INFO-8006: Introduction to machine learning

Project 1 - Classification algorithms

Pierre Hockers, Olivier Moitroux

Academic year 2018-2019

1 Decision tree

1.a Evolution of the decision boundaries across different depths

1.a.1 Illustrations

The following plots have been generated by using the function `plotDecBndaries()` in the file `dt.py`.

Depth 1: starting with the decision tree of depth 1, the dataset has been classified in two "confident" classes, along a straight boundary. This straight line split the dataset in two parts (two childrens), the red one containing the largest number of red dot and the reverse for the blue one. Let's note that a small light blue line can be seen around the boundary of the two main classes. However, a lot of points are not well classified. Even some points that are not in an overlapping area are not classified properly.

Depth 2: as the depth increases, the decision tree expands lighter areas that represent areas in which the points are less prone to be well classified, in other words, areas where there is a bit of overlap. We can see that the previous boundary is now larger (medium blue) and the red rectangle split in two, in order to let room for a light blue one. The latter one is a region where there is a bit of overlap between the two classes.

Depth 4: at depth 4, the decision tree is able to reduce the size of the different areas in such a way it can extends "confident" (darker) areas. More blue points are now incorporated in the dark blue area. The more interesting area is certainly the light blue one, which highlight a region of overlap that is now more precise than before. The medium blue boundary is reduced as well and the confidence of the model increases. An adjacent medium-red area also appears in the dark blue shape to fit the small number of points that are isolated. A rectangle with high uncertainty (few data there) is present around the xlabel 2.

Depth 8: The color of the different classes converge towards darker colors as the confidence of the model increases. Boundaries are also getting more complex as the tree tries to fit isolated points.

Depth unconstrained: lighter areas vanish as the decision tree is able to classify the learning dataset at best. Each leaf now is said to be pure.

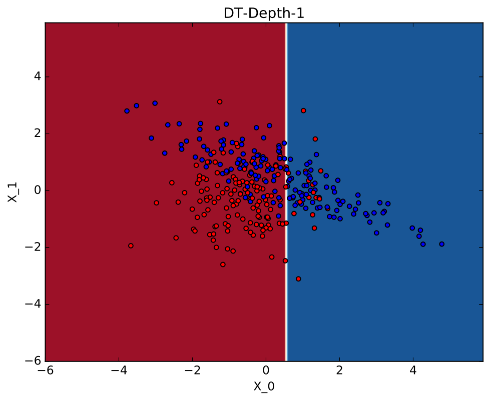
1.a.2 Underfitting and overfitting

A tree T overfits the learning sample iff T' such that:

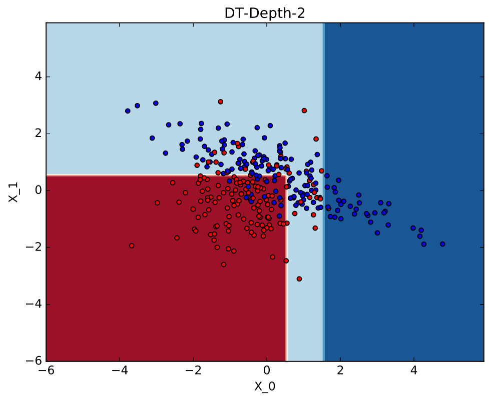
- $Error_{LS}(T) < Error_{LS}(T')$
- $Error_{unseen}(T) > Error_{unseen}(T')$

By increasing the depth, we can easily observe that the error on the learning sample decreases. Indeed, uncertain area (that are depicted by a lighter color), shrink to little shapes around the different boundaries. Most of the points get correctly classified, even the most isolated and sparse one. However, some of the classification done by the decision trees are not relevant. There is, indeed, few reasons to really fit the small red points that remains in the dark blue area as they could be assimilated, in real life, to noise. As can be point out in figure 3, the decision tree of depth 8 already overshoot a little bit as can be deduced from the non-adjacent vertical and horizontal red lines. The tree of depth unconstrained further overshoot by darkening and broadening these areas. When the depth is unconstrained, sparse island pop in the middle of the blue class, which certainly point out there is some overshoot going on. This phenomenon appears after a certain threshold where the accuracy score on the test set decreases while the accuracy score on the training set doesn't stop increasing. It is, of course, not desired.

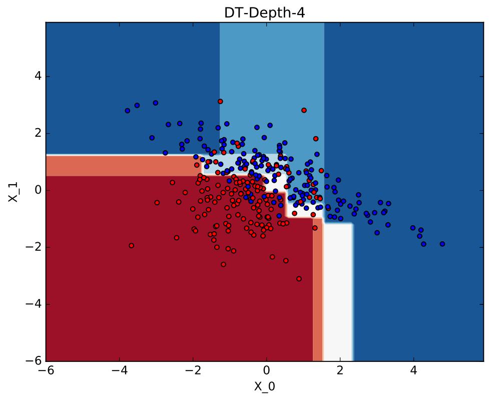
On the other side, underfitting also occurs in the early stage. The decision tree of depth 1 is too constrained in order to fit properly the data. A lot of dots are not correctly classified while there is plenty of room to expand areas that have few overlap. Some underfitting is also present in depth 2, because the light blue area could be further decrease to let room for a more confident boundary like the result produced by the depth 4 classifier.



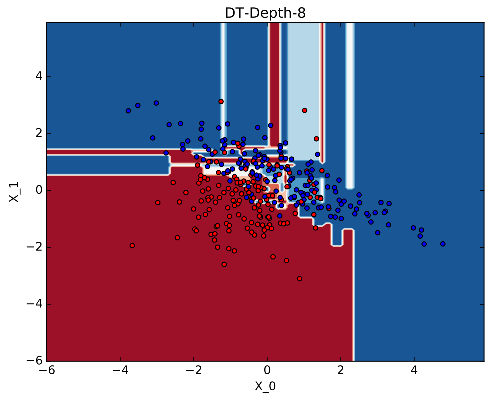
(a) Depth 1



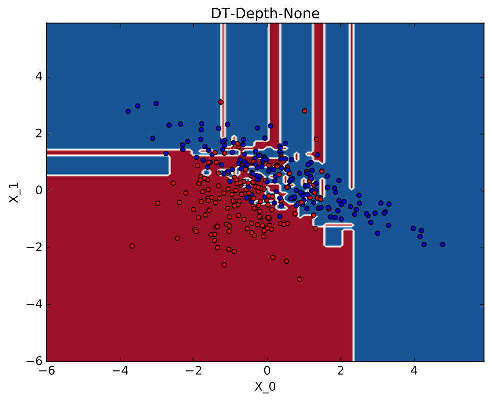
(b) Depth 2



(c) Depth 4



(d) Depth 8



(e) Depth unconstrained

Figure 1: Decision boundaries across different depths

1.a.3 Is the tree with a depth unconstrained better ?

When the depth is not constrained, the decision tree classifies the data such that each leaf is pure. This requires more time to compute for large dataset but it is not a problem at all here (decision trees are already fast anyway). However, classifying in such a way implies the model will overfit the data, especially in the region where there is an overlap between the two classes. At first glance, this method seems to be the best but one shouldn't forget that this classification mechanism is generally used for making predictions on unseen data. Testing this model against a new independent dataset (aka testing set) would show up that it is not optimal to go as far in the classification, as the score drops.

1.b Average test set accuracies and standard deviation

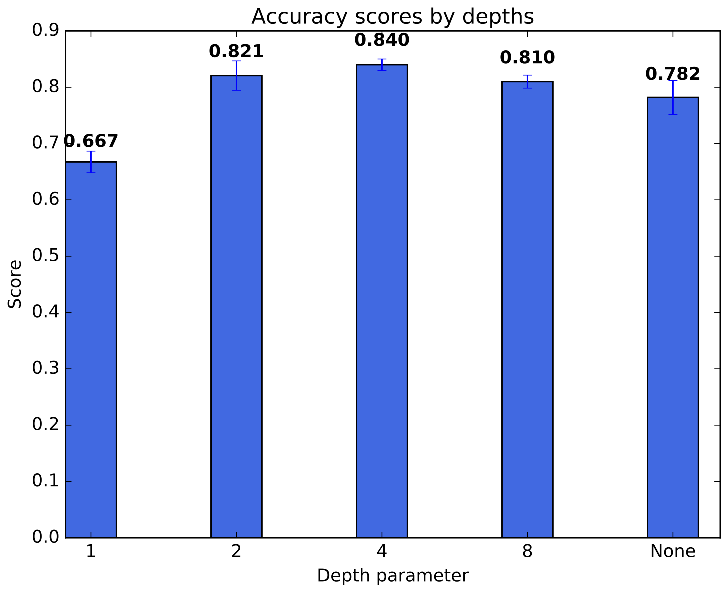


Figure 2: Accuracy score across several depth

Figure ?? displays the computed means and standard deviation of the accuracy score, averaged on five generations of dataset. The two statistics are also summarised in the table 1. These results tend to confirm what as been said before. At depth 1 and 2, the data is underfitted and reaches a maximum score when the depth reaches 4. Then, performance of our model drops due to overfitting. The decision tree of depth 4 performs well and produces, in addition, the lowest standard deviation, making it the best performer. As can be seen, the tree with unconstrained depth is not the best performer and has a quite high standard deviation.

	Depth 1	**Depth 2**	**Depth 4**	**Depth 8**	**Depth unconstrained**
Mean $[Score]$	0.66733333	0.82066667	0.8	0.81	0.782
Std $[Score^2]$	0.01925271	0.02602563	0.0101105	0.01154701	0.03015515

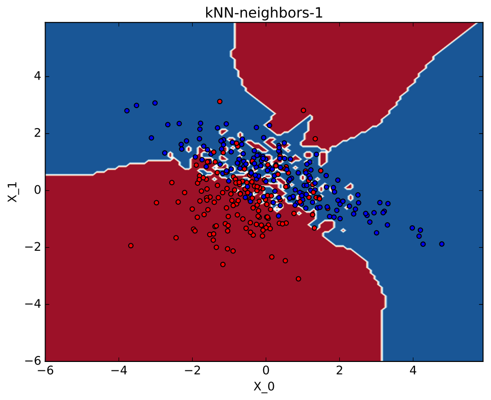
Table 1: Statistics of the score accuracy for different configurations

Overall, we can conclude that decision trees provide directly interpretable models but suffer from a high variance and are not that accurate. Indeed, just changing the seed can change the overall result ! Increasing the number of generation could increase the confidence in our statistics. Their main strength is, without a doubt, their scalability and fast computing time.

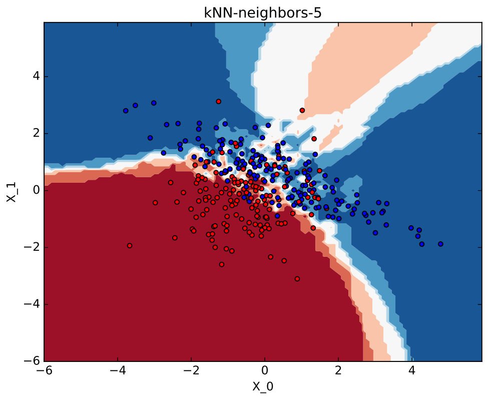
2 K-nearest neighbors

2.a Decision boundary analysis

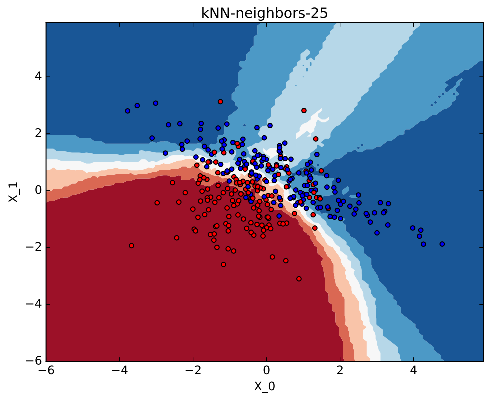
2.a.1 Decision boundary for each value of `n_neighbors`



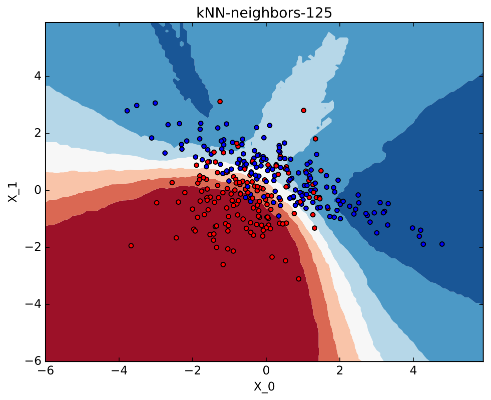
(a) 1 Neighbor



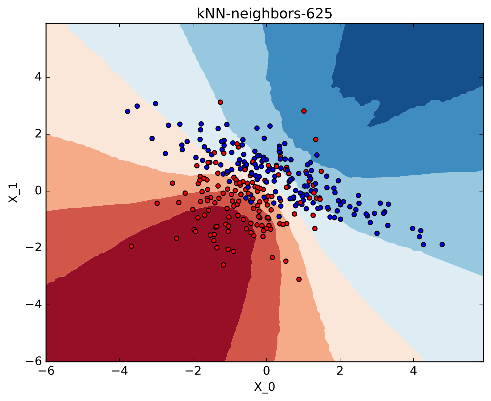
(b) 5 Neighbors



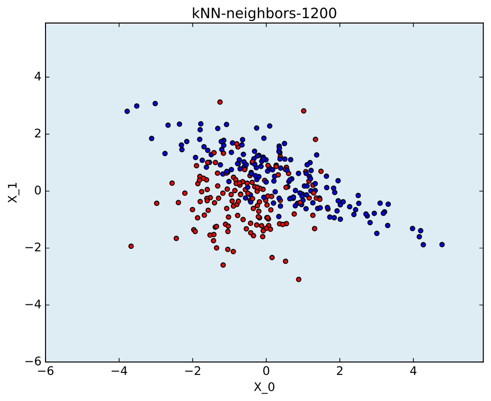
(c) 25 Neighbors



(d) 125 Neighbors



(e) 625 Neighbors



(f) 1200 Neighbors

Figure 3: Decision boundaries across different values of `n_neighbors`

2.a.2 Evolution of the decision boundary

K-NN finds the k nearest neighbors with respect to euclidian distance and so, outputs the most frequent class. As k increases, the error on the learning sample increases while the one on the test sample decreases until reaching an optimal of k. Then it goes up as well. We can observe a little anomaly in the case of $k = 1$, as a red region is present at the top of the graph. When considering 5 neighbors, the confidence in this classification decreases, until assimilating it to a blue region at 25 neighbors. It is quite logical to observe that the colours are getting lighter, as more points are used for the classification, and so potentially points that are red or blue in the overlapping areas. The decision boundary is not at all linear but as the number of neigbors considered approaches 625, it tends to follow a little bit the longest axis of the ellipsoid. When reaching 1200 neighbors, the model is unable to predict correctly, as too many points (of different colors) are considered at the same time for the classification. In this way, we can highlight that this method can be very sensitive to noisy variables.

When looking at the different graphs of figure 3, it seems that the best one is the one with 25 neighbors. Indeed, most of the test points are well classified and the less confident areas are due to overlaps or lack of data. (a) overfits the data but (b) lacks of precision at the top. (d) and (e) underfit the data while (f) is completely useless.

2.b Ten-fold cross validation

2.b.1 Methodology

In order to find the best value of `n_neighbors`, we split the dataset of 1500 samples in $k = 10$. We use one split as a test sample and the 9 remaining one as a train set. We then compute the boundary and store the accuracy score for each value of `n_neighbors`. Once it is done, we can do the same for the 9 remaining permutations of the test split. In this manner, we obtain a k×`n_neigbors` array from which we can determine the best value of `n_neighbors` by finding the indice corresponding to the column with the maximum average accuracy. This table is called `score` in our code and is printed in the console. Table **??** summarises the mean of the accuracy scores. As can be inferred from this table, the best value of `n_neighbors` is indeed 25 as was correctly guessed in the previous subsection. Because it maximises the average accuracy across the different and independent test set, it the best value for the parameter.

2.b.2 Score and optimal `n_neighbors`

n_neighbors	1	5	25	125	625 1200
Mean [Score]	0.77733333	0.82133333	0.84733333	0.83933333	0.82733333 0.55133333

Table 2: Caption

3 Linear Discriminant Analysis

3.a Linearity of the decision boundary

Thanks to the *Bayes theorem*, we can write:

$$P(y = k|x) = \frac{f_k(x)\pi_k}{\Sigma_{l=1}^{K} f_l(x)\pi_l} \tag{1}$$

Discriminant analysis assume that each class density function $f_k(x)(k = 1, \ldots, K)$ is a multivariate Gaussian:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)} \tag{2}$$

where $x \in \mathbb{R}^p$ is the feature vector and μ_k and Σ_k are respectively the mean and the covariance matrix corresponding to class k.

Linear discriminant analysis further assumes that the covariance matrices are equal: $\Sigma_k = \Sigma\ \forall k$. Let us assume we want to compare the two classes that are k and l. Taking the log-ratio of (1) and injecting (2) gives:

$$\begin{aligned} \log \frac{P(y=k|x}{P(y=l|x)} &= \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l} \\ &= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)^T \Sigma^{-1}(\mu_k - \mu_l) + x^T \Sigma^{-1}(\mu_k - \mu_l) \end{aligned} \tag{3}$$

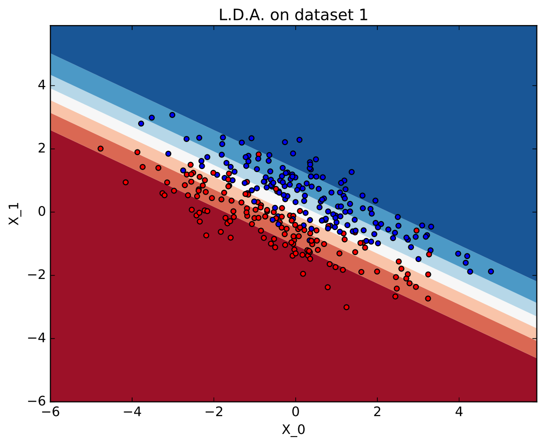
which is linear in x. Because the two covariance matrices are equal in this case, the normalization factors cancel as well as the quadratic component in the exponent. The set where $P(y = k|x) = P(y = l|x)$ is called the decision boundary between the classes k and l.

Equation (2) implies that this decision boundary is linear in x and an hyperplane if there are p dimensions. This is true for any pair of classes, so all the decision boundaries are linear. If we divide $\mathbb{R}^p$ into regions that are classified as class 1, class 2, etc., these regions will be separated by hyperplanes.

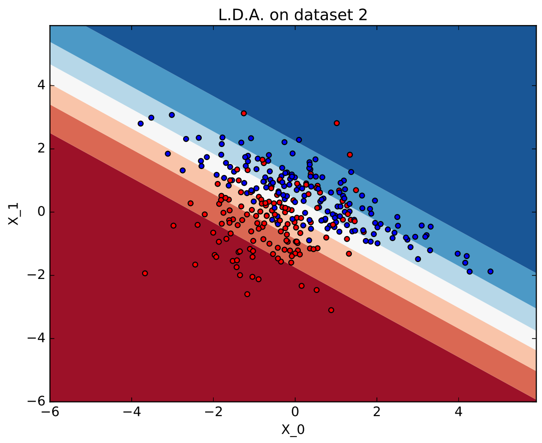
Reference: *The element of statistical learning*, Hastie et al.

3.b Decision boundaries of the two datasets

The decision boundaries are illustrated for both datasets in figure 4. As expected, the decision boundary is linear in both cases (cfr. above). We observe that the model is rather well suited, especially for the dataset 1 where the model is even more confident. This is unsurprising as the first one is generated in a way blue dots and red dots both come from an ellipsoid that has just been shifted. That is also why, the lighter areas are less present than in the previous models.



(a) Dataset 1



(b) Dataset 2

Figure 4: Decision boundaries computed with L.D.A. on two different datasets

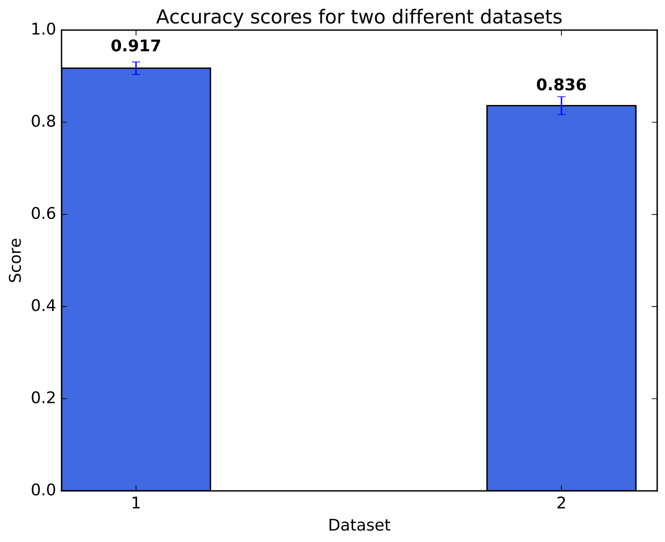


Figure 5: Means and standard deviations of the accuracy score on two different datasets using L.D.A.

3.c Average score accuracy and standard deviation

The average score accuracy and the standard deviation, averaged on 5 generations of datasets, are reported in table 3 and can be seen graphically in figure 5. The standard deviation is rather small, which is good (much better than decision trees f.e.). Unsurprisingly, the average score accuracy is fairly good for the dataset 1 and is still decent for the second dataset.

	Dataset 1	**Dataset 2**
Mean $[Score]$	0.91733333	0.836
Std $[Score^2]$	0.01372751	0.01936779

Table 3: Statistics of the score accuracy for different datasets

3.d Similarities and divergence between the two datasets.

As the first dataset is elliptic, the LDA performs better than second one. Indeed, elliptic shapes are better suited for a linear classification (along the longest axis) and that is why the mean of the score accuracy is higher. As it is made of two identical and shifted gaussians, the model is more confident for the decision boundary as the lighter areas are narrower. In the same manner, the resulting standard deviation is also smaller. The linear decision boundary of the second dataset has a steeper slope. However, both performs well and have success in differentiating the overlapping areas (with a little bit less confidence due to more overlap for the second dataset).