Introduction to Statistical Learning

Olivier Roustant
& Laurent Carraro for Part 2

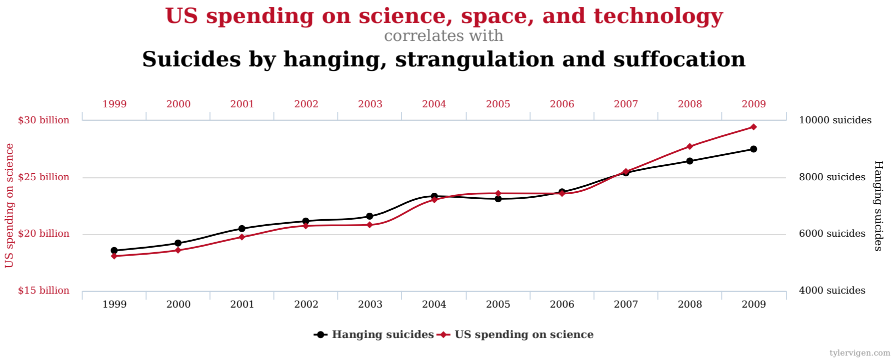
Mines Saint-Étienne

2016/09

Part 1 : Famous traps !

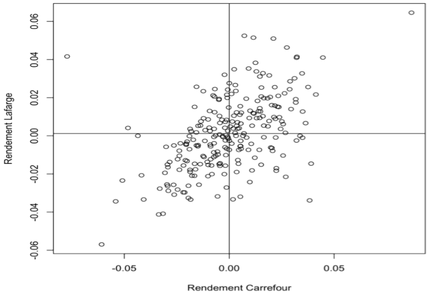
Trap #1- Spurious relationship, correlation $\neq$ causality

What do you think of the correlation of 0.99 between the two variables illustrated below ?



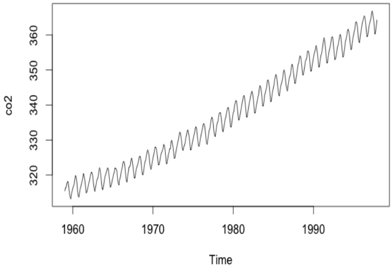
Trap #1- Spurious relationship, correlation $\neq$ causality

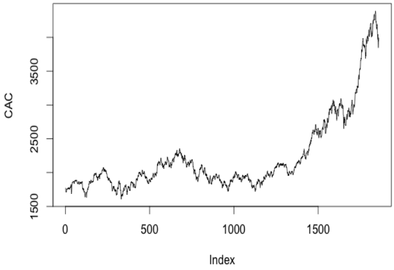
What do you think of the correlation of 0.52 between two daily returns of French stocks in 2 different sectors (food and construction) ?



Trap #1- Build your one spurious relationship !

Exercise 1 : Build a time series independently of the co2 curve, but with an estimated correlation > 0.95 with it !
Exercise 2 : Same question with CAC40 !





Trap #1- Spurious relationship !

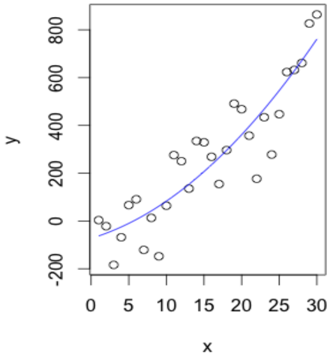
There are at least two problems :

- The ESTIMATOR of correlation is not consistent in presence of trend or seasonality !
- When it is (stationary time series for instance), then a THIRD variable can explain the observed correlations.

Never forget HUMAN THINKING !

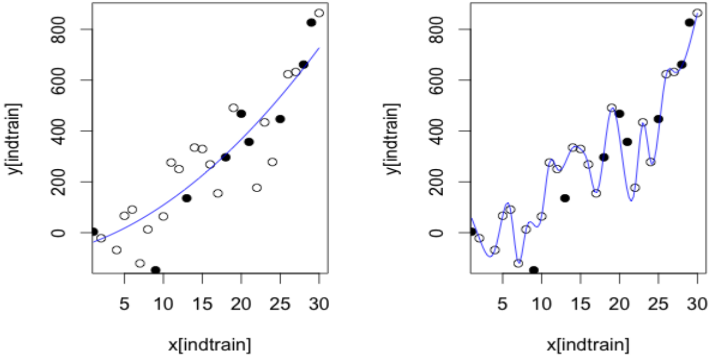
Trap #2- Overfitting

Here are some data from a physical phenomenon. What is your preferred model (2nd order polynomial or interpolation spline) ?



Trap #2- Overfitting

The same models, estimated on a training set of 20 data, chosen at random (empty points). Are the performances similar on the test set (filled points) ?



Trap #2- Overfitting

- Always look at the model performances on other data than the training set → external validation, cross-validation
- A good model should behave similarly on training & test sets

Part 2 : A guiding example

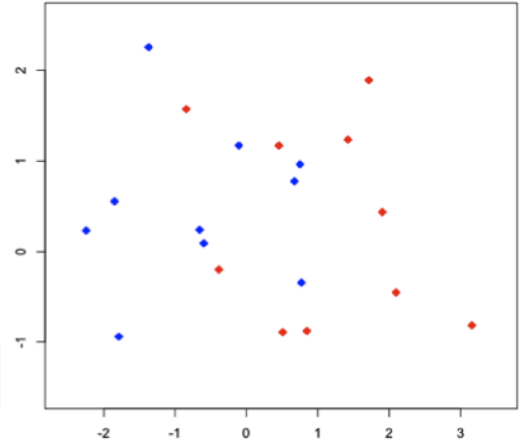
What follows is freely adapted from the book
The elements of Statistical learning, of T. Hastie, R. Tibshirani, J. Friedman (Springer, 2nd edition), available on internet.

We consider a simulated example for classification, where 2 populations "blue" and "red" are drawn from 2 mixtures of Gaussian distributions.

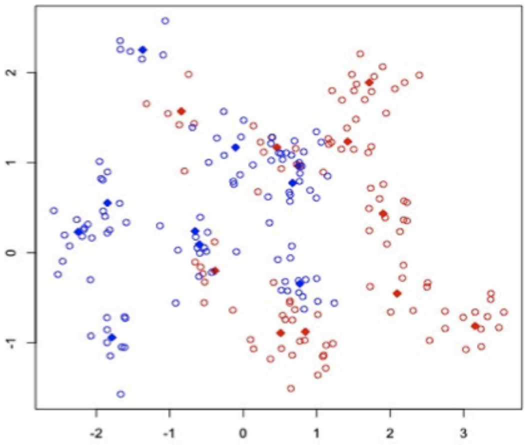
The aim is to find a rule to decide in which group a new individual should be classed.

Construction of the training sets

Step 1 : Simulate 10 points $M_1^1, \ldots, M_{10}^1$ for the "blue", drawn from $N(\mu_1, \Sigma)$, and 10 points $M_1^2, \ldots, M_{10}^2$ for the "red", from $N(\mu_2, \Sigma)$



Step 2 : Simulate a sample of size 100 as a mixture of $N(M_i^1, \Sigma')$ for the "blue", and $N(M_i^2, \Sigma')$ for the "red"



Bayes classifier

If we knew the simulation procedure, that is the distributions $f_{X|G=i}$, then we could use the Bayes classifier. Let x be a new point to classify.

- if $P(G = 1|X = x) > P(G = 2|X = x)$, then decide that x is "blue"
- if $P(G = 1|X = x) < P(G = 2|X = x)$, then decide that x is "red"
- if $P(G = 1|X = x) = P(G = 2|X = x)$, then ?

Here :

$$P(G = i|X = x) = \frac{0.5 f_{X|G=i}(x)}{0.5 f_{X|G=1}(x) + 0.5 f_{X|G=2}(x)}$$

Remark. Define $\hat{G}(x)$ as a decision rule at point x, and consider the 0-1 loss function :

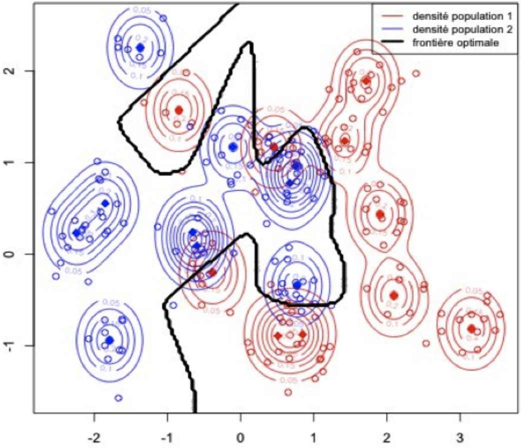
$$\begin{aligned} L(1,1) &= L(2,2) = 0 \\ L(1,2) &= L(2,1) = \alpha > 0 \end{aligned}$$

Then the Bayes classifier $\hat{G}$ minimizes the Expected Prediction Loss $E[L(G, \hat{G}(X))]$. It is enough to show that it is true knowing $X = x$:

$$\begin{aligned} EPL_x &= E[L(G, \hat{G}(X))|X = x] \\ &= L(1, \hat{G}(x))P(G = 1|X = x) + L(2, \hat{G}(x))P(G = 2|X = x) \end{aligned}$$

The Bayes classifier cancels $L(i, \hat{G}(x))$ where $P(G = i|X = x)$ is the highest.

The (optimal) frontier, obtained with Bayes classifier.



Classifiers from samples based on linear regression

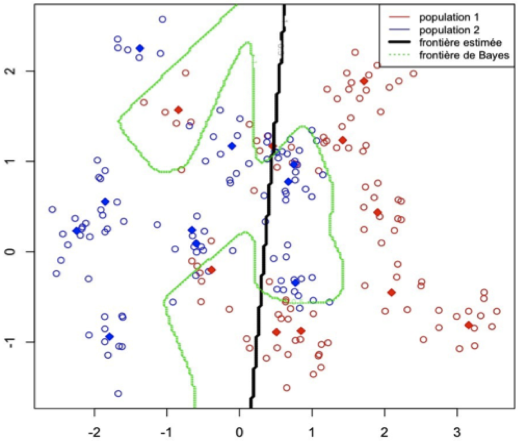
For each sample point define a value Y which is equal to 1 if "blue" and 0 otherwise, and let $\hat{Y}(x)$ be the prediction at a new point x :

$$\hat{Y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

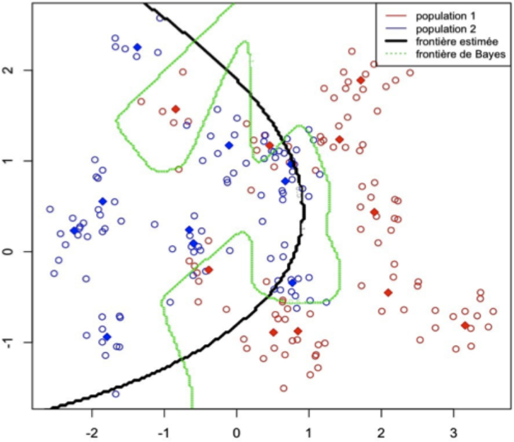
A classifier is :

- if $\hat{Y}(x) > 0.5$, then decide that x is "blue"
- if $\hat{Y}(x) < 0.5$, then decide that x is "red"
- if $\hat{Y}(x) = 0.5$, then ?

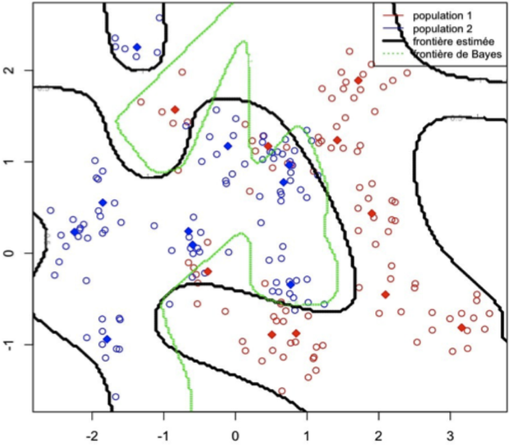
Linear frontier : classification rate 73.5 %



Quadratic frontier : classification rate 79.5 %



5th order polynomial frontier : classification rate 88 %



Nearest Neighbors Classifiers

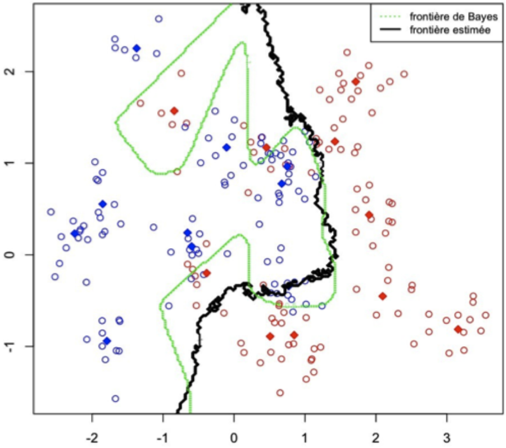
Let $N_k(x)$ the number of k-nearest neighbors of x, and $\hat{Y}(x)$ the proportion of these neighbors that belong to the "blue" :

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} Y_i$$

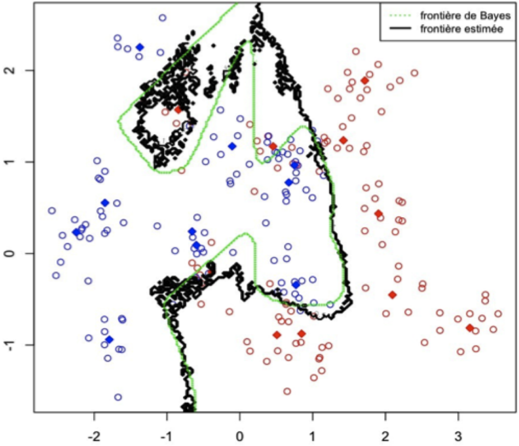
We can define a classifier by :

- if $\hat{Y}(x) > 0.5$, then decide that x is "blue"
- if $\hat{Y}(x) < 0.5$, then decide that x is "red"
- if $\hat{Y}(x) = 0.5$, then ?

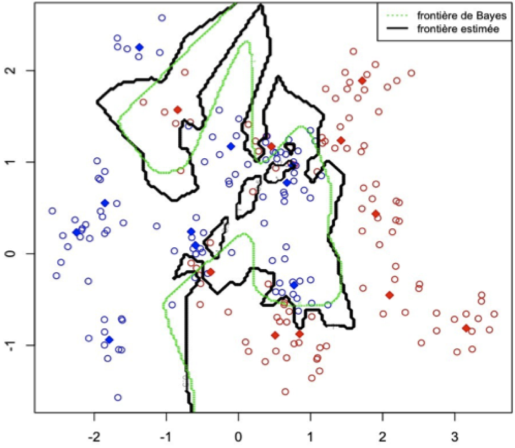
kNN with $k = 30$: classification rate 84 %



kNN with $k = 10$: classification rate 88 %



kNN with $k = 1$: classification rate 100 %



Temporary conclusions

- kNN is closer to the optimal method
- Parameters to estimate : k and d (polynomial degree)
- A classification rate of 100% is NOT the aim (see trap #2 'overfitting'...)

Error decomposition & bias-variance tradeoff

Assume that $Y(x)$ is deterministic, and let x be a new point. Denote $\mu(x) = E[\hat{Y}(x)]$. The quadratic error (risk) is decomposed as :

$$\begin{aligned} QE(x) &= E\left[\left(\hat{Y}(x) - Y(x)\right)^2\right] \\ &= (Y(x) - \mu(x))^2 + \mathrm{var}\left[\hat{Y}(x)\right] = \mathrm{Bias}^2 + \mathrm{Variance} \end{aligned}$$

Remarks

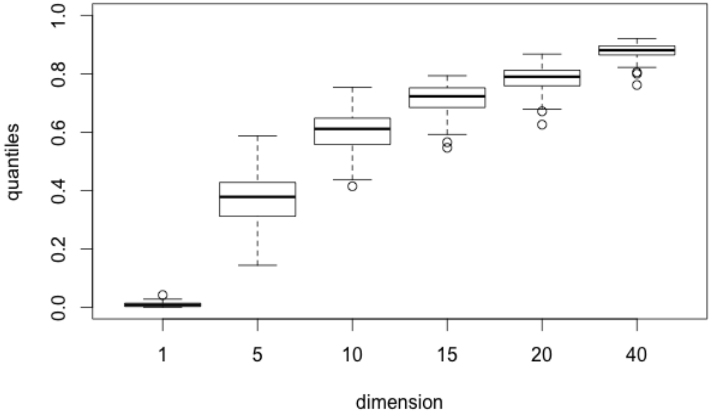
- for kNN, the bias is ≈ 0
- for the linear model, the bias is 0 if there is no model error (good basis functions).

The curse of dimensionality

Exercise : Let $X_1, \ldots, X_n$ i.i.d. uniforms on $[-1, 1]^d$, and consider the norm $\|h\|_\infty = max_{1 \leq j \leq d} |h_j|$.

- What is the distribution of $R = min_{1 \leq i \leq n} \|X_i\|_\infty$, the distance of the closest point to 0 ?
- What's happening when $d \to \infty$?

Boxplots for the distribution of the closest point to 0.



- In high dimensions, the sample points are close to the boundaries
- In 15D, the distance to the closest point is around 0.6

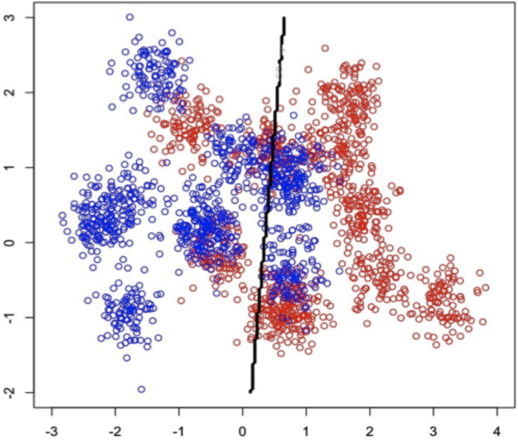
There are no neighbors in high dimensions $\rightarrow$ kNN cannot be used.
More generally any local method cannot be used.

Validation

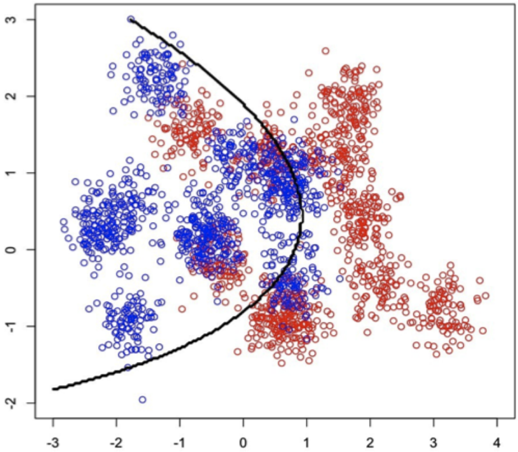
- Internal validation (on the training set only)
- External validation : Validate on a separate "test" set
- Cross validation : Choose the training set and test set inside the data (see later).

Validation results on the example

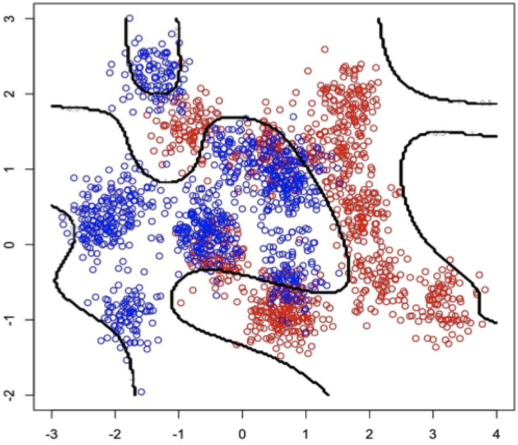
Linear frontier : classification rate 72.8 % (learning : 73.5 %)



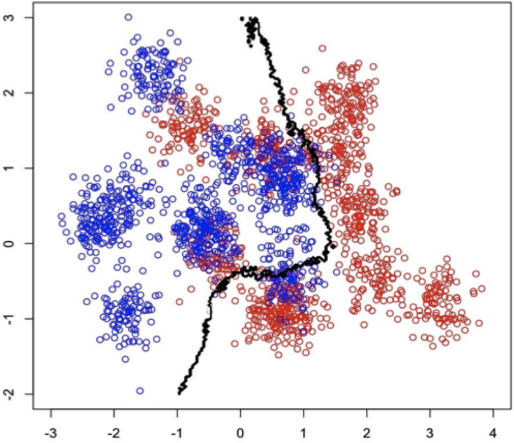
Quadratic frontier : classification rate 77.5 % (learning : 79.5 %)



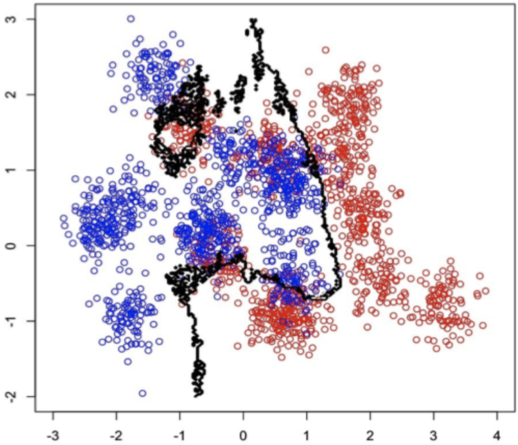
5th order poly. frontier : classification rate 84.5 % (learning : 88 %)



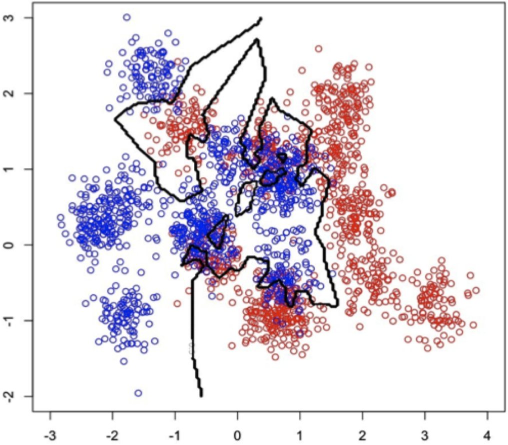
kNN with $k = 30$: classification rate 80.2 % (learning : 84 %)



kNN with $k = 10$: classification rate 84.9 % (learning : 88 %)



kNN with $k = 1$: classification rate 82 % (learning : 100 %)

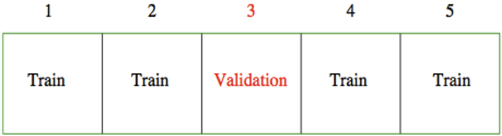


Conclusions

- The performance difference between training and test set is increasing with model complexity
- The performance on test sets does not always increase with model complexity
- Complex models sometimes take crazy decisions :
  - 5th order polynomial : boundaries of the x-axis
  - kNN for $k = 1$: islands in the middle

Cross validation

k-fold cross validation (CV) consists in choosing training & test sets among the data, and rotating them.
CV errors are computed by averaging.



(source : *The elements of Statistical learning*, T. Hastie, R. Tibshirani, J. Friedman)

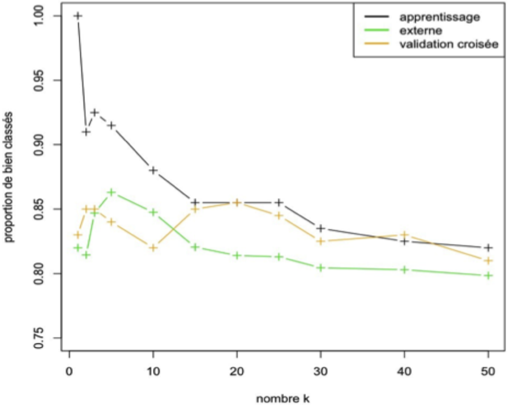
Define K 'folds' $F_1, \ldots, F_K$ in your data. For $k = 1, \ldots, K$, do :

- Estimate the model without F_k and predict on F_k
- Compute an error criterion (e.g. MSE) L_{-k} on the predicted values

Compute the CV error by averaging : $\frac{1}{k} \sum_{k=1}^{K} L_{-k}$

Cross-validation results on the example

Parameter k of kNN can be chosen by cross-validation



k	1	2	3	5	10	15	20	25	30	40	50
apprentissage	1,00	0,91	0,93	0,92	0,88	0,86	0,86	0,86	0,85	0,84	0,82
validation externe	0,82	0,82	0,85	0,86	0,85	0,82	0,81	0,81	0,81	0,80	0,80
validation croisée	0,83	0,85	0,85	0,84	0,82	0,85	0,86	0,85	0,83	0,83	0,81