

TIS02 Automne 2015

TP 1

Analyse discriminante

1 Données simulées

1.1 Implémentation

On se propose d'étudier et comparer trois modèles d'analyse discriminante *dans le cas binaire* (jeux de données comptant $g = 2$ classes) : l'analyse discriminante quadratique, l'analyse discriminante linéaire, et le classifieur bayésien naïf.

1. Rappeler les estimateurs des paramètres du modèle dans chacun de ces cas.
2. Compléter les fonctions `adq.app`, `adl.app`, `nba.app` et `ad.val`.

Les fonctions `adq.app`, `adl.app` et `nba.app` font l'apprentissage du modèle : elles prennent en argument d'entrée le tableau de données `Xapp` et le vecteur `zapp` des étiquettes associées, et retournent les paramètres estimés du modèle (proportions, vecteurs de moyennes et matrices de covariance des deux classes).

On pourra tirer parti de la fonction `array`, qui permet de créer des tableaux comptant plus de deux dimensions, pour stocker les matrices de covariance.

La fonction `ad.val` calcule les probabilités a posteriori pour un ensemble de données, puis effectue le classement en fonction de ces probabilités : elle prend donc en compte les paramètres du modèle et l'ensemble de données `Xtst` à classer, et retourne une structure contenant les probabilités a posteriori `prob` estimées et le classement associé.

Cette fonction s'appuie sur la fonction `mvdnorm`, disponible sur le site de l'UV, qui permet de calculer la densité d'une loi normale multivariée pour un tableau de données.

On pourra utiliser la fonction `lcur.ad`, disponible sur le site de l'UV, pour afficher les courbes de niveau des probabilités a posteriori $\mathbb{P}(\omega_1|\mathbf{x})$ estimées. La frontière de décision correspond à la courbe de niveau $\mathbb{P}(\omega_1|\mathbf{x}) = 0.5$.

1.2 Test sur données simulées

On souhaite comparer les performances de l'analyse discriminante sur un jeu de données simulées. Examiner le script `Donn.R` : que permet-il de faire ? On utilisera le script pour générer des jeux de données dont on contrôlera les caractéristiques.

Pour un jeu de données fixé, on observera le protocole expérimental suivant :

1. séparer le jeu de données en un ensemble d'apprentissage et un ensemble de test ;
2. apprendre les paramètres du modèle sur l'ensemble d'apprentissage,
3. effectuer le classement des données de test et calculer le taux d'erreur associé.

On répétera cette procédure 100 fois de manière à calculer l'erreur d'apprentissage.

2 Données réelles

2.1 Données « Pima »

On souhaite appliquer les trois modèles d'analyse discriminante et les deux modèles de régression logistique à la prédiction du diabète chez les individus d'une population d'amérindiens. On pourra charger les données au moyen du code suivant :

```
Donn <- read.csv("Pima.csv", header=T)
X <- Donn[,1:7]
z <- Donn[,8]
```

On utilisera ensuite le même protocole expérimental que pour les tests sur données simulées, en répétant l'expérience 100 fois. Calculer les taux moyens d'erreur de test pour chacun des cinq modèles étudiés. Que constate-t-on ? Comment expliquez-vous ces résultats ?

2.2 Données « breast cancer Wisconsin »

On considère à présent un problème de prédiction du niveau de gravité d'une tumeur à partir de descripteurs physiologiques. On récupérera les données sur le site de l'UV et on les chargera en utilisant le code suivant :

```
Donn <- read.csv("bcw.csv", header=T)
X <- Donn[,1:9]
z <- Donn[,10]
```

On répétera l'expérience (séparation, apprentissage et évaluation des performances) $N = 100$ fois. Calculer les taux moyens d'erreur de test. Que constate-t-on ? Interpréter et commenter.