Dimitri Charneux
Rémi Kroll

Rendu tp: Réduction de la dimension par Analyse en Composantes Principales et Analyse Factorielle Discriminante

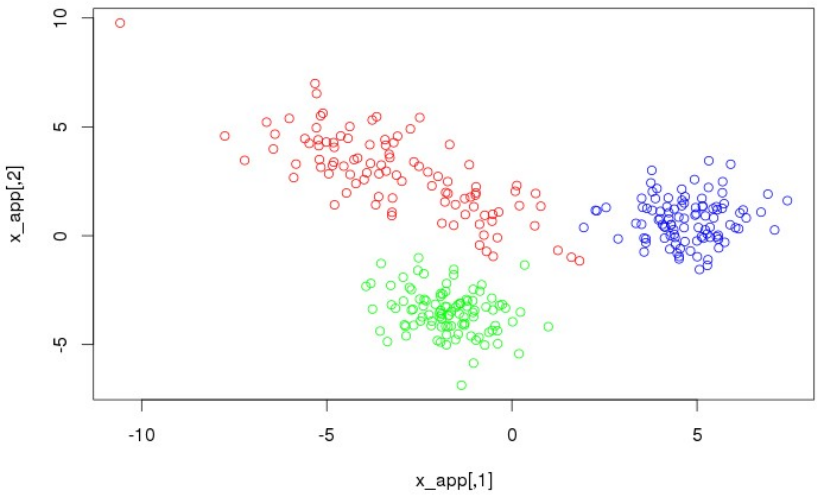
Introduction

Durant ce TP, nous cherchons à passer d'un espace de représentation bi-dimensionnel à un espace de représentation mono-dimensionnel. Pour cela, nous utiliserons deux méthodes : l'analyse en composantes principales et l'analyse factorielle discriminante. Nous manipulerons des données d'apprentissage afin de déterminer laquelle des méthodes est plus efficace dans notre cas.

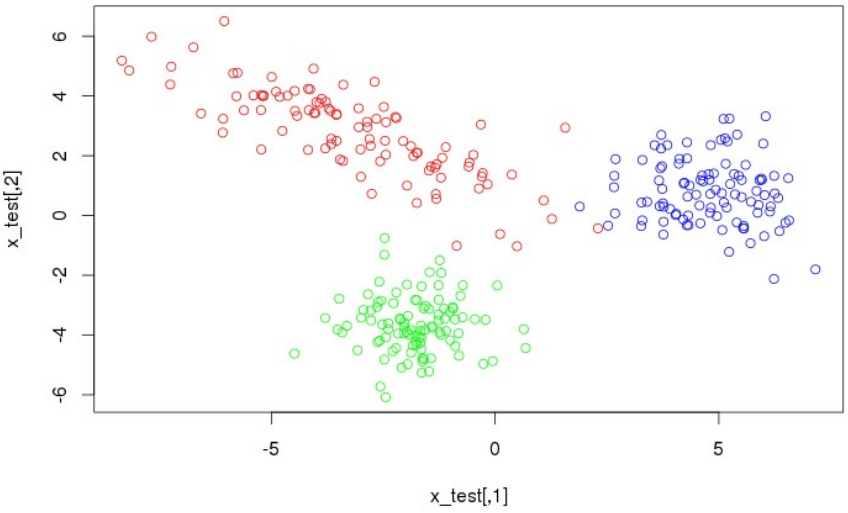
1 Chargement et Affichage des données apprentissage et test

1.1 Q1 : Affichage des données

Nous observons trois classes de données : la classe rouge est assez espacée tandis que pour les deux autres classes, leurs points sont regroupés. On en déduit que pour la classe représentée par les points de couleur rouge, nous aurons sûrement des difficulté à la séparer des deux autres.



Graphe des données d'apprentissage

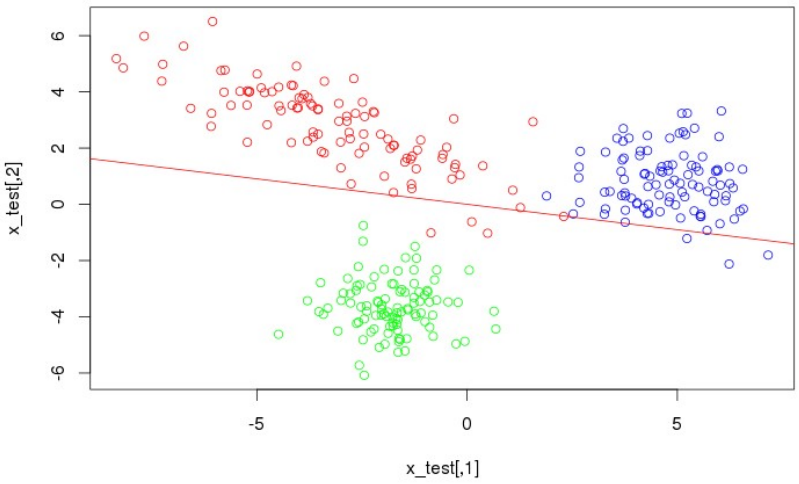


Graphe des données test

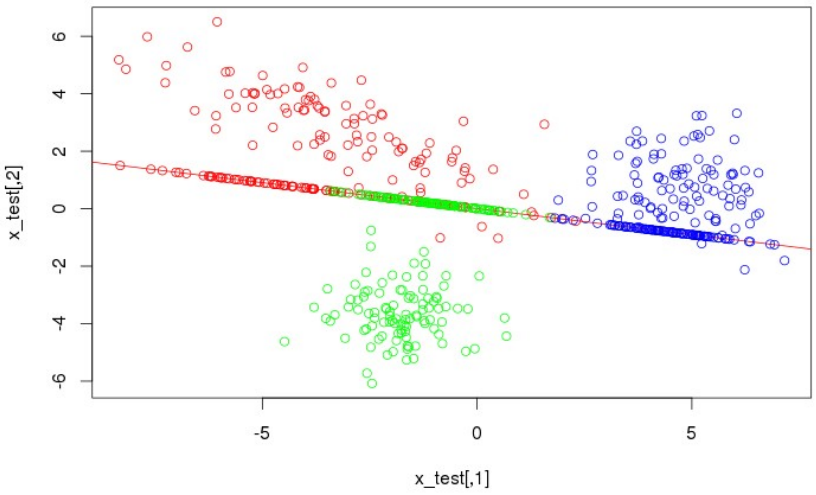
On remarque, en comparant les graphiques des données, que les données test sont similaires aux données d'apprentissage mais avec un espacement des points plus prononcé. On en déduit qu'on aura plus difficulté avec les données test pour séparer les classes.

2 Analyse en Composantes Principales (ACP)

2.1 Q2 : ACP sur données d'apprentissage



Axe discriminant des données d'apprentissage

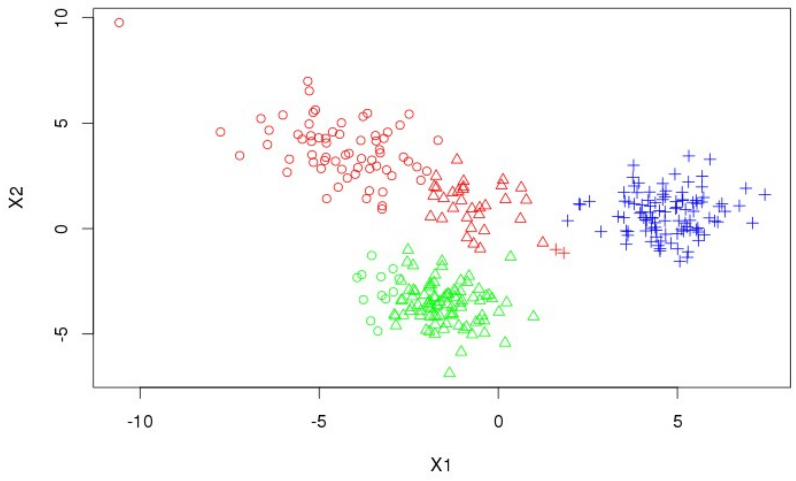


ACP des données d'apprentissage

Nous pensons que les données d'apprentissages projetées ne conviennent pas pour discriminer les trois classes car les rouges et les verts se chevauchent par contre le bleu moins (juste un point rouge dans les bleus).

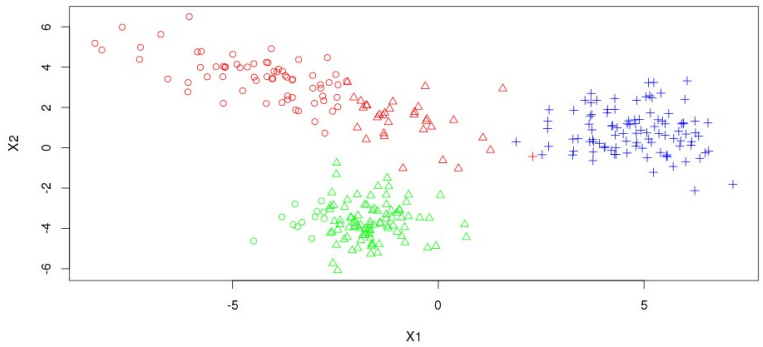
2.2 Q3 : Classification par analyse linéaire discriminante (ALD) des données d'apprentissage projetées

Le taux de bonne classification est de 84,67 %. La classe bleu est très bien classifié mais les classes vertes et rouges sont mélangées.



Classification par ALD des données d'apprentissage

2.3 Q4 : Classification par ALD des données test projetées



Classification par ALD des données test

Le taux de bonne classification est de 99 %. On remarque que les classes sont totalement séparées à 1 % près.

3 Analyse Factorielle Discriminante

3.1 Q5 : Estimation des moyennes et co-variances des classes

Pour générer les moyennes des données d'apprentissages des classes 1, 2 et 3, on effectue le calcul suivant :

```
mean1 <- colMeans(x_app[classe_app==1,])

mean2 <- colMeans(x_app[classe_app==2,])

mean3 <- colMeans(x_app[classe_app==3,])


# moyenne mean de l'ensemble des données d'apprentissage

mean <- colMeans(x_app)
```

Nous devons maintenant calculer les covariances de ces trois classes. Les moyennes et les covariances vont nous permettre de calculer la dispersion intra et inter-classe.

```
S1 <- cov(x_app[classe_app==1,])

S2 <- cov(x_app[classe_app==2,])

S3 <- cov(x_app[classe_app==3,])


Sw = S1+S2+S3

# covariance inter-classe

Sb = (mean1-mean)%*%t(mean1-mean) + (mean2-mean) %*%t(mean2-mean) + (mean3-mean)%*%t(mean3-mean)
```

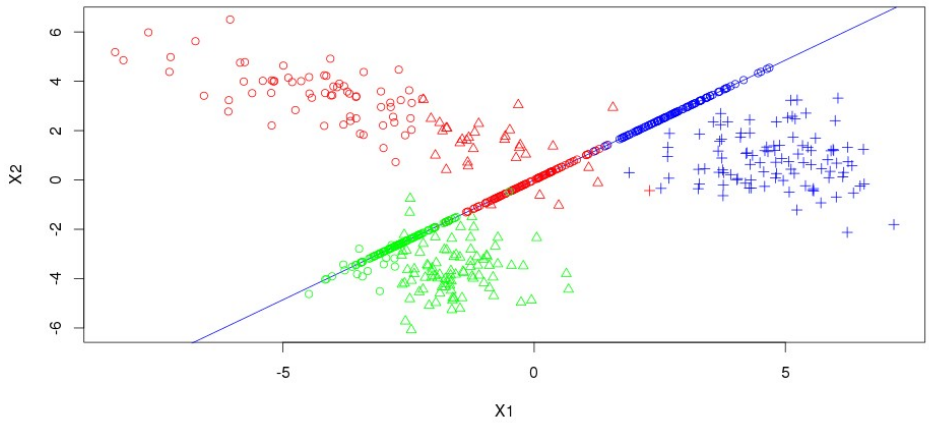
Ces données nous permettent de déterminer l'axe le plus adapté à l'analyse des données. Nous déterminons cet axe en prenant comme vecteur le vecteur propre de la valeur propre la plus élevé

3.2 Q6 : Analyse Factorielle discriminante

L'équation de la droite s'obtient de la manière suivante :

```
invSw = solve(Sw)

invSw_by_Sb = invSw %*% Sb

Vp <- eigen(invSw_by_Sb)
```

Nous obtenons le résultat suivant :



AFD des données test

Nous observons que l'axe discriminant possède une pente plus importante que dans une analyse en composantes principales. Cela permet d'avoir une meilleur séparation des classes rouges et vertes. Les couleurs des deux classes ne se chevauchent pas sur la droite. Par contre les classes rouge et bleue ne sont très bien classifié : on observe un chevauchement sur la droite.

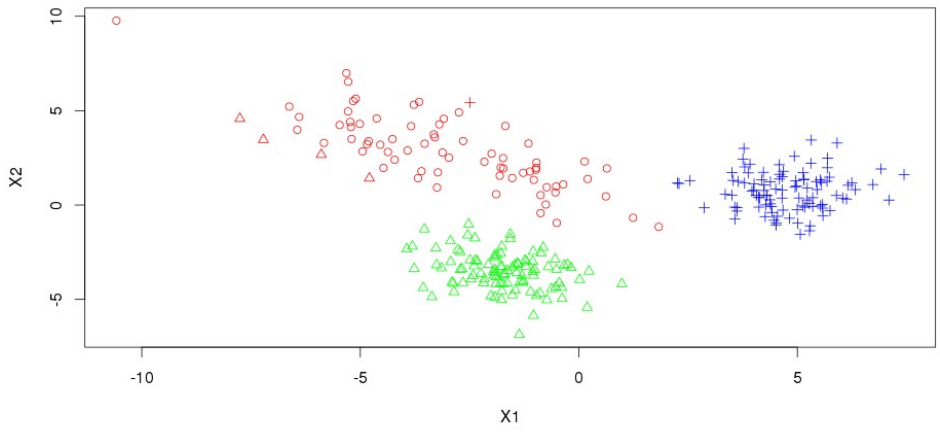
3.3 Q7 : Classification par ALD des données d'apprentissage projetées par AFD

Pour que les données d'apprentissage déterminé par AFD soient classifiées par ALD, nous nous sommes aidés du code R de l'ancien tp.

```
x_app_ACP.lda<-lda(ScalarProduct_app, classe_app)

assigne_app<-predict(x_app_ACP.lda, newdata = ScalarProduct_app)

# Estimation des taux de bonnes classifications

table_classification_app <-table(classe_app, assigne_app$class)
```

```
print("matrice de confusion :")
print(table_classification_app)

# table of correct class vs. classification
diag(prop.table(table_classification_app, 1))
# total percent correct
taux_bonne_classif_app <-sum(diag(prop.table(table_classification_app)))

# forme : les classe d'assignation fournie par l'ALD
shape<-rep(1,n_app)
shape[assigne_app$class==2]=2
shape[assigne_app$class==3]=3
# Affichage des projections apprentissage classees
plot(x_app,col=couleur2,pch=shape,xlab = "X1", ylab = "X2")
```

Nous obtenons le résultat suivant :



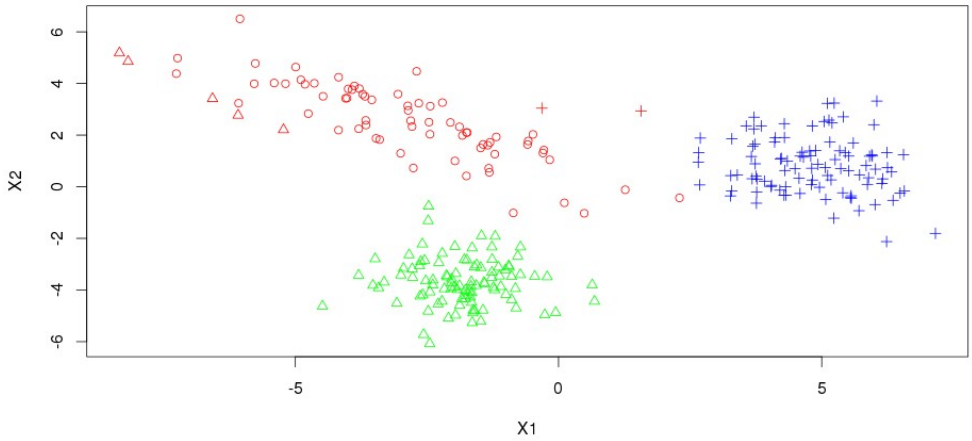
Classification par ALD des données d'apprentissage

Le taux de bonne classification des données d'apprentissage est de 97 % environ. Ce taux est meilleur que le taux obtenu par analyse en composantes principales

3.4 Q8 : Classification par ALD des données test projetées par AFD

Nous avons effectué la même analyse pour les données test. On obtient un taux de bonne

classification de 96 %.



Classification par ALD des données test

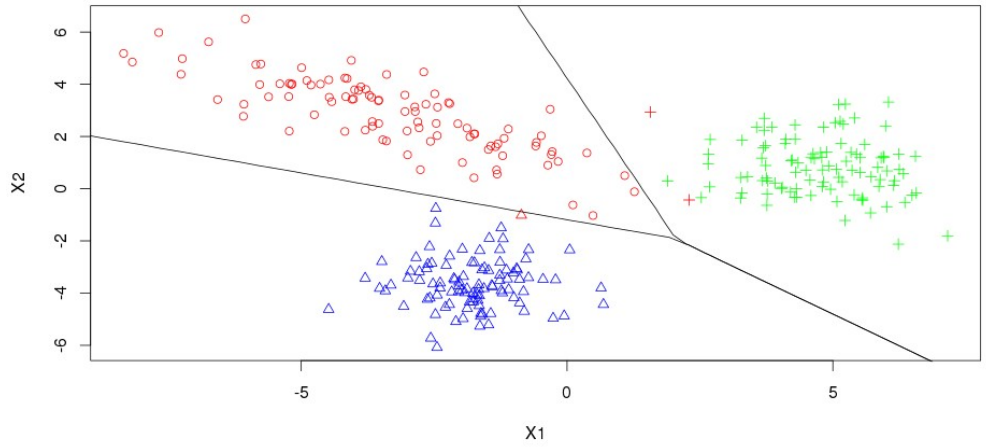
3.5 Q9 : Comparaison ACP vs AFD

Pour les données d'apprentissages ou les données test, nous obtenons toujours de meilleur taux de bonne classification avec l'analyse factorielle qu'avec l'analyse en composant principale.

Nous pouvons en conclure que l'analyse factorielle est meilleur que l'analyse en composant principale. La pente obtenu par l'analyse factorielle est plus grande que celle obtenu par l'analyse par composant principale, ce qui permet de séparer plus facilement les classes.

3.6 Q10 : Comparaison classification dans l'espace d'origine bi-dimensionnel vs sur le premier axe

Nous obtenons un taux de bonne classification dans l'espace d'origine de 99 %. L'analyse dans l'espace d'origine reste meilleur que celui sur le premier axe.



Classification dans l’espace d'origine bidimensionnel

Conclusion

Durant ce TP, nous avons pu voir qu'il n'y a pas d'analyse adapté à toutes les situations. Dans notre cas, l'analyse discriminante dans l'espace d'origine obtient un meilleur taux de bonne classification.