University of Liège

Machine Learning

Classification algorithms

Master 1 in Data Science & Engineering

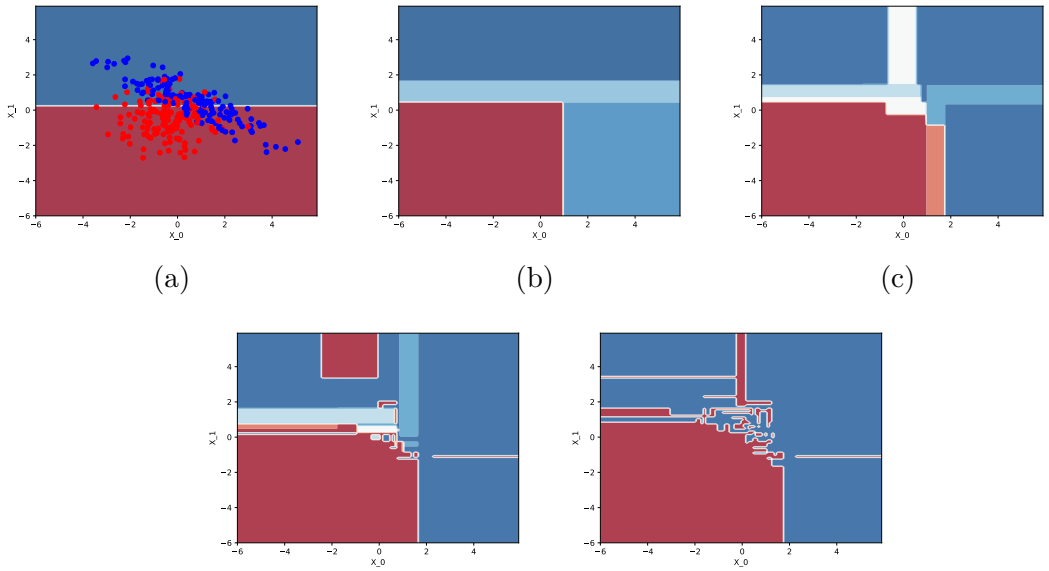
Authors :
Tom Crasset
Antoine Louis

Professors :
L. Wehenkel
P. Geurts

Academic year 2018-2019

1 Decision tree

1.1 Influence of the depth on the decision boundary



(a) (b) (c)

(d) (e)

FIGURE 1 – Decision boundary for a `max_depth` value of (a) 1, (b) 2, (c) 4, (d) 8 and (e) None. The points are intentionally left out of the first 4 figures to the decision boundary

As can be seen on Figure 1, a `max_depth` value of 1 splits the state space in two, placing the decision boundary horizontally in the middle of the set. The model doesn't properly fit the data at all and is clearly underfitting. The ellipsoid shape of the blue data points isn't detected at all, neither is the circular shape of the red ones.

Increasing the depth further to 2 introduces a linear separation, this time vertically, to select a majority of blue points.

As the depth even further increases, more and more linear separations will take place and thus will create rectangular surfaces all around the data points, sometimes trying to fit lonely data points inside another dense pocket of other coloured points, such as in Figure 1d or Figure 1e with a maxdepth of 8 or no constraint on the maximal depth. These clearly overfit the data, sometimes creating surfaces which don't even include a single data points they are trying to classify.

A maximal depth of 4 seems to be the ideal compromise for such a decision tree classifier as it captures the majority of the points without trying to contort itself around lonely data points and thereby skewing the accuracy.

The claim that the model seems to be more confident with an unconstrained maximal depth contradicts the results that came out of the testing. Indeed, the accuracy worsened as well as the variance on all the generations as can be seen on Table 1 in the following point.

1.2 Average test set accuracies

The Table 1 reports the average test set accuracies over five generations of the dataset 2 along with the standard deviation for each depth. It confirms what was previously said : with a maxdepth of 1, the model is clearly underfitting whereas it is clearly overfitting with no constraint on the depth. The maximal depth of 4 seems to be the ideal one with the best average accuracy and the lowest standard deviation over the five data generations.

	1	2	4	8	None
Average	0.667	0.819	0.833	0.832	0.791
Standard deviation	0.0491	0.0174	0.0098	0.0192	0.0304

TABLE 1 – Average test set accuracies and standard deviation for each depth

2 K-nearest neighbours

2.1 Influence of the n_neighbours on the decision boundary

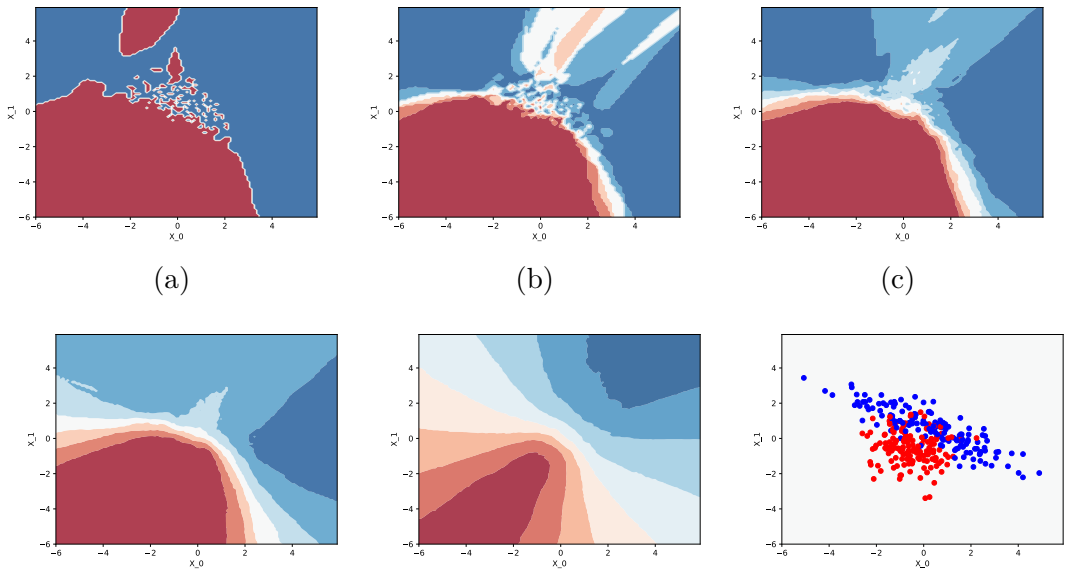


FIGURE 2 – Decision boundary for a `n_neighbours` value of (a) 1, (b) 5, (c) 25, (d) 125, (e) 625 and (f) 1200. The points are intentionally left out of the first 5 figures to show more detail of the decision boundary

The Figure 2 shows the different decision boundaries for varying values of `n_neighbours`. As one can see, with a value of 1 neighbour, the model clearly underfits the data, coloring regions on the upper portion of the graph in red, even though red points are nowhere near that region. As the number of neighbours increases, the model grows more confident and is able to better map the regions. Whereas there is still a faint red color on the upper portion of the graph for a value of 5, in the subsequent graphs for higher values, this region is blue again.

However, once a number of neighbours of 625 is used, the regions mapped start to become too broad and begin to fade into each other. This is because the total training set has only 1200 samples and with 625 being more than half of the points, the closest points encompass a large amount of the dataset. It is clearly overfitting.

As a number of 1200 is reached, the whole figure is colored a faint blue and that is understandable. In fact, it is to be expected because all the points are used to classify the points and it just so happens that the blue points won over the red ones, but it could have been reversed. The reason might be the disposition of the blue points, they are more spread apart than the red ones and thus cover a bigger region of the plane.

2.2 Optimisation with a ten-fold cross validation strategy

Our methodology for the 10-fold cross-validation testing is pretty straight forward. K-fold cross validation consists in separating the dataset into K sections, training on a subset of them and testing on the rest of the subset. To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds to give an estimate of the model's predictive performance.

In Python, a simple method for doing this is the `cross_val_score()` method. This is repeated for varying values of `n_neighbours` to find the optimal value, that is the value that has the lowest misclassification error.

As expected, the optimal number of neighbours is 125. This confirms the trend that is shown on Figure 2. Indeed, the number 125 is neither too high nor too low to accurately classify the points in the space. 125 is more or less 10% of the size of the training set used.

Additionally, the misclassification error has been plotted for the different values of `n_neighbours` on Figure 3 and it shows that 125 has the lowest error value and that for 1200 , the error is nearly 50%, which is equivalent to flipping a coin.

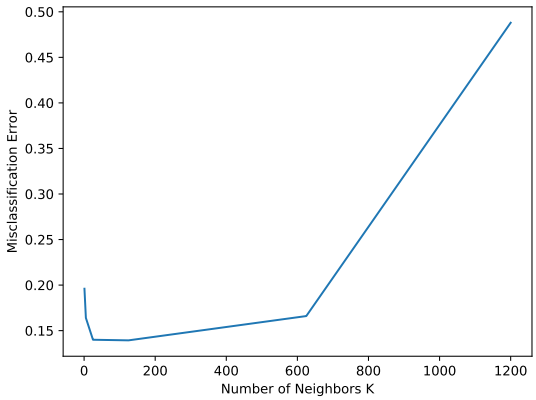


FIGURE 3 – Misclassification error for multiple values of `n_neighbours`

3 Linear discriminant analysis

3.1 Linearity of the decision boundary of LDA

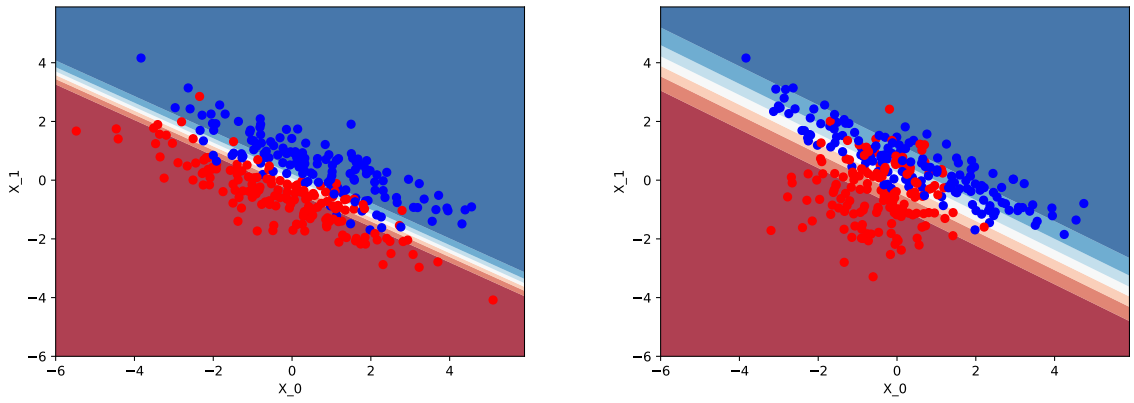
In the two classes case, it is quite easy to show that the decision boundary of LDA is linear. Each class density being modelled by multivariate Gaussian, LDA arises in the special case when we assume that the classes have a common covariance matrix $\sum_k = \sum \forall k$. In comparing two classes k and l, it is sufficient to look at the log-ratio. Then, we see that

$$\begin{aligned} log\frac{Pr(G=k|X=x)}{Pr(G=l|X=x)} &= log\frac{f_k(x)}{f_l(x)} + log\frac{\pi_k}{\pi_l} \\ &= log\frac{\pi_k}{\pi_l} - 0.5(\mu_k+\mu_l)^T \sum^{-1}(\mu_k-\mu_l) + x^T \sum^{-1}(\mu_k-\mu_l), \end{aligned} \tag{1}$$

is an equation linear in x. The equal covariance matrices cause the normalization factors to cancel, as well as the quadratic part in the exponents. This linear log-odds function implies that the decision boundary between classes k and l - the set where $Pr(G=k|X=x) = Pr(G=l|X=x)$ - is linear in x, in p dimensions a hyperplane. This is of course true for any pair of classes, so all the decision boundaries are linear.

If we divide R^p into regions that are classified as class 1, class 2, etc., these regions will be separated by hyperplanes.

3.2 Decision boundary



(a) (b)

FIGURE 4 – Decision boundary for the LinearDiscriminantAnalysis classifier on (a) dataset 1 and (b) dataset 2

The Figure 4 shows that the decision boundary separates the blunt of the two classes quite well, especially for the first data set.

3.3 Average test accuracies

	Dataset 1	Dataset 2
Average	0.903	0.839
Standard deviation	0.0092	0.0069

TABLE 2 – Average test set accuracies and standard deviation for each depth

The Table 2 reports the average test set accuracies over five generations of the dataset 1 and 2 along with the standard deviation. The average accuracy for dataset 1 is higher and the Figure 4 shows why : in the first dataset, the data clouds of the two classes have the same orientation and the same shape, whereas in the second one, the red class has a more circular shape. Another thing to note is that the standard deviation is very small for both datasets.

3.4 Similarities and divergences between datasets

As briefly commented in the previous point, the shape of the red points could differ from one dataset to the other. The fact that the linear discriminant analysis classifier gives better results for the first dataset is because the hypothesis of homoscedasticity holds true. Indeed, the covariance matrix describes the overall shape of the points in a given class and in the case of the dataset 1, the covariance matrices of both classes are indeed the same, so the hypothesis is satisfied. In dataset 2 however, as the Figure 4b shows, the red class has a more circular shape and thus the covariance matrix between the two classes is not the same, the hypothesis is not true and thus the accuracy of the prediction suffers.