Dimitri Charneux
Rémi Kroll

Rendu tp: Classification supervisée par Analyse Discriminante

Introduction

Durant ce TP, nous allons utiliser une analyse discriminante dans le but de classifier les données d'une image. Pour cela, nous allons utiliser les données présentent dans un fichier Rdata afin de calculer des moyennes de covariance et effectuer des analyses linéaires. Nous séparerons ainsi les données en plusieurs classes.

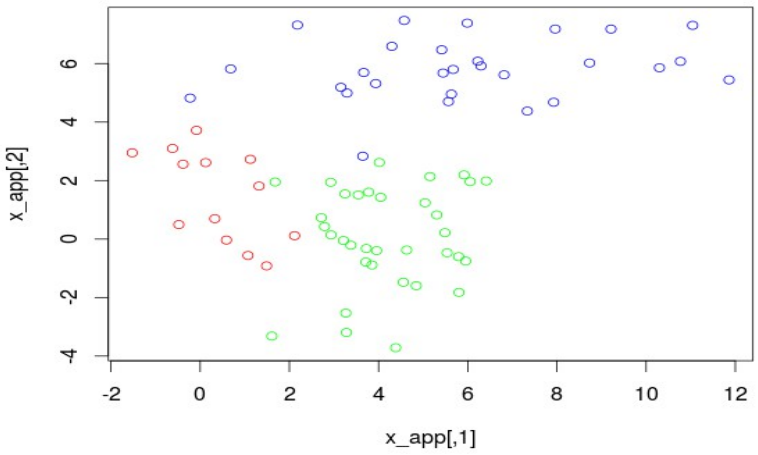
1. Classification de données gaussiennes

1.1 Affichage des observations d'apprentissage

Les données d'apprentissage sont des données qui nous permettent de diviser les points d'une image en trois ensemble. Les points des ensembles sont coloriés par trois couleurs pour représenter leur appartenance à une des trois classes. Pour faire cela, nous avons affecter respectivement la couleur rouge, bleue et vert pour la classe 1, 2 et 3 :

```
couleur[classe_app==1]='red'
couleur[classe_app==2]='blue'
couleur[classe_app==3]='green'
```

Ce qui nous donne le graphe suivant pour les données d'apprentissage :



Graphe des données d'apprentissage

La probabilité de la classe rouge p1 est de 0,2 (20%), celle de la classe bleue p2 est de 0,3 (30%) et la probabilité de la classe verte p3 est de 0,5 (50%). On en déduit que la classe 3 est plus présente dans notre ensemble, ce qui prouve que les probabilités ne sont pas équiprobables.

1.2 Estimation des moyennes et co-variance des observations d'apprentissage

Nous avons ensuite estimé une moyenne des données d'apprentissage sur les attributs de chaque classe. Cela va nous permettre de les comparer avec les valeurs des moyennes des données test. Nous allons alors compléter la marcro R par les lignes suivantes :

```
M1<-seq(1,2)
M1[1] = mean(x_app[classe_app==1,1])
M1[2] = mean(x_app[classe_app==1,2])

M2<-seq(1,2)
M2[1] = mean(x_app[classe_app==2,1])
M2[2] = mean(x_app[classe_app==2,2])

M3<-seq(1,2)
M3[1] = mean(x_app[classe_app==3,1])
M3[2] = mean(x_app[classe_app==3,2])
```

Dans un premier temps, on créé le vecteur moyenne avec la méthode seq. Puis on calcule la moyenne de la classe en faisant par exemple `mean(x_app[classe_app==1,1])` pour le premier attribut de la première classe et `mean(x_app[classe_app==1,2])` pour le deuxième attribut de la deuxième classe. Ainsi de suite, pour les autres classe des données d'apprentissage.

On obtient les résultats des moyennes ci-dessous :

```
matrice écart-type au carré théorique
classe 1 : moyenne théorique = 1  2, moyenne estimée 0.39  1.48
classe 2 : moyenne théorique = 6  6, moyenne estimée 5.98  5.82
classe 3 : moyenne théorique = 4  0, moyenne estimée 4.19  0.058
```

On observe que les données théorique sont ressemblantes aux données estimées pour les classes 2 et 3. Par contre, pour la classe 1, les valeurs ne sont pas proches. Par la suite, nous allons effectuer une analyse discriminante sur les données afin de les segmenter de manière automatique.

Calculons maintenant l'écart-type. Pour commencer, on créé la matrice de covariance de la classe puis on calcule la covariance entre l'attribut i et j allant de 1 à 2. Ci-dessous l'exemple du calcul pour la covariance de la classe 1 (même calcul pour les classes 2 et 3) :

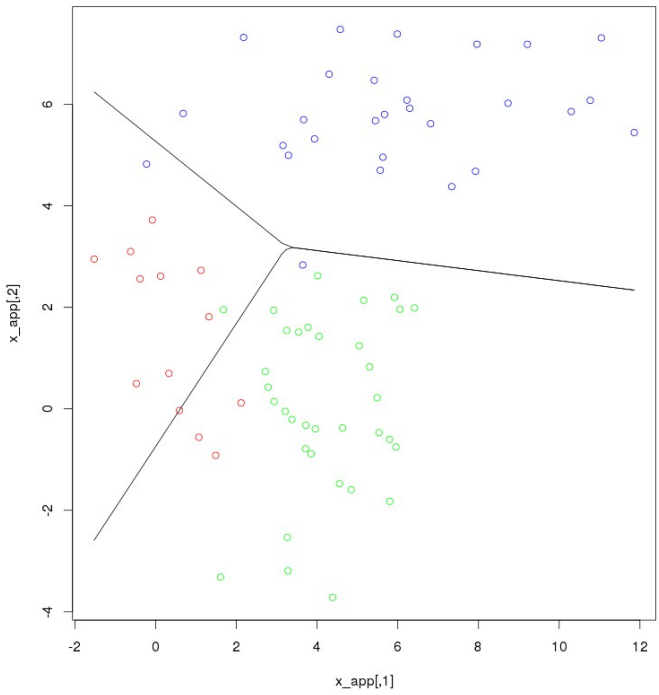
```
Sigma1<-matrix(1,2,2)
for(i in 1:2){
  for(j in 1:2){
    Sigma1[i,j]=cov(as.vector(x_app[classe_app==1,i]),
                    as.vector(x_app[classe_app==1,j]))
  }
}
```

On obtient les résultats des covariances des trois classes ci-dessous :

```
matrice covariance
S1 1.036  -0.93   S2 9.15  0.73   S3 1.59  0.29
   -0.93   2.46      0.73  1.16      0.29  2.92
```

On remarque que les valeurs théoriques et les valeurs calculées sont très différentes. Cela s'explique car la taille des échantillons est trop petite. Les attributs ne sont pas indépendants car dans la matrice, les valeurs (1,2) et (2,1) ne sont pas proches de 0.

1.3 Analyse linéaire discriminante (LDA)



Graphe des données d'apprentissage avec les lignes de décision

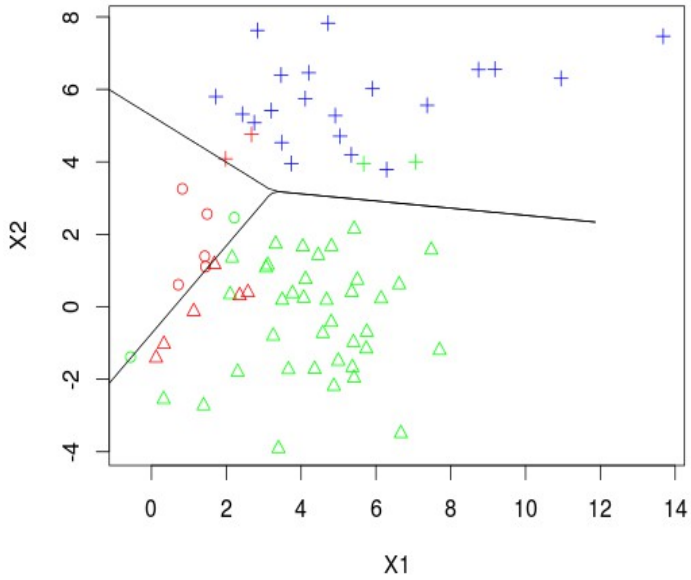
La qualité de discrimination des deux lignes de décisions nous semble plutôt bonne. En effet, comme nous pouvons le voir sur cette image, seul quelques points ne sont pas placés dans la bonne zone.

1.4 Analyse linéaire discriminante (LDA) – Affichage des observations test assignées

Avec l'analyse linéaire discriminante des données, leurs taux de bonne classification est de 84%.

Les taux de classifications des classes sont les suivants : 0.067 pour S1, 0.28 pour S2 et 0.49 pour S3. S3 est la classe dont les observations ont été le mieux classés.

On retrouve graphiquement les pourcentage obtenues précédemment comme nous le montre l'image ci-dessous :



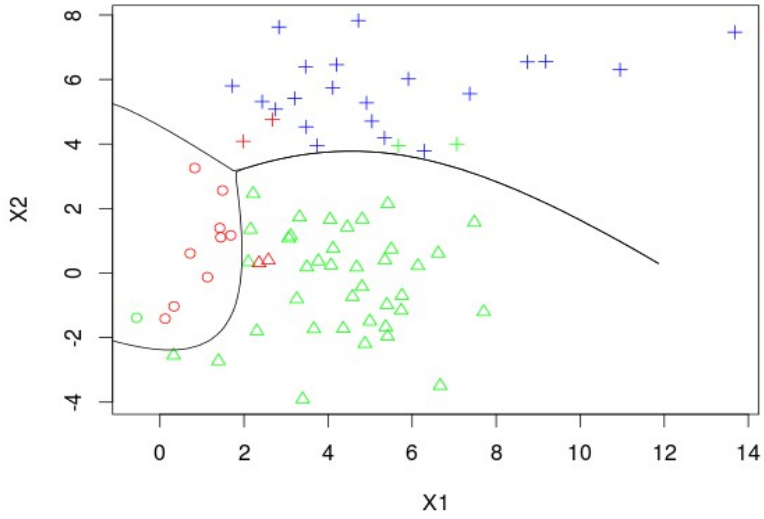
Graphe après analyse discriminante des données test

1.5 Analyse quadratique discriminante (QDA)

Nous avons appliqué l'analyse quadratique discriminante aux données de test. Avec cette analyse, le taux de bonne classification des données est de 90.7%.

Ce résultat est meilleur qu'avec l'analyse linéaire discriminante.

Nous le constatons grâce à l'image ci-dessous où on voit que les points positionnés au mauvais endroit sont peu nombreux.



Graphe des données test après analyse quadratique discriminante

Conclusion

Durant ce TP, nous avons pu effectuer des analyses discriminantes dans le but de classer les données d'une image. Nous avons également vu que l'analyse linéaire et quadratique sont des méthodes très efficaces, mais qui sont plus performantes sur certain type de donnée.