ML2 : The landscape of machine learning
2. Linear models for classification

Pierre CHAINAIS

Trajectory

1. Classification and decision theory [recall]
    - Definitions
    - Statistical decision theory
    - Inference and decision
    - A simple method : K nearest neighbours (K-NN)
    - Model selection
2. Linear models for supervised classification
    - Linear discriminant functions
        - Separating hyperplane between 2 classes
        - Separation between several classes
    - Least mean squares classification
    - Linear Discriminant Analysis
    - Quadratic Discriminant Analysis (QDA)
    - Bayesian naive approach
    - Logistic regression

Definitions

The training set

Set of pairs of observations $\mathcal{D} = \{(x_n, t_n),\ 1 \leq n \leq N\}$ considered as i.i.d. random variables :

$$\begin{cases} x_n \in \mathcal{X} \subset \mathbb{R}^D, \\ t_n \in \mathcal{T}, \quad \text{card}\mathcal{T} = K \quad \text{(finite set)} \end{cases}$$

Typically $t_n = 0$ if $x_n \in \mathcal{C}_1$, and $t_n = 1$ if $x_n \in \mathcal{C}_2$.
More generally if $K > 2$: $t_n = (0...1...0)$, the 1 is at position k.

A **classification rule** is a function $f : \mathcal{X} \longrightarrow \mathcal{T}$.
e.g. $y(x)$ $<$threshold $\Rightarrow x \in \mathcal{C}_1$, $y(x)$ $\geq$threshold $\Rightarrow x \in \mathcal{C}_2$.

Learning

To build a decision rule f from some training set $\mathcal{D}$.

Definitions

- **Supervised** classification : $\mathcal{D} = \{(x_n, t_n), \quad 1 \leq n \leq N\}$,
- **Unsupervised** classification : $\mathcal{D} = \{(x_n), \quad 1 \leq n \leq N\}$,
- **Semi-supervised** classification : $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$

Prediction error, cost/loss function

0-1 Loss function : $L(a,b) = \begin{cases} 0 & \text{si} \quad a = b, \\ 1 & \text{si} \quad a \neq b \end{cases}$

The real error rate

$$E(f) = \mathbf{E}_{t,x}[L(t, f(x))]$$

which becomes for a 0-1 Loss function : $E(f) = P(f(x) \neq t)$

The empirical error rate (supervised classification)

$$E_N(f) = \frac{1}{N} \sum_{n=1}^{N} L(t_n, f(x_n))$$

which becomes for a 0-1 Loss function : $E_N(f) = \dfrac{\text{card}\{t_n \neq f(x_n)\}}{N}$

Remark : one also uses the term of "empirical *risk*".

Example : binary classification

$x \in \mathbb{R}$, $t \in \{0, 1\}$: 2 regions of decision,

$$\mathcal{R}_1 = \{x : f(x) = 0\}$$

$$\mathcal{R}_2 = \{x : f(x) = 1\}$$

Then :

$$E(f) = P(x \in \mathcal{R}_1, t = 1) + P(x \in \mathcal{R}_2, t = 0)$$

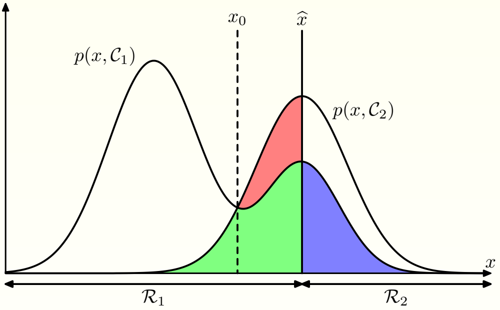
$$\boxed{E(f) = \int_{\mathcal{R}_1} p(x, t = 1)\ dx + \int_{\mathcal{R}_2} p(x, t = 0)\ dx}$$

that we want to minimize.

Remark : for K classes, it may be simpler to maximize

$$P(correct) = \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(x, \mathcal{C}_k) dx$$

Example : binary classification



- $\hat{x}$ = decision boundary
- error = blue + green + red
- optimum : if red region disappears $\Longleftrightarrow \hat{x} = x_0$
- possibility of using weights or a rejection region (no decision)

Statistical decision theory

Objective

Estimate f minimizing $E(f)$

$$\Longleftrightarrow$$

for all $(x_n, t_n) \in \mathcal{D}$, minimize $P(f(x) \neq t|x)$

Bayes' rule for classification

$$f^*(x) = \text{argmax}_k P(\mathcal{C}_k|x)$$

$=$ maximum a posteriori (MAP) estimate.

$E(f^*)$ is the **Bayesian error rate**.

Remark : $P(\mathcal{C}_k|x) = \dfrac{p(x|\mathcal{C}_k)P(\mathcal{C}_k)}{p(x)}$

cf. proba a posteriori $\propto$ likelihood x prior

Statistical decision theory

Theorem

Bayes' rule for classification is optimal.

Any other rule f is such that :

$$E(f^*) \leq E(f)$$

Exercise : binary classification

We assume that $t \in \{0,1\}$ and $x \in [0,5]$. Moreover :

$$P(t=0) = P(t=1) = \frac{1}{2},$$
$$p(x|t=0) = \mathcal{U}([0,2]),$$
$$p(x|t=1) = \mathcal{U}([1,5]).$$

1. Determine the Bayesian classifier and its error rate.

Inference and decision

- **inference :** determination of the $p(\mathcal{C}_k|x)$
- **decision :** use $p(\mathcal{C}_k|x)$ to affect classes

1. **generative model** : use $p(x|\mathcal{C}_k)$ and $p(\mathcal{C}_k)$ to deduce $p(\mathcal{C}_k|x)$.
   Rk : $p(x) = \sum_k p(x|\mathcal{C}_k)p(\mathcal{C}_k)$
   e.g. discriminant linear analysis...
   Easier interpretation, BUT the model can be badly adapted.
2. **discriminant models** : estimate $p(\mathcal{C}_k|x)$ directly
   e.g. logistic regression...
   Easier interpretation, BUT sometimes limited model.
3. **discriminant functions** : estimate $f(x)$ directly
   e.g. K nearest neighbours...
   Often loosely interpretable, BUT can be very efficient.

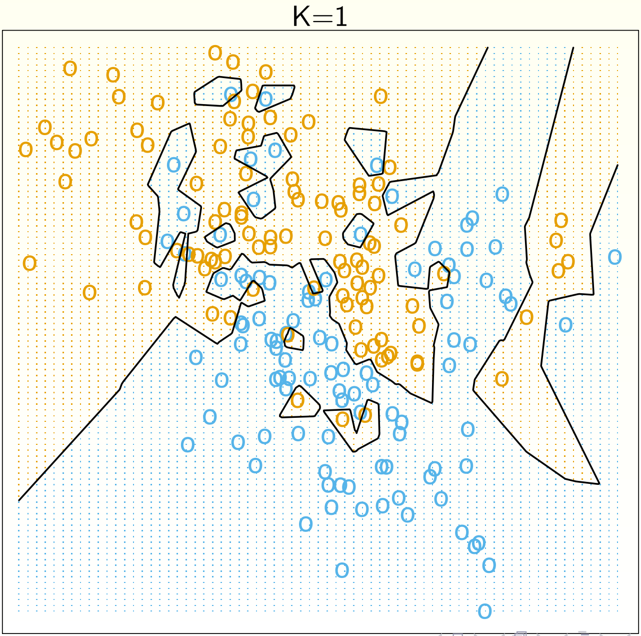
A simple method : K nearest neighbours (K-NN)

K-NN = K Nearest Neighbours

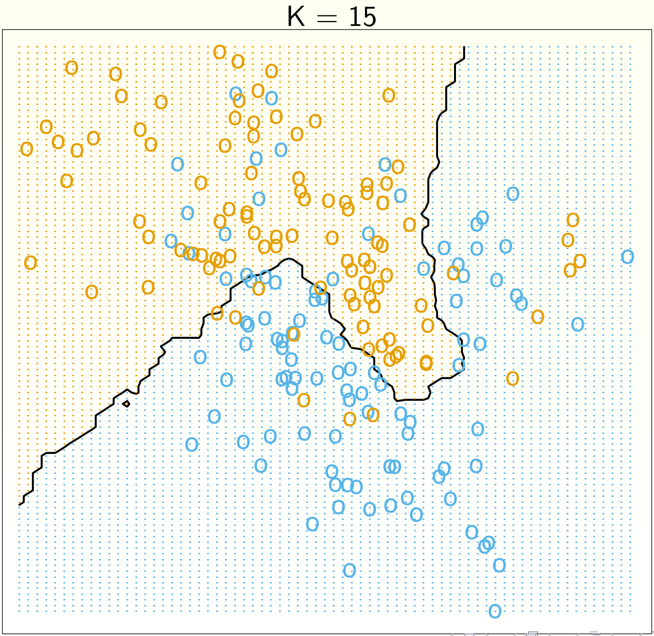
Idea : *Birds of a feather flock together ! (qui se ressemble s'assemble !)*

- **Data** $\mathcal{D}$ = N points avec N_k points $\in \mathcal{C}_k$, $\sum_{k=1}^{K} N_k = N$.
- **To classify** x **:** put x in the class that has majority among its K nearest neighbours.
- **Justification** :
  Let $\nu_k$the number of neighbours $\in \mathcal{C}_k$ (among K).
  One shows that $p(\mathcal{C}_k|x) \simeq \frac{\nu_k}{K}$
  so that one chooses $x \in \mathcal{C}_k$ such that ν_k is maximal.

A simple method : K nearest neighbours (K-NN)

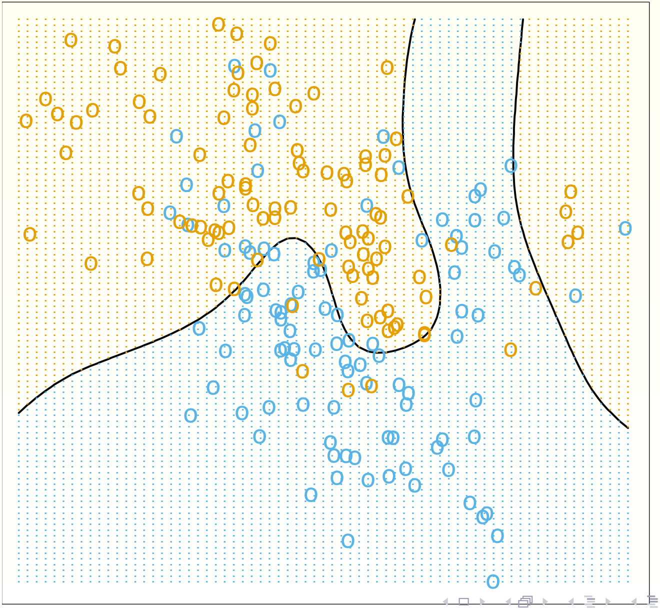


A simple method : K nearest neighbours (K-NN)



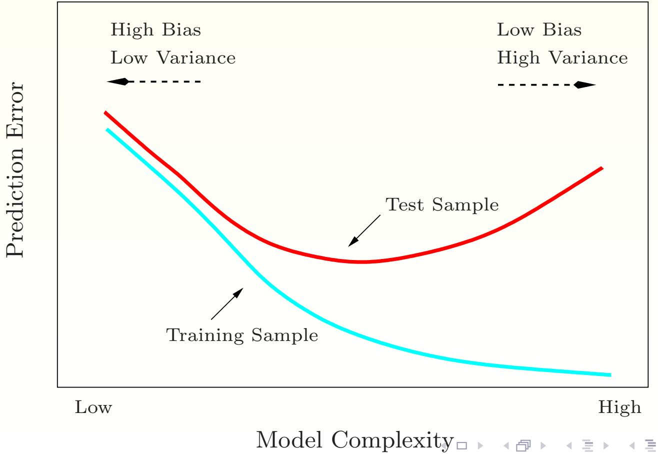
A simple method : K nearest neighbours (K-NN)

Bayes decision rule



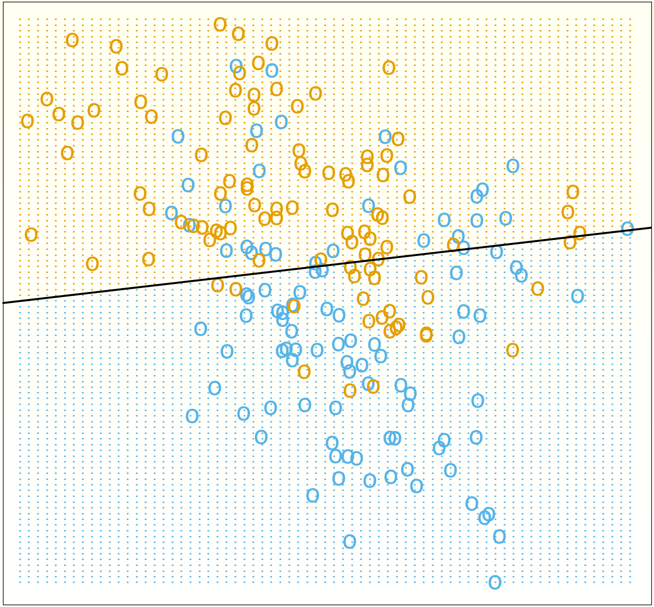
Model selection

- Too simple $\Rightarrow$ under-fitting (sous-apprentissage)
- Too complex (rich) $\Rightarrow$ over-fitting (sur-apprentissage)
- The learning error ($\neq$ test error) decreases with complexity ; it cannot be used on its own to choose the best model.

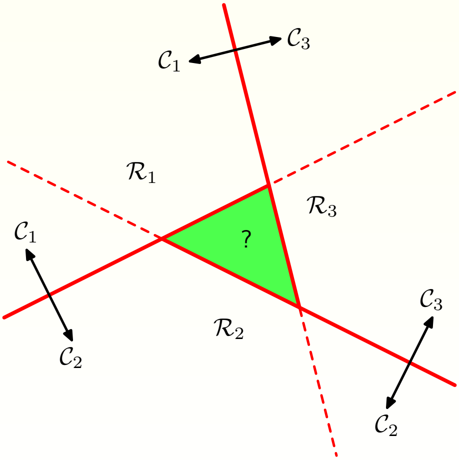


Trajectory

Separating hyperplane between 2 classes

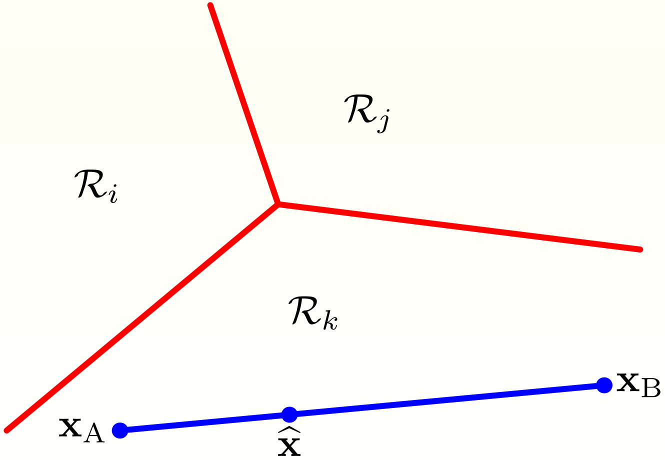


Combination of $\frac{K(K-1)}{2}$ classifiers 1 against 1

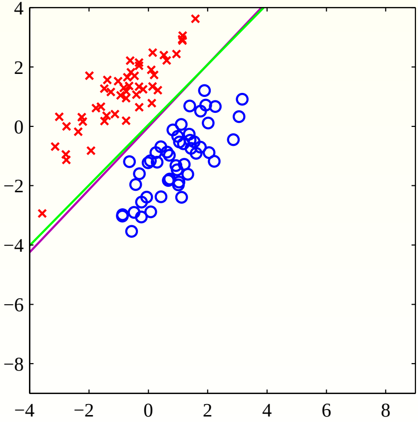


Separation between several classes

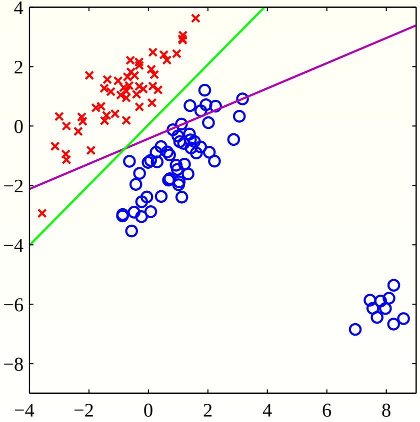
K linear classifiers and $y_k(\mathbf{x}) \geq y_j(\mathbf{x}), \forall j \neq k \Longrightarrow \mathbf{x} \in \mathcal{C}_k$



Sensitivity to outliers = extreme events

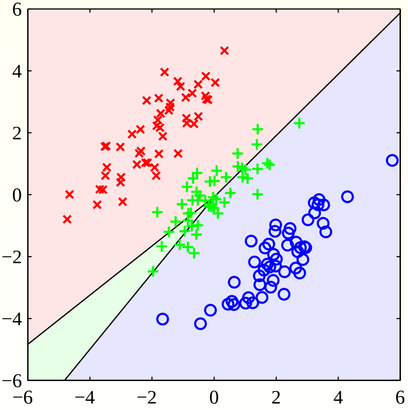


Sensitivity to outliers = extreme events

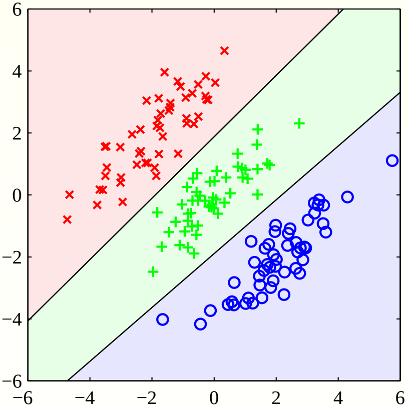


Least mean squares classification
Limitations

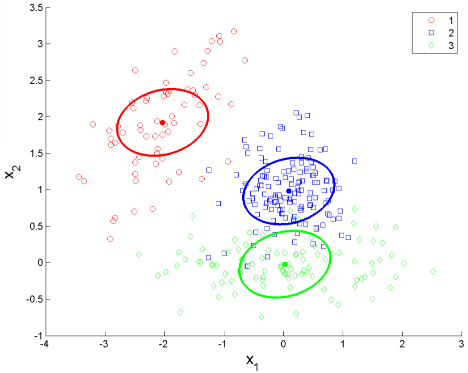
Unadaptated to certain situations



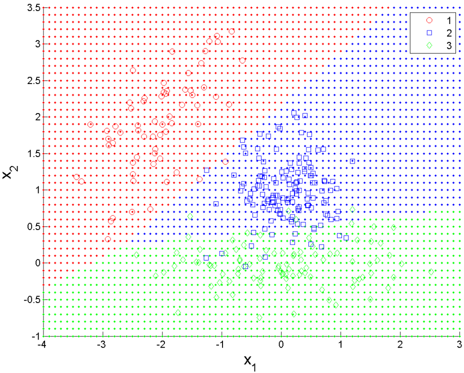
Unadaptated to certain situations



Linear Discriminant Analysis : a Gaussian model



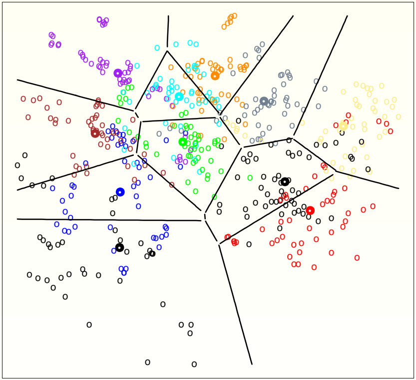
Linear Discriminant Analysis : a Gaussian model



Linear Discriminant Analysis (LDA)

Application to vowel recognition

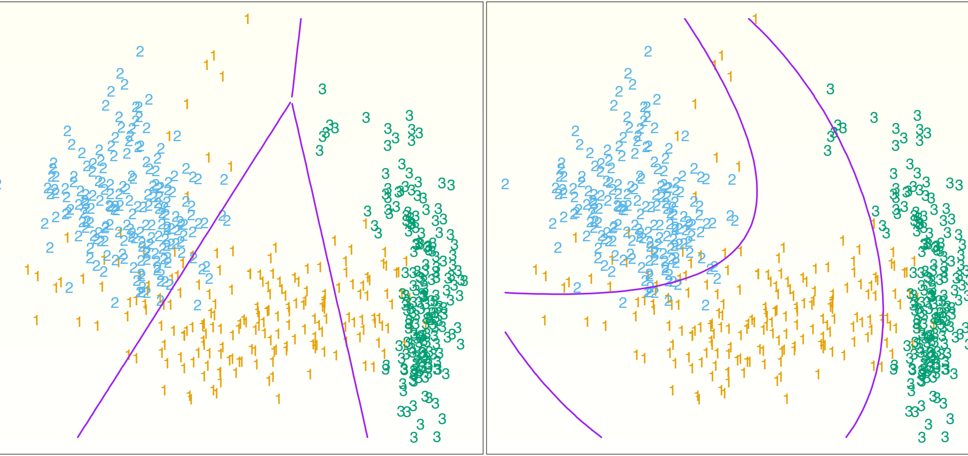
11 vowels $\Rightarrow K = 11$ classes described by $D = 10$ characteristics



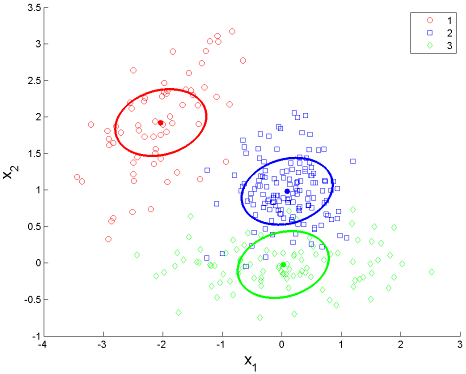
Classification based on sur 2 discriminant components (Fisher)
(see later on, Dimension Reduction)

Linear Discriminant Analysis (LDA)

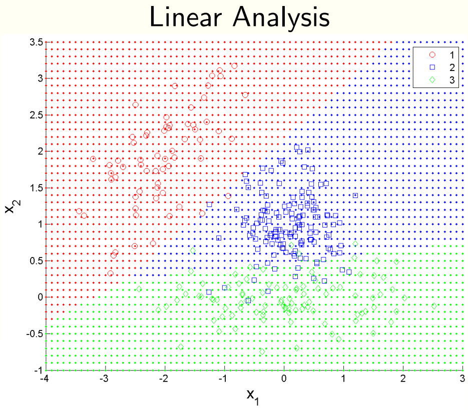
Application to $\phi_j(\mathbf{x})$: example, $x_1, x_2, x_1^2, x_1 x_2, x_2^2$



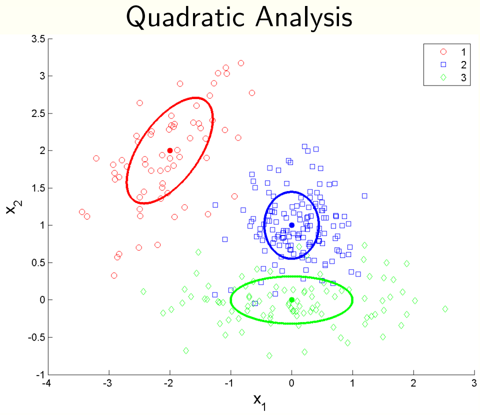
Linear Analysis



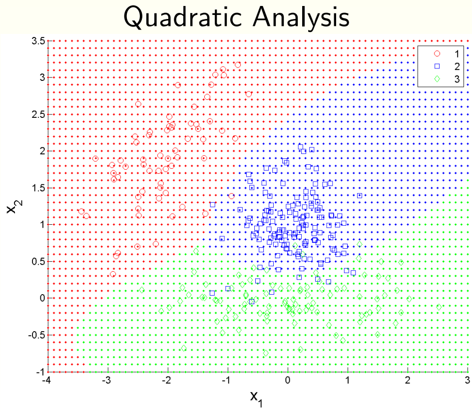
Quadratic Discriminant Analysis (QDA)



Quadratic Discriminant Analysis (QDA)

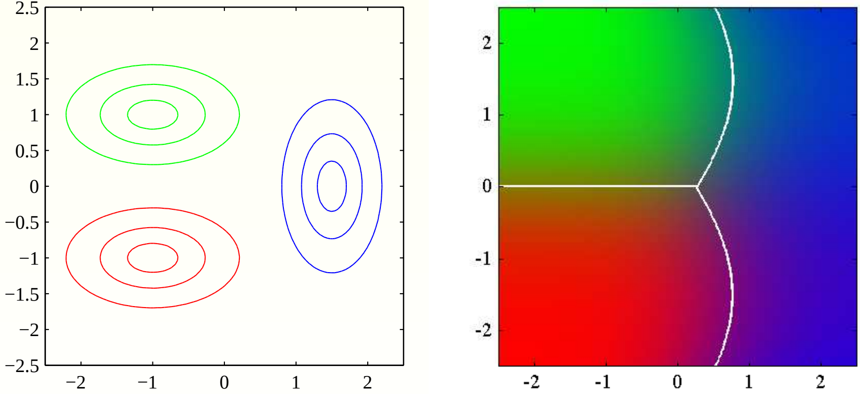


Quadratic Discriminant Analysis (QDA)



Quadratic Discriminant Analysis (QDA)

3 Gaussian classes, $\Sigma_1 = \Sigma_2 \neq \Sigma_3$



Bayesian naive approach

Example : prediction of the power of EDF counter

- **Objective :** predict the subscribed power (3, 6, 9 ou 12 kWh) from 3 binary informations.
- K = 4 classes for 3, 6, 9 ou 12 kWh
- 3 binary features $\mathbf{x} = (x_1, x_2, x_3)$, $x_i \in \{0, 1\}$,
  1. Electrical heating / other x_1
  2. House / Flat x_2
  3. Drying machine / no x_3
- **Naive assumption :** $p(x|\mathcal{C}_k) = \prod_{i=1}^{D} \pi_{ki}^{x_i} (1 - \pi_{ki})^{1-x_i}$
  - N training data $\mathcal{D} = \{(x_n, t_n), 1 \leq n \leq N\}$,
  - N_k entries in class $\mathcal{C}_k$,
  - n_{ki} entries in class $\mathcal{C}_k$ such that $x_i = 1$,

$$\begin{cases} \widehat{p(\mathcal{C}_k)} & = & \dfrac{N_k}{N} \\ \hat{\pi}_{ki} & = & \dfrac{n_{ki}}{N_k} \quad \text{(Bernoulli)} \end{cases}$$

$$
\begin{aligned}
f(\mathbf{x}) &= \operatorname{argmax}_k \ln \widehat{p(\mathbf{x}|\mathcal{C}_k)} + \ln \widehat{p(\mathcal{C}_k)} \\
&= \operatorname{argmax}_k \sum_{i=1}^{D} \ln \widehat{g_{ki}(x_i)} + \ln \frac{N_k}{N} \\
&= \operatorname{argmax}_k y_k(\mathbf{x})
\end{aligned}
$$

where

$$
y_k(\mathbf{x}) = \sum_{i=1}^{D} \left[x_i \ln \frac{n_{ki}}{N_k} + (1 - x_i) \ln(1 - \frac{n_{ki}}{N_k}) \right] \; + \; \ln \frac{N_k}{N}
$$

Bayesian naive approach

Case of binary variables (Bernoulli)

$$f(\mathbf{x}) = \mathrm{argmax}_k y_k(\mathbf{x})$$

where

$$y_k(\mathbf{x}) = \sum_{i=1}^{D} \left[x_i \ln \frac{n_{ki}}{N_k} + (1 - x_i) \ln(1 - \frac{n_{ki}}{N_k}) \right] + \ln \frac{N_k}{N}$$

- linear in $\mathbf{x}$: **boundaries = hyperplanes**
- generalises to any kind of variables x_i :
  - parametric models (normal laws $\Rightarrow \hat{\mu}_k, \hat{\sigma}_k$)
  - histograms,
  - kernel estimate of densities (Parzen's kernel : Gaussian kernel)

In summary :

- naive but often efficient (strong bias / low variance)
- useful in large dimension problems in particular.

- **Objective :** predict the subscribed power (3, 6, 9 ou 12 kWh) from 3 binary informations.
- K = 4 classes for 3, 6, 9 ou 12 kWh
- 3 binary features $\mathbf{x} = (x_1, x_2, x_3)$, $x_i \in \{0, 1\}$,
  1. Electrical heating / other x_1
  2. House / Flat x_2
  3. Drying machine / no x_3
- **Naive assumption :** $p(x|\mathcal{C}_k) = \prod_{i=1}^{D} \pi_{ki}^{x_i} (1 - \pi_{ki})^{1-x_i}$
  - N training data $\mathcal{D} = \{(x_n, t_n), 1 \leq n \leq N\}$,
  - N_k entries in class $\mathcal{C}_k$,
  - n_{ki} entries in class $\mathcal{C}_k$ such that $x_i = 1$,

$$\begin{cases} \widehat{p(\mathcal{C}_k)} & = & \dfrac{N_k}{N} \\ \hat{\pi}_{ki} & = & \dfrac{n_{ki}}{N_k} \quad \text{(Bernoulli)} \end{cases}$$

Bayesian naive approach

General case : laws g_{ki} such that $\ln g_{ki}(x_i)$= non linear function (x_i)

$$f(\mathbf{x}) = \text{argmax}_k y_k(\mathbf{x})$$

where

$$y_k(\mathbf{x}) = \sum_{i=1}^{D} \ln g_{ki}(\mathbf{x})$$

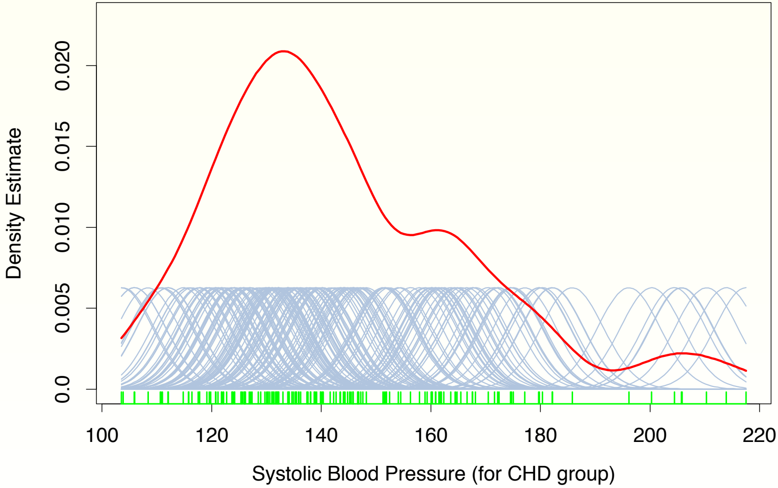
- non-linear in $\mathbf{x} \Rightarrow$ **boundaries $\neq$ hyperplanes**
- generalises to any kind of variables x_i :
  - parametric models (normal laws $\Rightarrow \hat{\mu}_k,\ \hat{\sigma}_k$)
  - histograms,
  - kernel estimate of densities (Parzen's kernel : Gaussian kernel)

In summary :

- naive but often efficient (strong bias / low variance)
- useful in large dimension problems in particular.

Bayesian naive approach

Kernel based estimate of a density



Bayesian naive approach
In summary

- Classification method **naive** = ignoring correlations between x_i

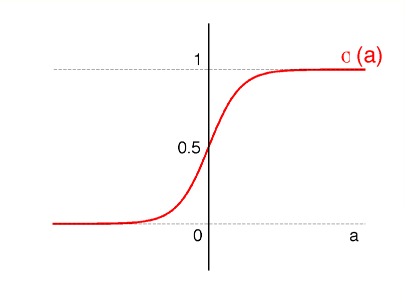
$$\forall 1 \leq k \leq K, \quad p(x_1, ..., x_D|\mathcal{C}_k) \simeq \prod_{i=1}^{D} p(x_i|\mathcal{C}_k)$$

- **Main advantages :**
  - $p(\mathbf{x}|\mathcal{C}_k)$ is rich but complex & difficult to access,
  - $p(x_i|\mathcal{C}_k)$ is rough but simple to estimate
- **Case of binary variables :**
  - joint proba. $p(\mathbf{x}|\mathcal{C}_k) \Rightarrow K \cdot 2^D$ quantities to estimate,
  - marginal proba. $p(x_i|\mathcal{C}_k) \Rightarrow K \cdot D$ quantities to estimate,
- **Interest** : very useful simplification if D is very large
- Can generalize to any kind of laws $p(x_i|\mathcal{C}_k)$
- Example : EDF counters...

Trajectory

Logistic regression

Principle : translation of the estimation of a 'conviction' degree (binary case)



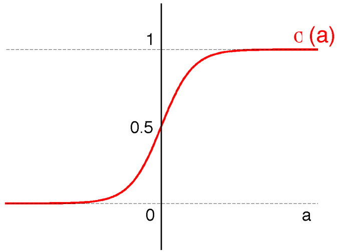
- Sigmoid logistic function :

$$\sigma(a) = \frac{1}{1+\exp(-a)} = \frac{\exp(a)}{1+\exp(a)}$$

- Inverse = function *logit* : $a = \ln\left(\frac{\sigma}{1-\sigma}\right) = \text{logit}(\sigma)$

Logistic regression

Principle : translation of the estimation of a 'conviction' degree (binary case)



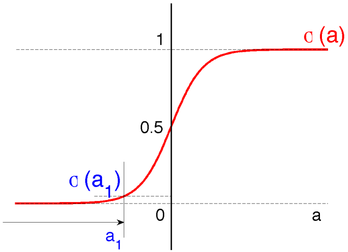
- One looks for an *activation* which linearly depends on features :

$$a = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$$

- Model : the decision will depend on $p(\mathcal{C}_1|\mathbf{x}) = \sigma(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}})$.

Logistic regression

Principle : translation of the estimation of a 'conviction' degree (binary case)



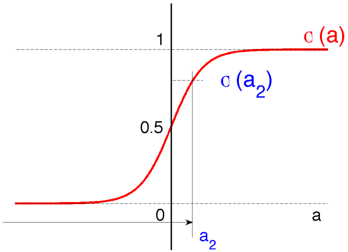
- One looks for an *activation* which linearly depends on features :

$$a = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$$

- Model : the decision will depend on $p(\mathcal{C}_1|\mathbf{x}) = \sigma(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}})$.

Logistic regression

Principle : translation of the estimation of a 'conviction' degree (binary case)



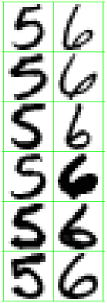
- One looks for an *activation* which linearly depends on features :

$$a = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$$

- Model : the decision will depend on $p(\mathcal{C}_1|\mathbf{x}) = \sigma(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}})$.

Logistic regression

Example : recognition of manuscript figures



- Recognition of manuscript figures 5 & 6 :

  $t_n \in 5, 6$

- Extraction of significant pixels :

  with standard deviation greater than 0,5

- $\mathcal{D} = \{X_n \in \mathbb{R}^{173},\ t_n \in 5, 6\}$, $N = 345$ examples,

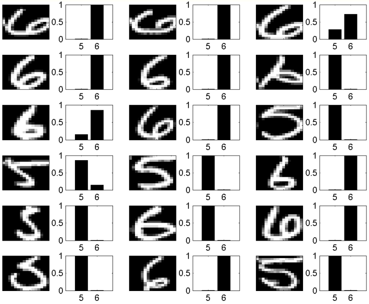
- Estimate of $\mathbf{w} \in \mathbb{R}^{174}$ using logistic regression

Probability of belonging to a class on test set :

$$p(t=5|\mathbf{x}) = \sigma(\tilde{\mathbf{w}}^T\tilde{\mathbf{x}}) \tag{1}$$

$$= \frac{1}{1+\exp(-\tilde{\mathbf{w}}^T\tilde{\mathbf{x}})} \tag{2}$$

$$= \frac{1}{1+\exp(-w_0 - \sum_{i=1}^{173} w_i x_i)} \tag{3}$$



Iterated Reweighted Least Squares (IRLS)

$X =$ matrix of features for learning (+ col. of 1), dim. $D+1$,
$\mathbf{t}$ =vector of targets for the training set,

$$\begin{aligned} \tilde{\mathbf{w}}^{(old)} &= \tilde{\mathbf{w}} &&= \text{zeros}(D+1, 1) \\ \mathbf{y} &= \sigma(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}) &&= \tfrac{1}{2}\text{ones}(N_{train}, 1) \\ R &= \text{diag}(y_n(1-y_n)) &&= \text{diag}(1/4) \end{aligned}$$

$\mathbf{z} = X\tilde{\mathbf{w}}^{(old)} - R^{-1}(\mathbf{y} - \mathbf{t})$
$\tilde{\mathbf{w}} = (X^T R X)^{-1} X^T R \mathbf{z}$

While ($(||\tilde{\mathbf{w}} - \tilde{\mathbf{w}}^{(old)}||/||\tilde{\mathbf{w}}|| > \varepsilon)$ and (max number of iterations))

$\mathbf{y} = \sigma(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}})$
$R = \text{diag}(y_n(1-y_n))$
$\tilde{\mathbf{w}}^{(old)} = \tilde{w}$
$\mathbf{z} = X\tilde{\mathbf{w}}^{(old)} - R^{-1}(\mathbf{y} - \mathbf{t})$
$\tilde{\mathbf{w}} = (X^T R X)^{-1} X^T R \mathbf{z}$

End of While

Target variable : presence or absence of myocard failure

Features :

sbp	systolic blood pressure
tobacco	cumulative tobacco (kg)
ldl	low density lipoprotein cholesterol
adiposity	index
famhist	family history of heart disease (yes/no)
typea	type-A behavior
obesity	index
alcohol	current alcohol consumption
age	age at onset
chd	response, coronary heart disease

from the example South African Heart Disease, Hastie & Tibshirani

	Coefficient	Std. Error	Z Score
`(Intercept)`	-4.130	0.964	-4.285
`sbp`	0.006	0.006	1.023
`tobacco`	0.080	0.026	3.034
`ldl`	0.185	0.057	3.219
`famhist`	0.939	0.225	4.178
`obesity`	-0.035	0.029	-1.187
`alcohol`	0.001	0.004	0.136
`age`	0.043	0.010	4.184

	Coefficient	Std. Error	Z score
`(Intercept)`	-4.204	0.498	-8.45
`tobacco`	0.081	0.026	3.16
`ldl`	0.168	0.054	3.09
`famhist`	0.924	0.223	4.14
`age`	0.044	0.010	4.52

e.g. tobacco = in kg consumed

$$+\ 1\text{kg} \Longrightarrow \frac{p(\mathcal{C}_1|\mathbf{x})}{p(\mathcal{C}_2|\mathbf{x})} \times \exp(0,081) = 1,084$$

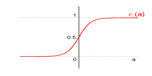
that is an increase in risk of 8,4%.

Taking into account the uncertainty at level 95% :

$$\exp(0,081 \pm 2 \times 0.026) = [1,03, 1.14].$$

Logistic regression
In summary

- Classification method (decision) as a function of an activation :



$a \Rightarrow \sigma(a)$ conviction degree

- $a = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$ linear combination of features,
- Interpretation of weights $\tilde{\mathbf{w}}$ = influence of features,
- Often used in biology, medecine, human sciences...
- Generalizes to K classes and features $\phi_j(\mathbf{x})$,
- Logistic regression *vs* Discriminant Analysis (LDA, QDA) :
  - results are often comparable,
  - LDA/QDA : more sensitive to extreme events,
  - Warning : if the data are linearly separable,

    log. reg. $\Rightarrow \tilde{\mathbf{w}} \to \infty$! ! !