Renaud Sirdey

# Discrimination paramétrique

Éditions des Nik's News
*www.niksnews.com/editions/*

1998

# Introduction

Ce travail consiste en l'application et l'étude des méthodes de discrimination paramétrique sous l'hypothèse gaussienne. Dans un premier temps, une brève description des données utilisées est présentée. Par la suite, on étudie les méthodes paramétriques sous différentes hypothèses gaussienne plus ou moins contraintes. Enfin, une attention particulière est portée au comportement de la règle de décision quadratique dans l'espace des distances de Mahalanobis.

# Table des matières

# Chapitre 1

# Etude des données

## 1.1 Description des données

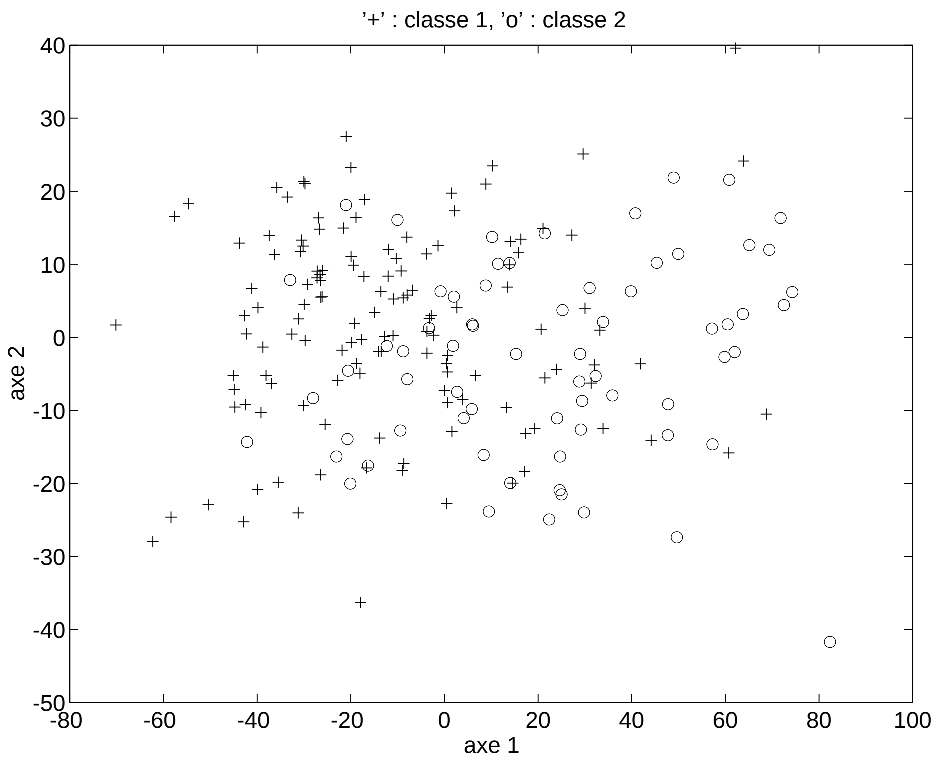
Il s'agit de données issues d'une étude sur le diabète effectuée auprès de femmes agées d'au moins 21 ans et appartenant à la tribu des indiens Pima (près de Phoenix, Arizona). Il y a 7 variables, plus la classe

- `npreg` : nombre de grossesses ;
- `glu` : taux de glucose dans le plasma ;
- `bp` : pression artérielle ;
- `skin` : épaisseur de la peau au niveau des triceps ;
- `bmi` : indice de masse corporelle ;
- `ped` : « diabete pedigree function » ;
- `age` : âge ;
- `classe` : 1 absence de diabète, 2 présence de diabète.

## 1.2 Analyse en composantes principales

La représentation des données d'apprentissage dans le plan des deux premiers axes donnés par l'analyse en composantes principales (voir figure 1.1) permet de faire correctement apparaitre les deux classes.

Fig. **1.1** – Représentation de l'ensemble d'apprentissage dans le plan des deux premiers axes donnés par l'analyse en composantes principales.



# Chapitre 2

# Discrimination paramétrique dans le cas gaussien

## 2.1 Règles de décision

Il s'agit d'un problème à deux classes $\omega_1$ et $\omega_2$, on se place dans le cadre suivant

- $f(x/\omega_1) = \mathcal{N}(\mu_1, \Sigma_1)$ ;
- $f(x/\omega_2) = \mathcal{N}(\mu_2, \Sigma_2)$.

La règle de décision optimale pour des coûts $\{0, 1\}$ est la règle de Bayes, à savoir qu'on décide que $x \in \omega_1$ si

$$\frac{f(x/\omega_1)}{f(x/\omega_2)} > \frac{\pi(\omega_2)}{\pi(\omega_1)}$$

où $\pi(\omega_1)$ et $\pi(\omega_2)$ représentent la connaissance a priori que l'on a de la proportion d'individus qui appartiennent respectivement à $\omega_1$ et à $\omega_2$ dans la population étudiée.

En général, on cherche une fonction discriminante $g(x)$ telle que

$$d_x = \begin{cases} 1 \text{ si } g(x) > 0 \\ 2 \text{ sinon} \end{cases}$$

On choisira

$$g(x) = g_1(x) - g_2(x)$$

avec, par exemple

- $g_1(x) = \log(f(x/\omega_1)) + \log(\pi(\omega_1))$ ;
- $g_2(x) = \log(f(x/\omega_2)) + \log(\pi(\omega_2))$.

### 2.1.1 Cas $\Sigma_1 = \Sigma_2 = \sigma^2 I$

D'après un calcul fait en cours, on a

- $g_1(x) = \frac{1}{\sigma^2}\mu_1^T x - \frac{1}{2\sigma^2}\mu_1^T\mu_1 + \log(\pi(\omega_1))$ ;
- $g_2(x) = \frac{1}{\sigma^2}\mu_2^T x - \frac{1}{2\sigma^2}\mu_2^T\mu_2 + \log(\pi(\omega_2))$.

### 2.1.2 Cas $\Sigma_1 = \Sigma_2 = \Sigma$

D'après un calcul fait en cours, on a

- $g_1(x) = (\Sigma^{-1}\mu_1)^T x - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \log(\pi(\omega_1))$ ;
- $g_2(x) = (\Sigma^{-1}\mu_2)^T x - \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + \log(\pi(\omega_2))$.

### 2.1.3 Cas $\Sigma_1$ et $\Sigma_2$ quelconques

D'après un calcul fait en cours, on a

- $g_1(x) = -\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1) - \frac{1}{2}\log(|\Sigma_1|) + \log(\pi(\omega_1))$ ;
- $g_2(x) = -\frac{1}{2}(x-\mu_2)^T\Sigma_2^{-1}(x-\mu_2) - \frac{1}{2}\log(|\Sigma_2|) + \log(\pi(\omega_2))$.

## 2.2 Résultats et mise en oeuvre

L'ensemble d'apprentissage est composé de $n = 200$ individus répartis en $n_1 = 132$ individus de la classe $\omega_1$ et $n_2 = 68$ de la classe $\omega_2$.

Dans un premier temps, on estime les probabilités a priori

- $\hat{\pi}(\omega_1) = \frac{n_1}{n} = \frac{132}{200} = 0.66$ ;
- $\hat{\pi}(\omega_2) = \frac{n_2}{n} = \frac{68}{200} = 0.34$.

De la même façon on estime les moyennes de chaque classe

$$\bar{x}_1 = \begin{pmatrix} 2.91 \\ 113.10 \\ 69.54 \\ 27.20 \\ 31.07 \\ 0.41 \\ 29.23 \end{pmatrix} \text{ et } \bar{x}_2 = \begin{pmatrix} 4.83 \\ 145.05 \\ 74.58 \\ 33.11 \\ 34.70 \\ 0.54 \\ 37.69 \end{pmatrix}$$

### 2.2.1 Cas $\Sigma_1 = \Sigma_2 = \sigma^2 I$

#### Estimation de $\sigma^2$

Estimateur

$$\hat{\sigma}^2 = \frac{1}{MN} \sum_{k=1}^{M} \sum_{i=1}^{N_k} (x_k^i - \bar{x}_k)^T (x_k^i - \bar{x}_k)$$

Estimation [1]

$$\hat{\sigma}^2 = 175.37$$

#### Mise en oeuvre

Voir pages 23 et 23.

#### Résultats sur l'ensemble de test

Estimation ponctuelle de la probabilité d'erreur

$$\hat{p}_\epsilon = 0.2259$$

Estimation par intervalle de confiance

Une utilisation très approximative des tables d'intervalles de confiance de la loi binomiale [2] donne

$$P(0.15 \leq \hat{p} \leq 0.25) \approx 0.95$$

pour $n = 250$ et $\hat{p} = 0.2$.

En utilisant les résultats du calcul présenté page 14 on obtient

$$P(0.1836 \leq \hat{p} \leq 0.2747) = 0.95$$

Le tableau de contingence 2.1 présente les résultats de classification.

### 2.2.2 Cas $\Sigma_1 = \Sigma_2 = \Sigma$

#### Estimation de $\Sigma$

Estimateur

$$\hat{\Sigma} = \frac{1}{N} \sum_{k=1}^{M} N_k \hat{\Sigma}_k$$

1. Ici et dans tout ce qui suit on ne se préoccupe pas des problèmes de biais pour les estimateurs des moments d'ordre 2. L'ensemble d'apprentissage est considéré suffisament grand.
2. L'utilisation d'une loi binomiale est justifiée page 14.

| | $d(x) = 1$ | $d(x) = 2$ |
|---|---|---|
| $x \in \omega_1$ | 192 | 31 |
| $x \in \omega_2$ | 44 | 65 |

TAB. 2.1 – Résultats de la classification sous l'hyposthèse $\Sigma_1 = \Sigma_2 = \sigma^2 I$.

| | $d(x) = 1$ | $d(x) = 2$ |
|---|---|---|
| $x \in \omega_1$ | 198 | 25 |
| $x \in \omega_2$ | 43 | 66 |

TAB. 2.2 – Résultats de la classification sous l'hyposthèse $\Sigma_1 = \Sigma_2$.

avec $N = \sum_{k=1}^{N} N_k$ Estimation

$$\hat{\Sigma} = \begin{pmatrix} 10.56 & 4.32 & 7.59 & 1.74 & -0.37 & -0.18 & 18.56 \\ 4.32 & 776.77 & 61.80 & 38.49 & 15.95 & -0.37 & 58.71 \\ 7.59 & 61.80 & 126.70 & 29.06 & 12.71 & -0.32 & 39.82 \\ 1.74 & 38.49 & 29.06 & 130.28 & 42.70 & 0.16 & 21.26 \\ -0.37 & 15.95 & 12.71 & 42.70 & 34.74 & 0.25 & 1.92 \\ -0.18 & -0.37 & -0.32 & 0.16 & 0.25 & 0.09 & -0.49 \\ 18.56 & 58.71 & 39.82 & 21.26 & 1.92 & -0.49 & 104.92 \end{pmatrix}$$

### Mise en oeuvre

Voir pages 23 et 25.

### Résultats sur l'ensemble de test

Estimation ponctuelle de la probabilité d'erreur

$$\hat{p}_\epsilon = 0.2048$$

Estimation par intervalle de confiance

$$P(0.1643 \leq p \leq 0.2522) = 0.95$$

Le tableau de contingence 2.2 présente les résultats de classification.

### 2.2.3 Cas $\Sigma_1$ et $\Sigma_2$ quelconques

#### Estimation des $\Sigma_k$

Estimateur

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} (x_k^i - \bar{x}_k)(x_k^i - \bar{x}_k)^T$$

Estimations

$$\hat{\Sigma}_1 = \begin{pmatrix} 7.87 & 10.77 & 8.19 & 2.91 & -0.03 & -0.20 & 16.82 \\ 10.77 & 709.56 & 81.43 & 13.23 & 19.03 & -0.51 & 59.01 \\ 8.19 & 81.43 & 122.84 & 33.87 & 16.61 & -0.07 & 46.78 \\ 2.91 & 13.23 & 33.87 & 119.44 & 50.12 & 0.07 & 18.47 \\ -0.03 & 19.03 & 16.61 & 50.12 & 40.72 & 0.14 & 7.00 \\ -0.20 & -0.51 & -0.07 & 0.07 & 0.14 & 0.07 & -0.53 \\ 16.82 & 59.01 & 46.78 & 18.47 & 7.00 & -0.53 & 91.08 \end{pmatrix}$$

$$\hat{\Sigma}_2 = \begin{pmatrix} 15.77 & -8.19 & 6.42 & -0.51 & -1.03 & -0.13 & 21.93 \\ -8.19 & 907.25 & 23.71 & 87.51 & 9.98 & -0.08 & 58.12 \\ 6.42 & 23.71 & 134.18 & 19.70 & 5.15 & -0.79 & 26.30 \\ -0.51 & 87.51 & 19.70 & 151.32 & 28.28 & 0.34 & 26.67 \\ -1.03 & 9.98 & 5.15 & 28.28 & 23.14 & 0.45 & -7.91 \\ -0.13 & -0.08 & -0.79 & 0.34 & 0.45 & 0.12 & -0.41 \\ 21.93 & 58.12 & 26.30 & 26.67 & -7.91 & -0.41 & 131.79 \end{pmatrix}$$

#### Mise en oeuvre

Voir pages 23 et 26.

#### Résultats sur l'ensemble de test

Estimation ponctuelle de la probabilité d'erreur

$$\hat{p}_\epsilon = 0.2289$$

Estimation par intervalle de confiance

$$P(0.1864 \leq p \leq 0.2778) = 0.95$$

Le tableau de contingence 2.3 présente les résultats de classification.

| | $d(x) = 1$ | $d(x) = 2$ |
|---|---|---|
| $x \in \omega_1$ | 194 | 29 |
| $x \in \omega_2$ | 47 | 62 |

TAB. 2.3 – Résultats de la classification sous l'hyposthèse $\Sigma_1$ et $\Sigma_2$ quelconques.

## 2.3 Conclusion

Finalement, en termes d'estimation de probailité d'erreur, les trois règle se comporte de manière assez identique. La deuxième hypothèse semble avoir un léger avantage sur les deux autres, mais comme la comparaison est seulement basée sur l'estimation de la probabilité d'erreur à partir de l'ensemble d'apprentissage, on se gardera de conclure trop vite !

Cependant, il est nécessaire d'adapter les hypothèses à la taille de l'ensemble d'apprentissage. En effet, l'utilisation de la $3^{iéme}$ hypothése avec un ensemble d'apprentissage relativement petit conduira à une règle de décision instable (puisque les estimations ne seront pas de très bonne qualité). Ceci explique peut être les résultats un peu moins bon de la règle quadratique, puisque la matrice de variance-covariance dans la classe $\omega_2$ a été estimée à partir d'un échantillon de taille 68 (ce qui est relativement petit).

## 2.4 Calcul des bornes de l'intervalle de confiance à 95%

On s'intéresse à la variable aléatoire $E$ qui correspond au nombre de fois que l'on est susceptible de se tromper sur un ensemble de test de taille $n$. Comme cette variable correspond à une somme de schémas de Bernouilli (« se tromper sur un individu ou pas ») elle suit une loi binomiale.

$$E \rightsquigarrow \mathcal{B}(n, p)$$

où $p$ est la probabilité d'erreur que l'on cherche à estimer.

On montre que l'estimateur du maximum de vraisemblance de $p$ est

$$\hat{p} = \frac{E}{n}$$

et que $E[\hat{p}] = p$ et $Var[\hat{p}] = \frac{p(1-p)}{n}$.

On cherche les bornes d'un intervalle telles que

$$P(p \in [\alpha_1, \alpha_2]) = 0.95$$

D'après le théorème central limite, on a

$$\hat{p} \rightsquigarrow \mathcal{N}\left(p, \frac{p(1-p)}{n}\right),\ n \to +\infty$$

donc

$$\frac{\hat{p} - p}{\sqrt{\frac{p(1-p)}{n}}} \rightsquigarrow \mathcal{N}(0,1)$$

Dans un premier temps, on cherche à trouver $u$ tel que

$$P\left(-u \leq \frac{\hat{p} - p}{\sqrt{\frac{p(1-p)}{n}}} \leq u\right) = 0.95$$

soit $u = 1.96$ d'après les fractiles de la loi normale.

Les bornes de l'intervalle sont donc les solutions de

$$-u\sqrt{\frac{p(1-p)}{n}} \leq p - \hat{p} \leq u\sqrt{\frac{p(1-p)}{n}}$$

et donc de l'inéquation

$$(p - \hat{p})^2 \leq u^2 \frac{p(1-p)}{n}$$

D'où

$$\begin{aligned} (p - \hat{p})^2 - u^2 \frac{p(1-p)}{n} &\leq 0 \\ p^2 \underbrace{\left(1 + \frac{u^2}{n}\right)}_{a} - p \underbrace{\left(2\hat{p} + \frac{u^2}{n}\right)}_{-b} + \underbrace{\hat{p}^2}_{c} &\leq 0 \end{aligned}$$

Les bornes recherchées sont donc les solutions de l'équation $ap^2 + bp + c = 0$.

Soit

$$\begin{aligned} \Delta &= b^2 - 4ac \\ &= \left(2\hat{p} + \frac{u^2}{n}\right)^2 - 4\left(1 + \frac{u^2}{n}\right)\hat{p}^2 \\ &= 4\hat{p}^2 + 4\hat{p}\frac{u^2}{n} + \frac{u^4}{n^2} - 4\hat{p} - 4\hat{p}^2\frac{u^2}{n} \\ &= 4\hat{p}\frac{u^2}{n}\underbrace{(1-\hat{p})}_{\geq 0} + \frac{u^4}{n^2} \geq 0 \end{aligned}$$

Les bornes de l'intervalle de confiance vérifient alors

$$\begin{aligned}\alpha_{1,2} &= \frac{-b \pm \sqrt{\Delta}}{2a} \\ &= \frac{2\hat{p} + \frac{u^2}{n} \pm \sqrt{4\hat{p}\frac{u^2}{n}(1-\hat{p}) + \frac{u^4}{n^2}}}{2\left(1 + \frac{u^2}{n}\right)}\end{aligned}$$

# Chapitre 3

# Représentation dans $< d_1(x), d_2(x) >$

On choisit de représenter les données dans l'espace des variables

- $d_1(x) = (x - \hat{\mu}_1)^T \hat{\Sigma}_1^{-1} (x - \hat{\mu}_1)$ ;
- $d_2(x) = (x - \hat{\mu}_2)^T \hat{\Sigma}_2^{-1} (x - \hat{\mu}_2)$.

et d'étudier le comportement de la règle quadratique, mise en oeuvre page 13, dans ce nouvel espace.

## 3.1 Frontière de décision

La règle de décision quadratique consiste en

- choisir $x \in \omega_1$ si $\hat{g}_1(x) > \hat{g}_2(x)$ ;
- choisit $x \in \omega_2$ sinon.

Les fonctions $\hat{g}_1(x)$ et $\hat{g}_2(x)$ étant les estimations des fonctions présentées page 10.

Maintenant, on décide $x \in \omega_1$ si

$$-\frac{1}{2} d_1(x) + \alpha_{\omega_1} > -\frac{1}{2} d_2(x) + \alpha_{\omega_2}$$

où

- $\alpha_{\omega_1} = -\log(|\hat{\Sigma}_1|) + \log(\hat{\pi}(\omega_1))$ ;
- $\alpha_{\omega_2} = -\log(|\hat{\Sigma}_2|) + \log(\hat{\pi}(\omega_2))$.

ce qui revient à décider $x \in \omega_1$ si

$$d_2(x) > d_1(x) + 2(\alpha_{\omega_2} - \alpha_{\omega_1})$$

La frontière de décision correspond donc à la droite d'équation

$$d_2(x) = d_1(x) + 2(\alpha_{\omega_2} - \alpha_{\omega_1})$$

ou encore

$$z = t + 2(\alpha_{\omega_2} - \alpha_{\omega_1})$$

Les figures 3.1 et 3.2 représentent l'ensemble d'apprentissage dans $< z, t >$ ainsi que la frontière de décision. Les figures 3.3 et 3.4, quant à elles, représentent l'ensemble de test dans le même espace.

FIG. **3.1** – Représentation de l'ensemble d'apprentissage dans $< d_1(x), d_2(x) >$.

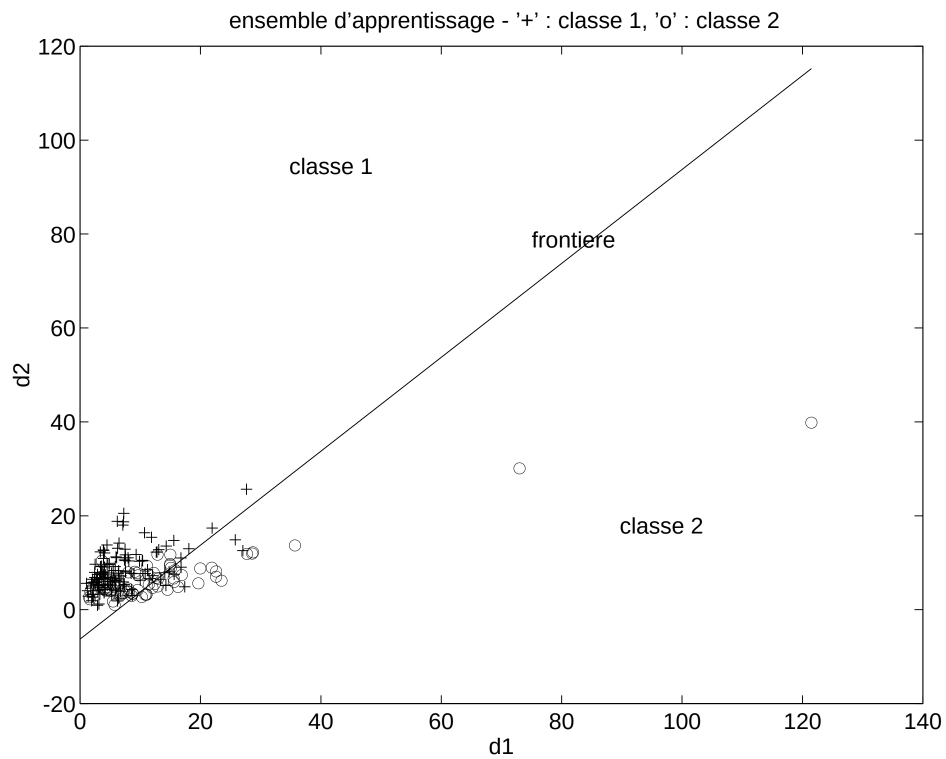


Fig. **3.2** – Zoom sur la figure précédente.

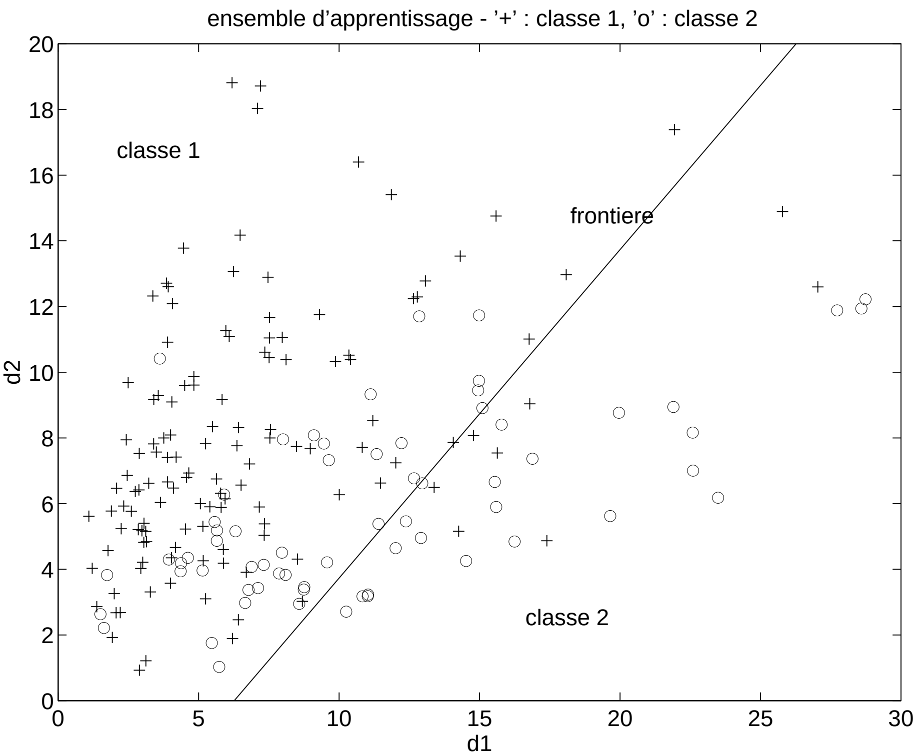


FIG. **3.3** – Représentation de l'ensemble de test dans $< d_1(x), d_2(x) >$.

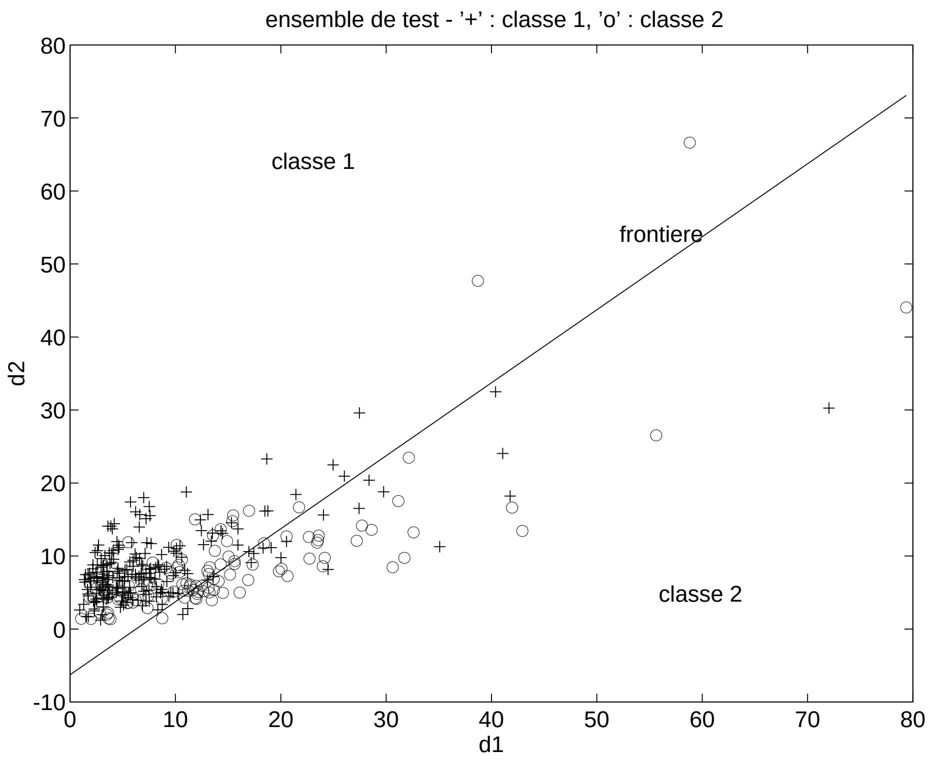
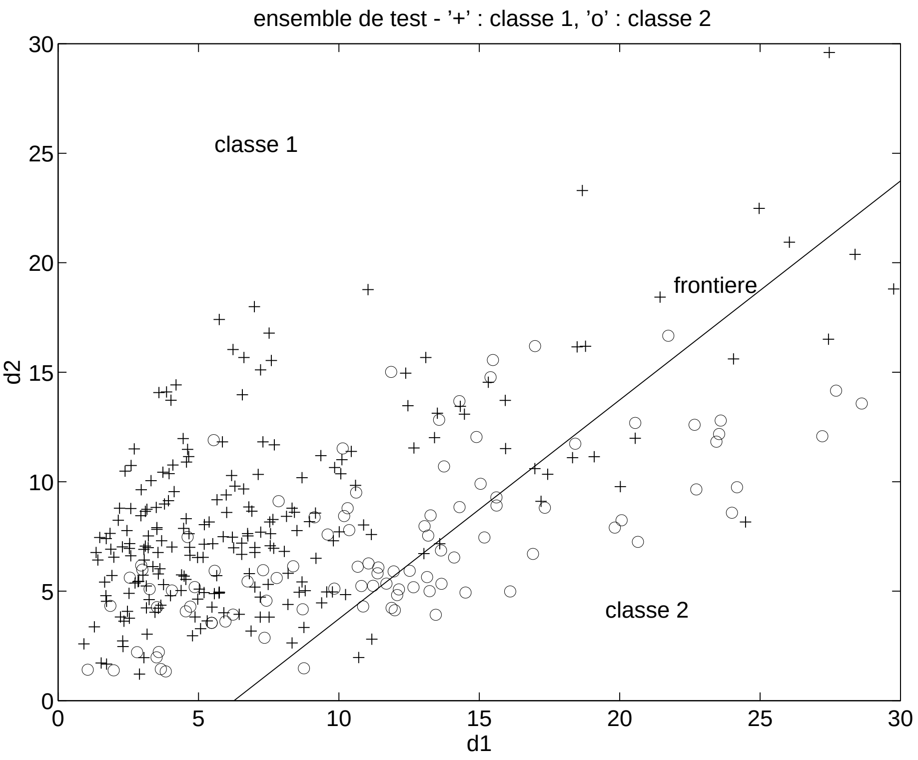


FIG. **3.4** – Zoom sur la figure précédente.



## 3.2 Ammélioration « manuelle »

Le fait d'estimer la règle optimal de Bayes ne garantit pas d'obtenir un taux d'erreur minimal sur l'ensemble d'apprentissage, puisque cette estimation dépend de celle des moments et de l'hypothèse gaussienne.
Il est donc intéressant d'ajuster manuellement la frontière de décision ($z = t + 2(\alpha_{\omega_2} - \alpha_{\omega_1})$) afin de séparer au mieux les deux classes (c'est-à-dire d'avoir une erreur minimale sur l'ensemble d'apprentissage).

On obtient alors une régle de décision « corrigée », et on décide $x \in \omega_1$ si

$$d_2(x) > \lambda_1 d_1(x) + 2\lambda_2(\alpha_{\omega_2} - \alpha_{\omega_1})$$

où $\lambda_1$ et $\lambda_2$ sont les coefficients d'ajustement déterminés empiriquement sur l'ensemble d'apprentissage.

On remarque cependant que la frontière de décision linéaire optimale (au sens de l'erreur quadratique moyenne) peut être obtenue de manière systématique dans le cadre de l'utilisation d'un réseau de neuronnes unidirectionnel sans couche cachée.

# Annexe A

# Routines

## A.1 Paramétrage

### A.1.1 Entrées

- $l$ : ensemble d'apprentissage ;
- $x$ : ensemble de test.

### A.1.2 Sorties

- $d$ : « vecteur décision » ;
- $g$ : résultat de l'application de la fonction discriminante ;
- $\hat{p}_\epsilon$ : estimation de la probabilité d'erreur ;
- $n_{\epsilon_1}$ : nombre de fois que l'on a décidé $\omega_1$ à la place de $\omega_2$ ;
- $n_{\epsilon_2}$ : nombre de fois que l'on a décidé $\omega_2$ à la place de $\omega_1$.

## A.2 Cas $\Sigma_1 = \Sigma_2 = \sigma^2 I$

```
function [d,g,p_err,nE1,nE2]=case1(l,x)

% separation des individus de chaque classe
    l1 = l(find(l(:,size(l,2))==1),:);
    l2 = l(find(l(:,size(l,2))==2),:);

% estimation des moyennes
```

```
    m1 = mean(l1(:,1:size(l1,2)-1))´;
    m2 = mean(l2(:,1:size(l2,2)-1))´;

% estimation de la variance
    sig1 = mean(var(l1(:,1:size(l,2)-1)));
    sig2 = mean(var(l2(:,1:size(l,2)-1)));
    sig  = mean([sig1;sig2]);

% parametres des fonctions discriminantes
    p1 = size(l1,1)/size(l,1);
    a1 = inv(sig) * m1´;
    b1 = - inv(2*sig) * m1´*m1 + log(p1);

    p2 = size(l2,1)/size(l,1);
    a2 = inv(sig) * m2´;
    b2 = - inv(2*sig) * m2´*m2 + log(p2);

% calcul de la fonction discriminante pour chaque individus
    buf = x(:,1:size(x,2)-1);
    g = a1 * buf´ + b1 - a2 * buf´ - b2;

% calcul du vecteur des decisions
    g = g´;
    d = zeros(size(g));
    f1 = find(g>0);
    d(find(g>0))=ones(size(f1));
    f2 = find(g<=0);
    d(find(g<=0))=2*ones(size(f2));

% estimation de la probabilite d´erreur
    p_err = size(find(d~=x(:,size(x,2))),1) / size(x,1);

% "decide 1 a la place de 2"
    nE1 = size(find(d==1 & x(:,size(x,2))==2),1);
% "decide 2 a la place de 1"
    nE2 = size(find(d==2 & x(:,size(x,2))==1),1);
```

## A.3 Cas $\Sigma_1 = \Sigma_2 = \Sigma$

```
function [d,g,p_err,nE1,nE2]=case2(l,x)

% separation des individus de chaque classe
    l1 = l(find(l(:,size(l,2))==1),:);
    l2 = l(find(l(:,size(l,2))==2),:);

% estimation des moyennes
    m1 = mean(l1(:,1:size(l1,2)-1))´;
    m2 = mean(l2(:,1:size(l2,2)-1))´;

% estimation de la matrice de covariance
    sig1 = cov(l1(:,1:size(l1,2)-1));
    sig2 = cov(l2(:,1:size(l2,2)-1));
    sig  = (size(l1,1)*sig1 + size(l2,1)*sig2) / size(l,1);

% parametres des fonctions discriminates
    p1 = size(l1,1) / size(l,1);
    a1 = (inv(sig) * m1)´;
    b1 = -.5 * m1´ * inv(sig) * m1 + log(p1);

    p2 = size(l2,1) / size(l,1);
    a2 = (inv(sig) * m2)´;
    b2 = -.5 * m2´ * inv(sig) * m2 + log(p2);

% calcul de la fonction discriminante pour chaque individus
    buf = x(:,1:size(x,2)-1);
    g = a1 * buf´ + b1 - a2 * buf´ - b2;

% calcul du vecteur des decisions
    g = g´;
    d = zeros(size(g));
    f1 = find(g>0);
    d(find(g>0))=ones(size(f1));
    f2 = find(g<=0);
    d(find(g<=0))=2*ones(size(f2));

% estimation de la probabilite d´erreur
```

```
    p_err = size(find(d~=x(:,size(x,2))),1) / size(x,1);

% "decide 1 a la place de 2"
    nE1 = size(find(d==1 & x(:,size(x,2))==2),1);
% "decide 2 a la place de 1"
    nE2 = size(find(d==2 & x(:,size(x,2))==1),1);
```

## A.4 Cas $\Sigma_1$ et $\Sigma_2$ quelconques

```
function [d,g,p_err,nE1,nE2]=case3(l,x)

% separation des individus de chaque classe
    l1 = l(find(l(:,size(l,2))==1),:);
    l2 = l(find(l(:,size(l,2))==2),:);

% estimation des moyennes
    m1 = mean(l1(:,1:size(l1,2)-1))';
    m2 = mean(l2(:,1:size(l2,2)-1))';

% estimation des matrices de covariances
    sig1 = cov(l1(:,1:size(l1,2)-1));
    sig2 = cov(l2(:,1:size(l2,2)-1));

% parametres des fonctions discriminantes
    p1 = size(l1,1) / size(l,1);
    b1 = -.5 * log(det(sig1)) + log(p1);
    i1 = inv(sig1);

    p2 = size(l2,1) / size(l,1);
    b2 = -.5 * log(det(sig2)) + log(p2);
    i2 = inv(sig2);

    g=zeros(size(x,1),1);

% calcul de la fonction discriminate pour chaque individus
    for i=1:size(x,1)
        ind = x(i,1:size(x,2)-1)';
        g(i,1) = -.5 * (ind - m1)' * i1 * (ind - m1) + b1
```

```
            + .5 * (ind - m2)´ * i2 * (ind - m2) - b2;
    end;

% calcul du vecteur des decisions
    d = zeros(size(g));
    f1 = find(g>0);
    d(find(g>0))=ones(size(f1));
    f2 = find(g<=0);
    d(find(g<=0))=2*ones(size(f2));

% estimation de la probabilite d´erreur
    p_err = size(find(d~=x(:,size(x,2))),1) / size(x,1);

% "decide 1 a la place de 2"
    nE1 = size(find(d==1 & x(:,size(x,2))==2),1);
% "decide 2 a la place de 1"
    nE2 = size(find(d==2 & x(:,size(x,2))==1),1);
```