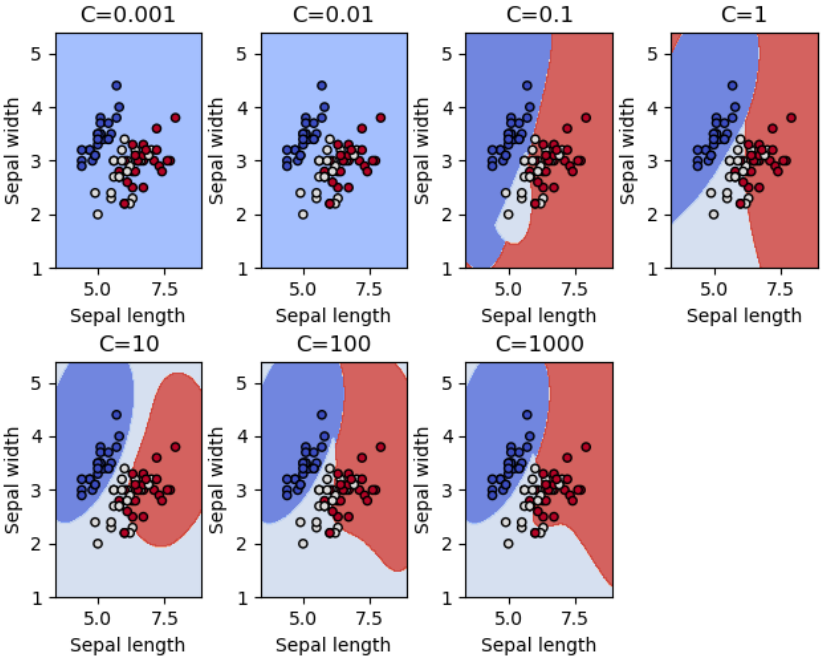

Figure 4: Decision boundaries of RBF kernel SVM for different values of C

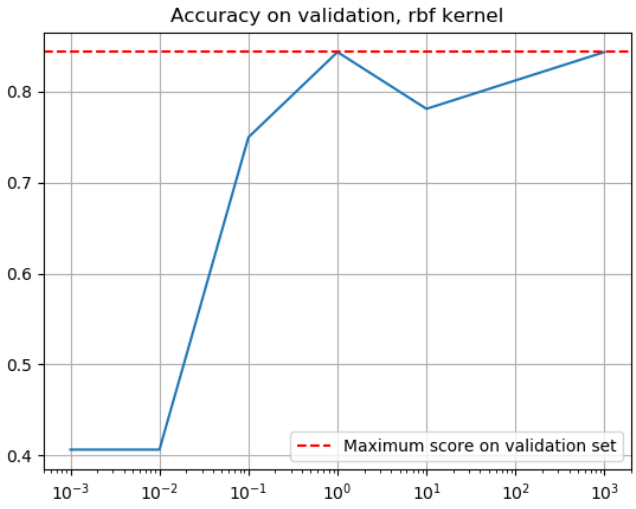

Figure 5: Accuracy on validation set as a function of C - RBF kernel

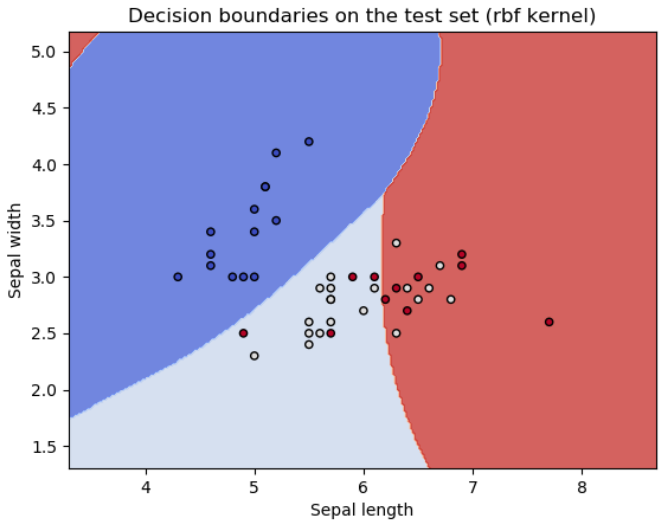

Figure 6: Decision boundaries on the test set - RBF Kernel

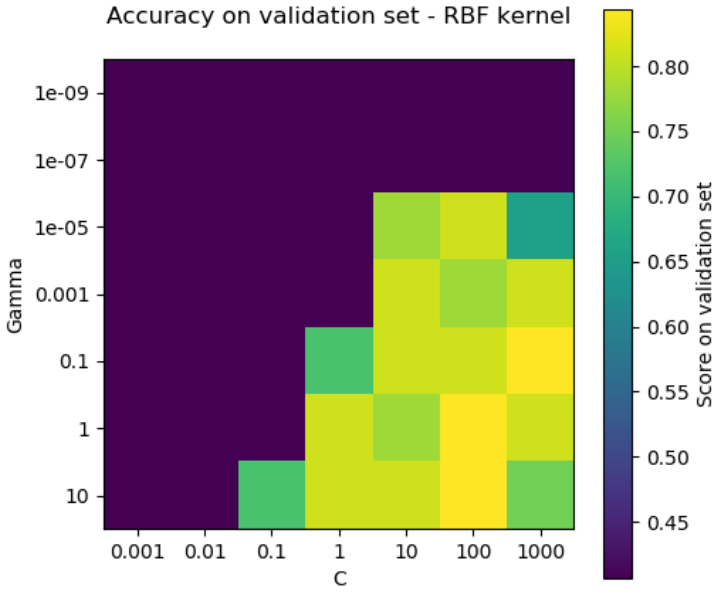

Figure 7: Accuracy on validation set for different C and γ

boundaries will be affected by individual data points). When γ is too small, the model will be too generalized because the area of influence of each support vector will include the whole training set.
The best found parameters are used to evaluate the model on the test set. The resulting accuracy is equal to 0.76. The decision boundaries of the classifier are shown in Figure 8: the decision regions create "islands" around data points because of the high value of γ.

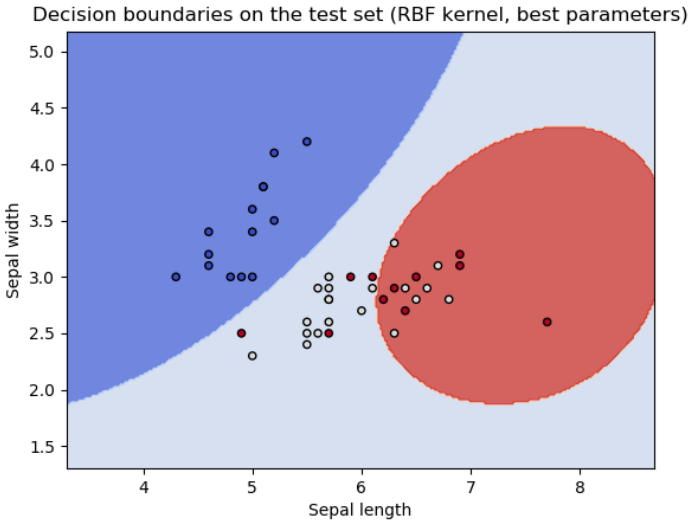

Figure 8: Decision boundaries on the test set with the best parameters

5 K-Fold Cross-Validation

K-Fold Cross-Validation (**CV**) is a procedure that tries to solve one the problems coming from the dataset split into training, validation and test sets. In fact, the use of the validation set basically improves the generalization performance of the model, but the results depend on the random chosen partition, especially if a small dataset like *Iris* is used. When this particular validation model is implemented, the train and validation sets are merged. The resulting training set is divided into k equal sized smaller subsets. Of the k subsamples, *k-1* are used as training set, while the remaining one is retained to validate the model. The partitioning is repeated k times and for each loop the accuracy is calculated. The k results are then averaged to produce a single estimation.
Here k is chosen equal to 5. The training and validation sets are merged so that they constitute the 70% of the data.
The grid search to find the best values of C and γ is repeated, performing a 5-Fold CV. For each possible combination of C and γ, the accuracy is calculated as the average of the scores obtained by the five iterations of the 5-fold CV (Figure 9). In this case, the best parameters result to be C=1000 and γ=0.001 with a final score of 0.84, the finest accuracy obtained so far.
The achievement of the best result happened thanks to the improvements introduced by the 5-Fold Cross-Validation procedure. In fact, the main advantage of CV is that all the partitions are used for both training and validation, and each subset is used as validation set exactly once: in this way, the dependency on the chosen pair of (training, validation) sets is avoided and generalization performance is enhanced.

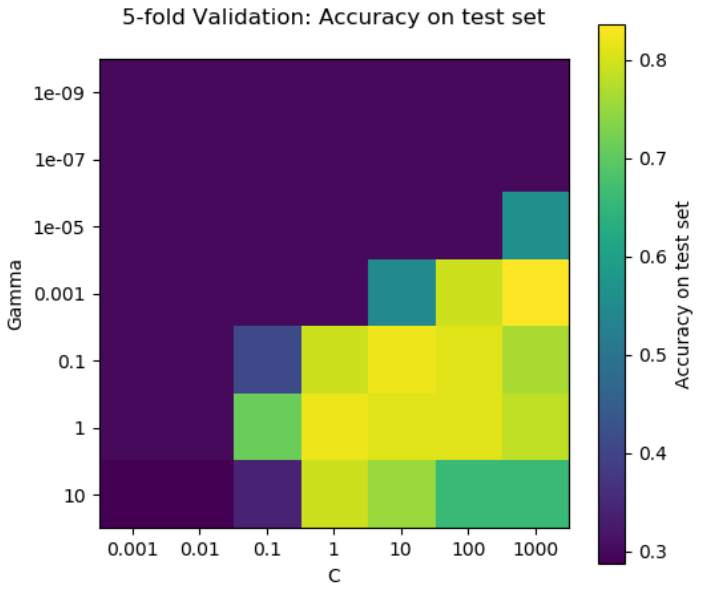

Figure 9: Accuracy on the test set using a 5-Fold CV

6 Conclusions

Support-Vector Machines are supervised learning models that have been used to categorize data points belonging to the *Iris* dataset in the presented experience.
The Linear and Radial Basis Functions kernels were proposed as case study: more complex functions involve more parameters, but let the model adjust to the spread of the data.
Because of the non-linearity of the dataset, the introduction of the regularization parameter C was necessary to obtain a feasible solution. The decision boundaries of the classifier were analyzed to detect the changes depending on the different values of C: large values of C imply a smaller-margin hyperplane, with the risk of overfitting; small values of C bring the optimizer to choose a larger-margin separating hyperplane, exposing the results to underfitting.
At last, the generalization performance of SVM was improved introducing the 5-Fold Cross-Validation model.