Master IVI S2 Rdf TP 7
Réduction de la dimension par Analyse en Composantes Principales et Analyse Factorielle Discriminante

ludovic.macaire@univ-lille1.fr

15 mars 2016

L'objectif du TP est de réduire l'espace de représentation en identifiant un axe de projection optimal. Lorsque l'apprentissage est non supervisé, nous utiliserons l'analyse en composantes principales. Lorsque l'apprentissage est supervisé (connaissance des classes), l'axe de projection sera optimal au sens du critère de Fisher (analyse factorielle discriminante). Pour les expériences, nous manipulerons des données d'apprentissage pour déterminer l'axe de projection et appliquerons nos règles de décision sur des données test.

Vous devez rendre un fichier pdf décrivant les réponses aux questions et décrivant les macros R commentées que vous avez développées.

1 Chargement et Affichage des données apprentissage et test

Dans une macro R charger les 300 données d'apprentissage du fichier 'x-app.data' et les classes associées stockées dans le fichier 'classe-app.data'. (load(file='x-app.data')).

Faire la même chose avec les 300 données test stockées dans le fichier 'x-test.data' et leurs classes associées stockées dans le fichier 'classe-test.data'.

1.1 Q1 : Affichage des données

Afficher les données d'apprentissage de telle sorte que celles associées aux classes 1, 2 et 3 soient respectivement représentées par des points rouges, verts et bleus.

Montrer cette figure et commenter les données.

Afficher les données test de telle sorte que celles associées aux classes 1, 2 et 3 soient respectivement représentées par des points rouges, verts et bleus.

Montrer cette figure et comparer la avec la figure des données d'apprentissage.

2 Analyse en Composantes Principales (ACP)

2.1 Q2 : ACP sur données d'apprentissage

Tout d'abord, il faut estimer la co-variance des données d'apprentissage par la fonction cov.

Pour déterminer l'axe le plus discriminant, il faut identifier le vecteur propre associé à la valeur propre la plus élévée par les instructions

```
# Résolution equation
Vp<- eigen(Covariance)
```

$Vp\$vectors[,1]$ contient le vecteur propre principal.

La droite dont le vecteur directeur est ce vecteur propre principal s'affiche par les instructions

```
# Affichage dela droite correspondant au vecteur propre
# dont la valeur propre la plus élevée
pente <- Vp$vectors[2,1]/Vp$vectors[1,1]
abline(a = 0, b = pente, col = "blue")
```

Calculer le vecteur des données d'apprentissage projetées sur l'axe principal. Pour ce faire, il faut stocker dans $ScalarProduct\_app$ le produit scalaire $(\% * \%)$ des données d'apprentissage x_app avec le premier vecteur propre $Vp\$vectors[,1]$, produit scalaire normalisé par la norme du premier vecteur propre :

```
sqrt(sum(Vp$vectors[,1]*Vp$vectors[,1]))
```

La projection x_app_ACP des points d'apprentissage sur l'axe le plus discriminant sera alors évaluée par

```
x_app_ACP[,1]= ScalarProduct *Vp$vectors[1,1] (le $ est bon)
```

Afficher les données d'apprentissage projetées avec des codes couleur identiques à la représentation dans l'espace d'origine et indiquer si les données d'apprentissage projetées peuvent être correctement discriminées.

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

En appliquant le code R suivant, la règle de décision de l'ALD est apprise avec les données d'apprentissage projetées et représentées par la variable $ScalarProduct\_app$.

```
x_app_ACP.lda<-lda(ScalarProduct_app,classe_app)
assigne_app<-predict(x_app_ACP.lda)
# Estimation des taux de bonnes classifications
table_classification_app <-table(classe_app, assigne_app:class)
# 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)))

# couleur de la classe 1 LABEL ORIGINAL
couleur<-rep("red",n_app) ;


# forme de la classe 1 LABEL ASSIGNATION
shape<-rep(1,n_app) ;

# Affichage des projections apprentissage classées
plot(x_app,col=couleur,pch=shape,xlab = "X1", ylab = "X2")
```

Remplacer dans le code le symbole : par le symbole $.

Compléter ce code qui affiche les points dans l'espace d'origine, de telle sorte que la couleur des points corresponde aux classes originales et la forme aux classes d'assignation fournie par l'ALD appliquée sur les données projetées.

Quel est le taux de bonne classification des données d'apprentissage projetées sur le 1er axe ?

2.3 Q4 : Classification par ALD des données test projetées

Modifier le code R de la question 3, de telle sorte que ce sont les données test projetées sur le 1er axe qui sont classifiées, et ce à partir de la règle de décision déterminée grâce à l'ALD appliquée sur les données d'apprentissage.

Quel est le taux de bonne classification des données test ?

Sont-ils satisfaisants ? Pourquoi ?

3 Analyse Factorielle Discriminante

Vous allez analyser maintenant les données d'apprentissage en tenant compte des classes.

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

Voici un script R qui estime la moyenne et la matrice de co-variance des données d'apprentissage la classe 1. Compléter le pour générer celles

des classes 2 et 3. Calculer également la moyenne *mean* de l'ensemble des données d'apprentissage.

```
# moyenne classe1
mean1 <- colMeans(x_app[classe_app==1,])

# covariance intra-classe classe 1
S1 <- cov(x_app[classe_app==1,])
#
```

Vous devez alors pouvoir calculer la dispersion intra-classe Sw et inter-classe Sb :

```
Sw=S1+S2+S3
# covariance inter-classe
Sb=(mean1-mean)%*%t(mean1-mean)+
(mean2-mean)%*%t(mean2-mean)+
(mean3-mean)%*%t(mean3-mean)
```

3.2 Q6 : Analyse Factorielle discriminante

Pour déterminer l'axe le plus discriminant pour les données d'apprentissage, il faut identifier le vecteur propre associé à la valeur propre la plus élevée par les instructions

```
# Résolution equation
invSw= solve(Sw)
invSw_by_Sb= invSw %*% Sb
Vp<- eigen(invSw_by_Sb)
```

$Vp\$vectors[,1]$ contient le vecteur propre principal.

La droite dont le vecteur directeur est ce vecteur propre principal s'affiche par les instructions

```
# Affichage de la droite correspondant au vecteur propre
# dont la valeur propre la plus élevée
pente <- Vp$vectors[2,1]/Vp$vectors[1,1]
abline(a = 0, b = pente, col = "blue")
```

Calculer le vecteur des données d'apprentissage projetées et projeter les sur l'axe principal avec des codes couleur identiques de la représentation dans l'espace d'origine.

Afficher les données d'apprentissage projetées et commenter le graphique

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

Ecrire le code R de telle sorte que les données d'apprentissage projetées sur l'axe principal déterminé par AFD soient classifiées par ALD.

Afficher les points d'apprentissage de telle sorte que la couleur des points correspond aux classes originales et la forme aux classes d'assignation par l'ALD.

Quel est le taux de bonne classification des données d'apprentissage projetées sur le 1er axe ?

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

Modifier le code R de la question 7, de telle sorte que ce sont les données test projetées sur le 1er axe qui sont classifiées à partir de la règle de décision déterminée grâce à l'ALD obtenue à partir des données d'apprentissage.

Quel est le taux de bonne classification des données test ?

3.5 Q9 : Comparaison ACP vs AFD

Comparer les taux de bonne classification des données tests obtenus par ACP et AFD. Quelle est la raison d'une telle différence ?

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

Développer une macro R telle que le classifieur ALD soit appris à partir des données d'apprentissage dans l'espace d'origine. Les données test sont alors classifiées selon cet ALD dans l'espace d'origine.

Comparer les taux de bonne classification quand les données test sont projetées par ACP ou AFD, avec les cas où les données test sont représentées dans l'espace original bi-dimensionnel.

Quelle est la solution qui permet d'avoir une règle de décision la moins gourmande en temps de calcul ?