SY09 - TP04
Analyses discriminantes quadratique et linéaire

Bertrand Bon - Antoine Hars

June 12, 2013

Introduction

Dans le cadre de ce tp, nous avons étudié les analyses discriminantes quadratique et linéaire.

Exercice 1 : Règle de Bayes.

On suppose que la population est répartie en deux classes, en proportions π_1 et π_2 = 1 - π_1, issues des distributions gaussiennes bivariées $\mathcal{N}(\mu_1, \Sigma_1)$ et $\mathcal{N}(\mu_1, \Sigma_1)$.

1. Donner une équation de la frontière de décision de la règle de Bayes dans chacun des cas suivants :

(a) $\pi_1 = \mathbf{0.5}$, $\mu_1 = \mathbf{(0,0)}'$, $\mu_2 = \mathbf{(1,1)}'$, $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ **:**

(b) $\pi_1 = \mathbf{0.1}$, $\mu_1 = \mathbf{(0,0)}'$, $\mu_2 = \mathbf{(1,1)}'$, $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ **:**

(c) $\pi_1 = \mathbf{0.5}$, $\mu_1 = \mathbf{(0,0)}'$, $\mu_2 = \mathbf{(1,1)}'$, $\Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & -0.3 \\ -0.3 & 1 \end{pmatrix}$ **:**

(d) $\pi_1 = \mathbf{0.6}$, $\mu_1 = \mu_2 = \mathbf{(1,1)}'$, $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$ **:**

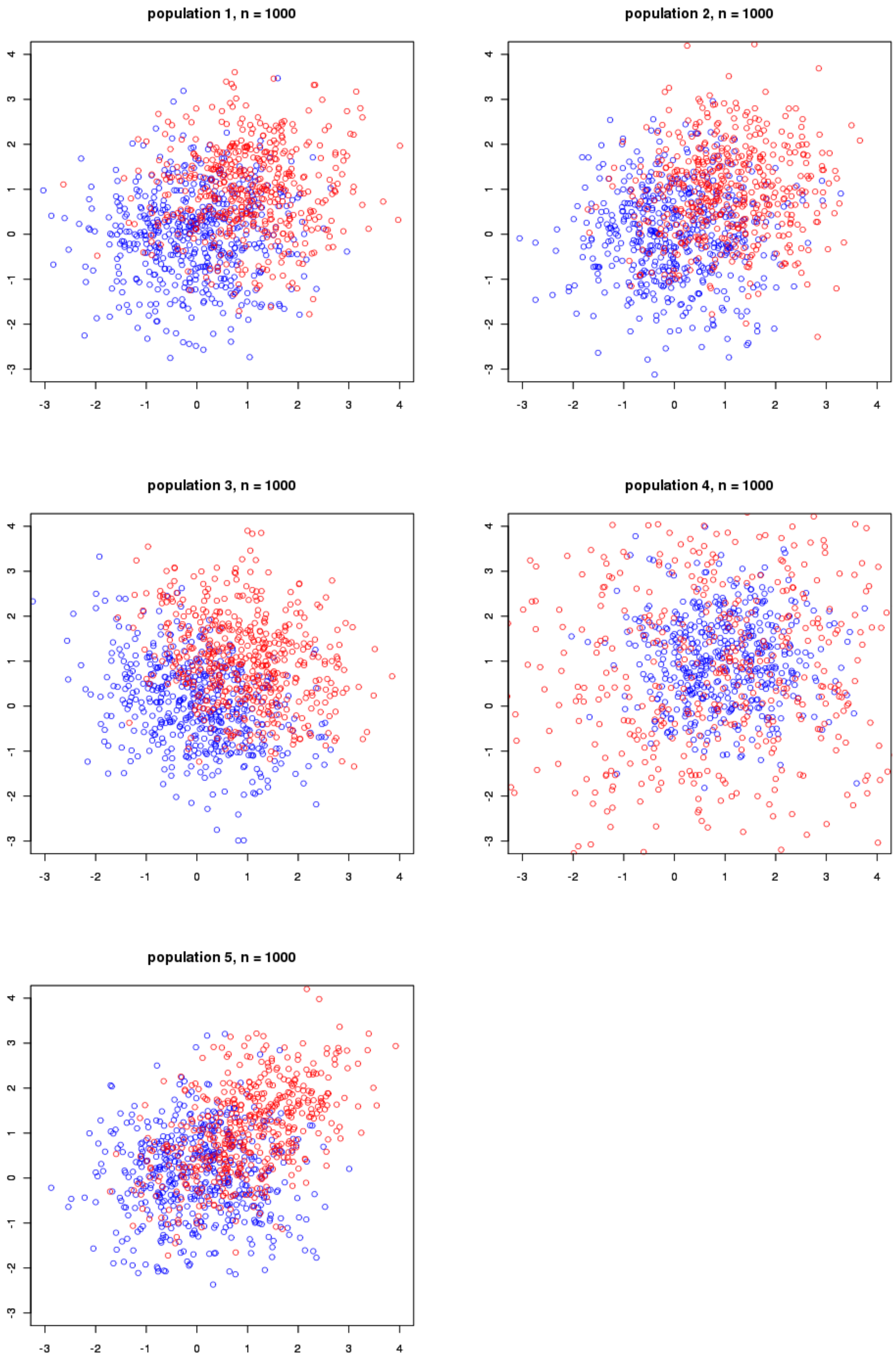
(e) $\pi_1 = \mathbf{0.6}$, $\mu_1 = \mathbf{(0,0)}'$, $\mu_2 = \mathbf{(1,1)}'$, $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$ **:**

2. Simulation de la règle de Bayes dans R :

Pour chacune des cinq populations précédentes, en utilisant la fonction *simul* réalisée au TD3 (Théorie de la décision), nous avons généré un échantillon de taille $n = 1000$.
Pour rappel, le code de la fonction *simul* est comme suit :

```
simul <- function (n, pi, mu1, mu2, sigma1, sigma2) {

  # On crée la matrice contenant l'échantillon résultat de la fonction.
  result = matrix(nrow = n, ncol = 3)

  # Affectation de chaque élément de l'échantillon final à la classe 0 ou 1.
  for (k in 1:n) {

    rand = sample(0:1, 1)
    result[k, 3] = rand

    if (result[k, 3] == 0) {
      result[k, c(1, 2)] = mvrnorm(1, mu1, sigma1)
    } else {
      result[k, c(1, 2)] = mvrnorm(1, mu2, sigma2)
```

```
    }
  }

  return (result)
}
```



Nous avons par la suite tracé le nuage associé à chaque échantillon avec le tracé de la frontière de décision pour les trois premiers cas.
Pour chaque cas de figure, nous avons déterminé l'expression d'un estimateur de la probabilité d'erreur, ainsi que sa réalisation sur l'échantillon correspondant.

Cela nous a permis de le comparer avec la probabilité d'erreur théorique.

Exercice 2 : Analyse discriminante sur les données *Crabes.*

1. Expliquer en deux lignes ce que fait chacune des fonctions :

lda : Cette fonction sert à effectuer l'analyse discriminante linéaire qui cherche à détecter sur la matrice de covariance est singulière.
Cette fonction peut être appelée avec en paramètre une formule, un data frame, une matrice.
qda : Cette fonction est utilisée pour effectuer l'analyse discriminante quadratique en utilisant une décomposition QR qui retournera un message d'erreur si la variance du groupe est singulière pour chaque groupe.
contour : Cette fonction est utile pour créer un graphe de contour ou pour ajouter une ligne de contour à un graphe existante.
sample : Cette fonction nous permet de récupérer un échantillon de taille spécifiée d'éléments de l'ensemble X.
predict : Predict() est une fonction générique de prédictions de modèle.
predict.lda : Cette fonction classifie des observations multi-variables en utilisant l'analyse discrimante linéraire.
Comparaison entre predict et predict.lda : predict est générique et predict.lda non.

Exercice 3 :

Conclusion