

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 $\pi_2 = 1 - \pi_1$, issues des distributions gaussiennes bivariées $\mathcal{N}(\mu_1, \Sigma_1)$ et $\mathcal{N}(\mu_2, \Sigma_2)$.

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

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

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

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

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

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

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) {
```

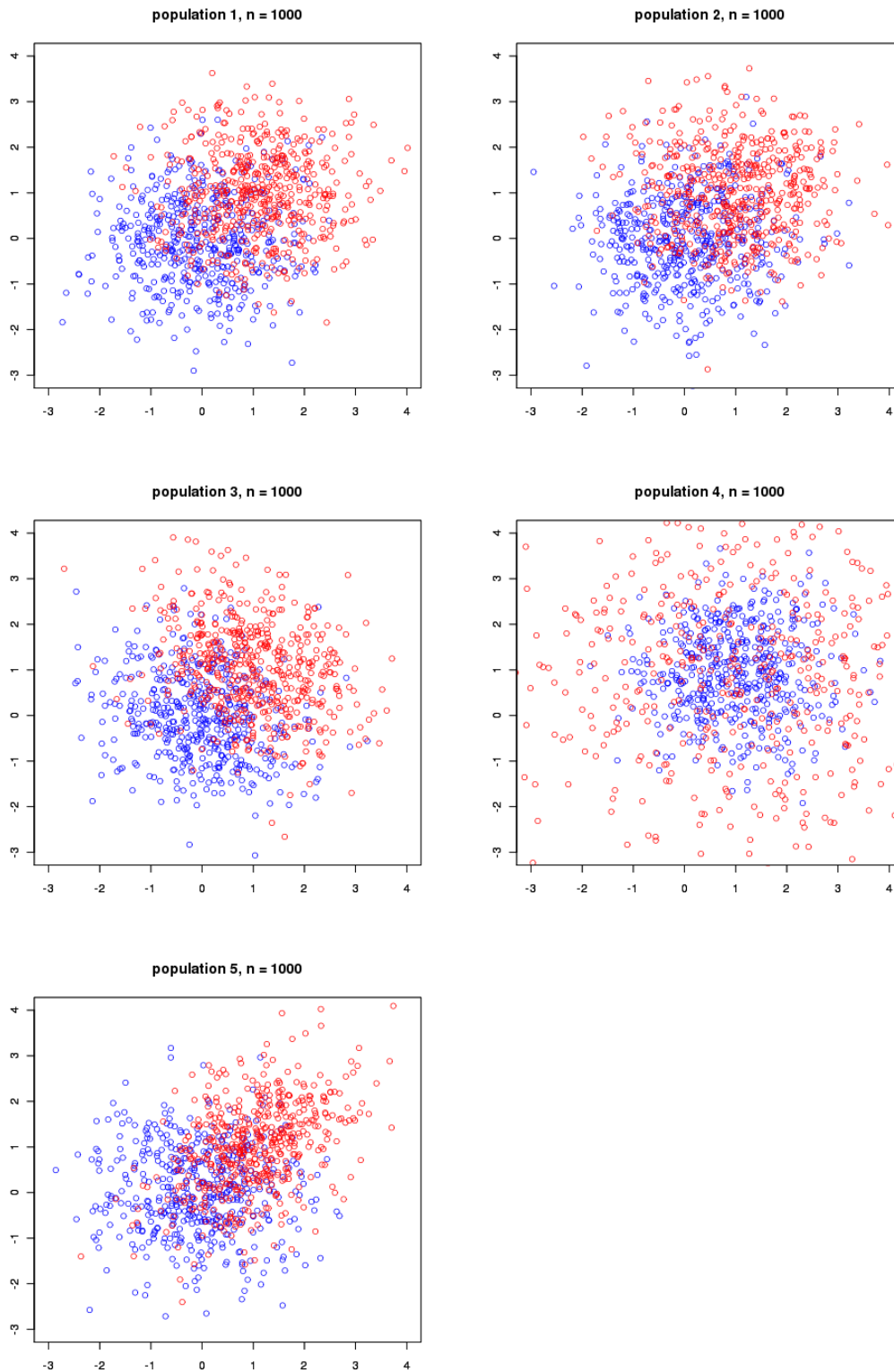
```

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

return (result)
}

```

Pour chacun des échantillons de population, nous avons tracé les nuages suivants associés, 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 :

Sa réalisation sur les échantillons nous donne les valeurs suivantes :

population	μ_1	μ_2	Probabilité d'erreur (%)
1	(-0.028, -0.093)	(1.062, 0.981)	27.6
2	(0.346, 0.006)	(1.015, 0.982)	31.1
3	(-0.063, -0.045)	(1.036, 0.915)	30.6
4	(1.018, 0.919)	(1.221, 0.965)	49
5	(-0.127, -0.018)	(1.001, 1.002)	28.2

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

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

Dans cet exercice, nous désirons utiliser l'analyse discriminante linéaire et l'analyse discriminante quadratique sur les données **crabs** afin de déterminer une fonction permettant de distinguer le sexe à partir des mesures *FL* et *RW*.

1. Expliquer ce que font les fonctions suivantes :

lda : La fonction `lda` sert à effectuer l'analyse discriminante linéaire de données (elle prend en paramètre une formule, un data frame ou une matrice).

Elle cherche à détecter si la matrice de covariance d'une classe est singulière.

qda : Cette fonction est utilisée pour exécuter une analyse discriminante quadratique sur des données en utilisant une décomposition QR qui retournera un message d'erreur si la variance du groupe est singulière pour chaque groupe.

contour : Il s'agit d'une fonction générique utile pour créer un graphe de contour ou pour ajouter des lignes de contour à un graphe existant.

sample : Cette fonction nous permet de récupérer un échantillon de taille spécifiée d'éléments de l'ensemble X en remettant en place ou non les éléments.

predict : `Predict()` est une fonction générique de prédictions à partir des résultats des diverses fonctions de création de modèles. La forme retournée dépend de la classe entrée en paramètre.

predict.lda : Cette fonction classe des observations multi-variables en utilisant l'analyse discriminante linéaire et projette les données sur les discriminantes linéaires. Cette fonction centre les discriminantes linéaires de sorte que le nombre moyen pondéré des centres de gravité du groupe soit à l'origine.

Comparaison entre predict et predict.lda : `predict.lda` est une méthode de la fonction générique `predict` pour la classe `lda`. On peut soit appeler `predict()` sur une classe `lda` d'un objet spécifié ou appeler la fonction `predict.lda()` sans se soucier de la classe de l'objet.

Exercice 3 :

Conclusion