Antonio Norelli

Machine learning *a.a 2016/2017* *prof. Barbara Caputo*

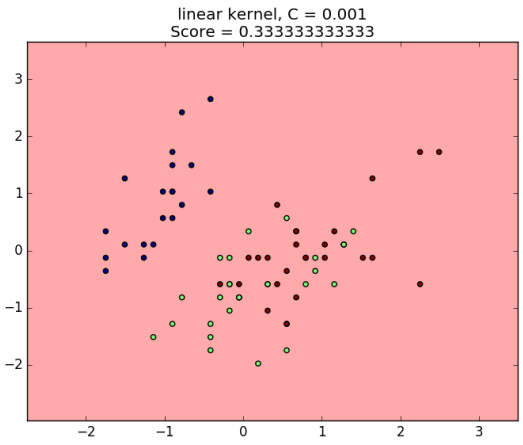
Homework 4: SVM

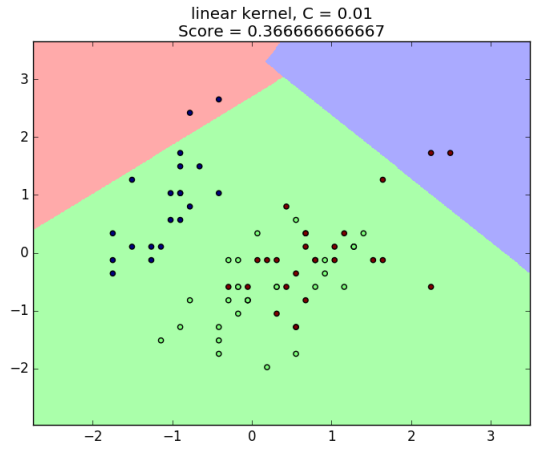
Support vector machine (SVM) with a linear kernel

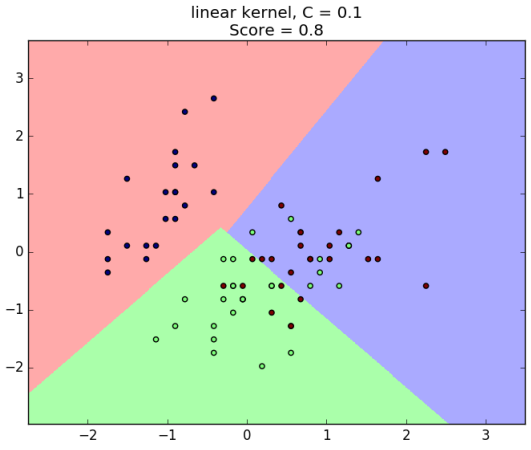
The dataset are the first two features of the Iris dataset available in the sklearn's standard dataset library.

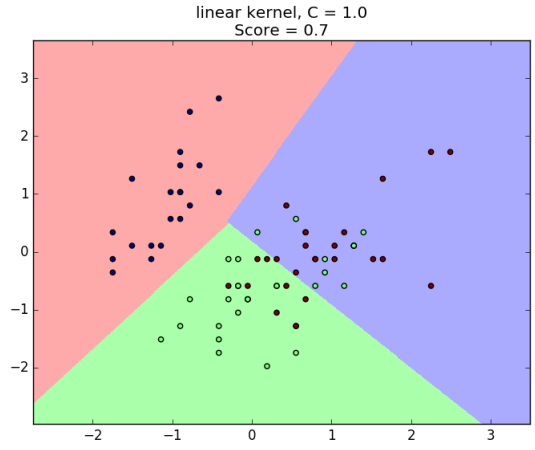
We have three classes, the data is split into train, validation and test set in proportion 5:2:3.

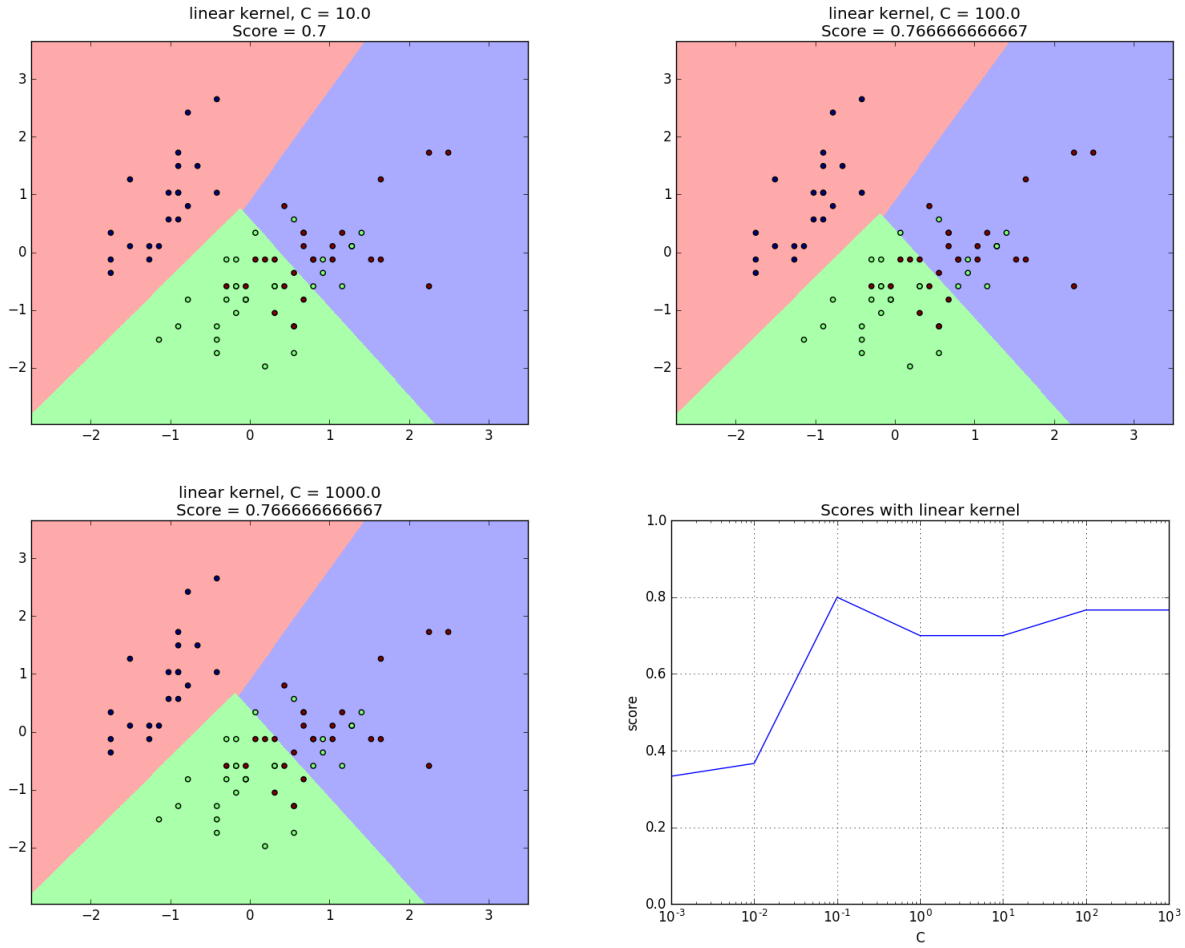
In order to find the best value of the C parameter for our linear SVM, we test it on the validation set for different values of C. The plots below show the training data and the boundaries of a SVM classifier with a linear kernel for different values of C. In the title of each graph is reported the score of the classifier with the validation set. In the final graph is showed the trend of the score.











How do the boundaries change? Why?

SVMs search for the separation with largest margins, this separation can be soft or hard, depending on how much we care about data inside the margins. The C parameter defines the weight of how much the samples inside the margins contribute to overall error, so looking at the extremes:

- a C=0 corresponds to a totally soft margin, with no penalization for samples inside the margins
- a C=∞ corresponds to a totally hard margin, with an infinite penalization for samples inside the margins

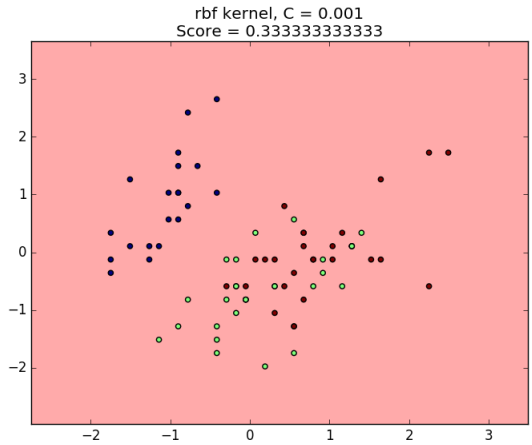
Therefore, the changing of the boundaries is caused by the value of the penalization of samples inside the margins given by C.

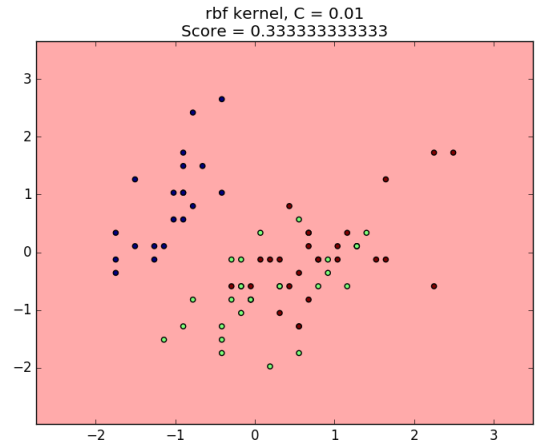
Evaluation of the best model on the test set

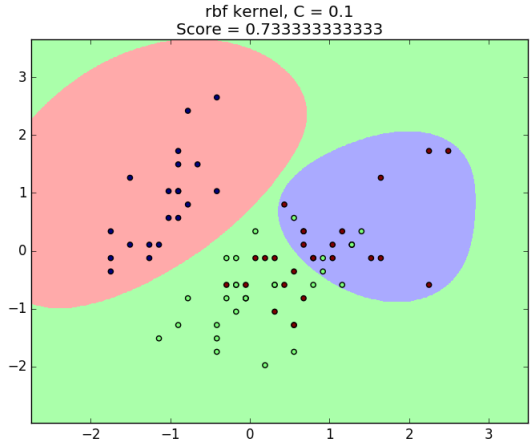
The best classification happens with C=0.1, the score with the validation set is: $score_v$=80%. So we test this classifier (C=0.1) on the test set, the result obtained is very good: $score_t$=82.2%. No overfitting symptoms.

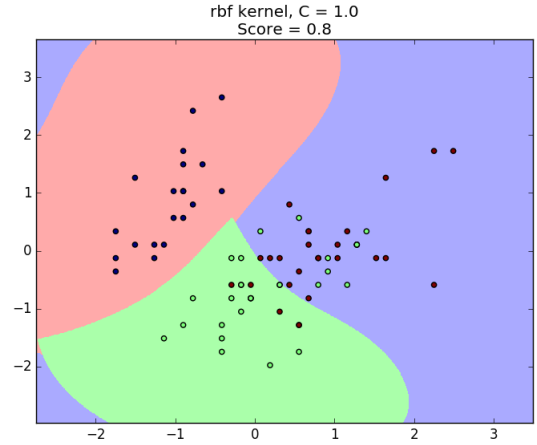
Support vector machine (SVM) with a Radial basis function (RBF) kernel

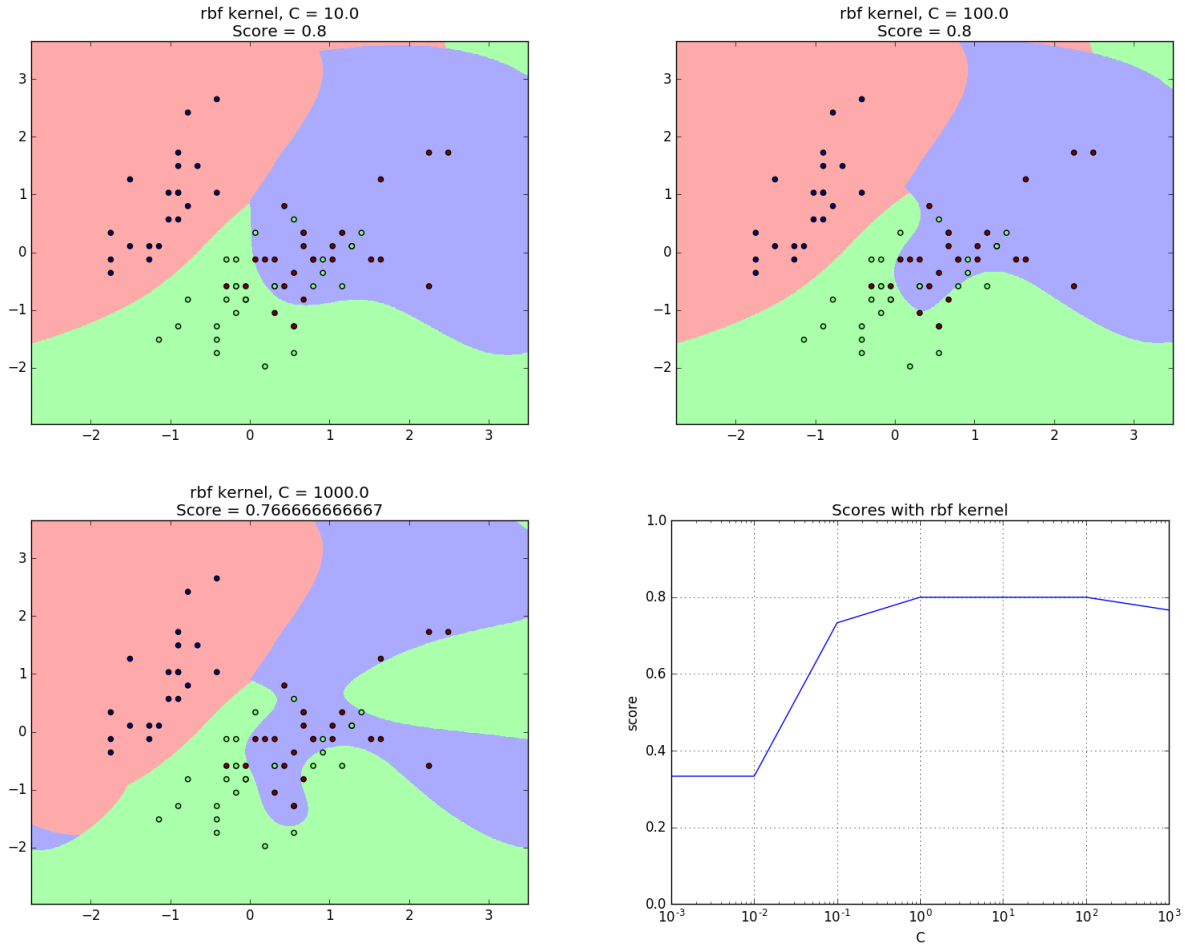
In order to find the best value of the C parameter for our SVM with a RBF kernel, we test it on the validation set for different values of C. The plots below show the training data and the boundaries of a SVM classifier with a RBF kernel for different values of C. In the title of each graph is reported the score of the classifier with the validation set. In the final graph is showed the trend of the score.


rbf kernel, C = 0.001
Score = 0.333333333333


rbf kernel, C = 0.01
Score = 0.333333333333


rbf kernel, C = 0.1
Score = 0.733333333333


rbf kernel, C = 1.0
Score = 0.8



Are there any differences compared to the linear kernel? How the boundaries are different?
The main difference is that the classifier is more complex and can draw non-linear boundaries. This is possible because using a RBF kernel the SVM can work in a higher-dimensional space, the non-linear boundaries are projection on a 2d space of linear boundaries in an higher-dimensional space (in the case of the RBF an infinite-dimensional space).

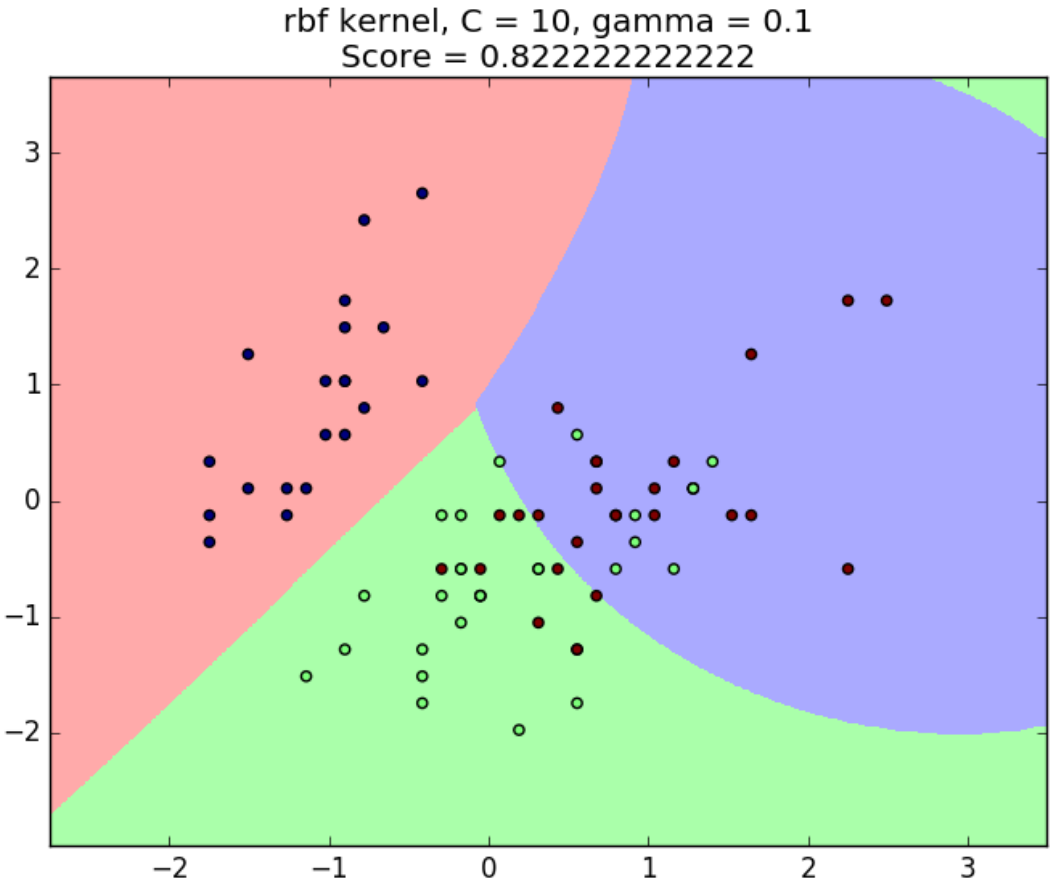
A more complex classifier involves in an increased vulnerability to overfitting, we can see this in the decrease of performance of the C=1000 SVM where, looking also at the decision boundaries, likely the model has learned some noise.

Evaluation of the best model on the test set
The best classification happens with C=1, 10, 100, the chosen value is C=10, the score with the validation set is: $score_v$=80%. So, we test this classifier (C=10) on the test set, the result obtained is good: $score_t$=77.8%. No big overfitting symptoms.

Grid search

In order to find the best value of the C and γ parameter for our SVM with a RBF kernel, we test it on the validation set for different values of C and γ, performing a grid search. The table below shows the score of the classifier with the validation set. In the final plot is showed the training data and the boundaries of the best configuration for the classifier (C=10, gamma=0.1). In the title is reported the score of the classifier with the test set. The result obtained, $score_t$=82.2%, is very good and as expected major than the previous one without the optimization of gamma.

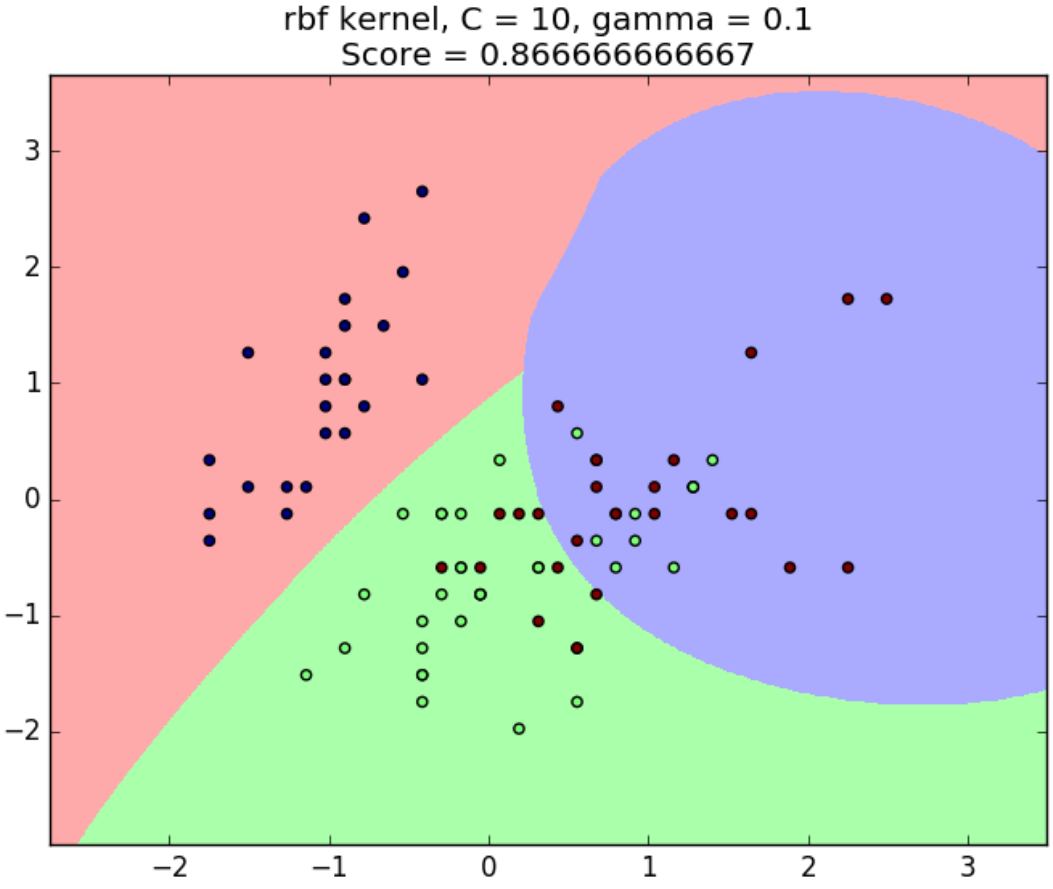
gamma\C	0.1	1.0	10.0	100.0	1000.0
1E-09	0.33	0.33	0.33	0.33	0.33
1E-07	0.33	0.33	0.33	0.33	0.33
1E-05	0.33	0.33	0.33	0.33	0.77
0.001	0.33	0.33	0.77	0.77	0.7
0.1	0.57	0.8	0.8	0.8	0.77
10	0.33	0.6	0.67	0.57	0.57



Grid search with a 5-fold validation

The procedure is the same of the previous point except for the samples used, instead of dividing rigidly in the three sets, train, validation and test; we perform a 5-fold validation. This procedure looks promising because of the limited size of the dataset used. So, in the table is reported the average of the score of five different fits with slightly different training and validation sets for each couple of C and gamma while the plot is analogous to the previous one.

gamma\C	0.1	1.0	10.0	100.0	1000.0
1E-09	0.28	0.28	0.28	0.28	0.28
1E-07	0.28	0.28	0.28	0.28	0.28
1E-05	0.28	0.28	0.28	0.28	0.56
0.001	0.28	0.28	0.56	0.74	0.72
0.1	0.45	0.77	0.77	0.73	0.74
10	0.28	0.63	0.65	0.62	0.62



Is the final score different [with k-fold]? Why?

The final score is different, the score with the test set after the 5-fold validation is higher, $score_{t,kf}$=0.87>0.82. As mentioned before perform the 5-fold validation with this little dataset is a good choice, the computational time is higher, but acceptable because of the few samples, and the improvement is remarkable. The better results obtained with a cross validation are consequences of a better sampling, the classifier is less vulnerable to overfitting and to the loss of information due to too small sets.