SY09 Printemps 2019
TP 9 — Analyse discriminante

1 Analyse discriminante de données gaussiennes

1.1 Implémentation

Introduction

On souhaite comparer les performances de trois modèles d'analyse discriminante (analyse discriminante quadratique, analyse discriminante linéaire, et classifieur bayésien naïf sous hypothèse de normalité des classes), sur les jeux de données simulées `Synth1-1000`, `Synth2-1000` et `Synth3-1000`. Pour chacun de ces jeux de données, la distribution conditionnelle à chaque classe est gaussienne, les paramètres pouvant en revanche changer d'un jeu de données à l'autre.

On implémentera l'analyse discriminante via quatre fonctions : `adq.app`, `adl.app`, `nba.app` et `ad.val`. Les trois premières font l'apprentissage de chacun des modèles considérés : elles doivent donc prendre en arguments d'entrée le tableau de données `Xapp` et les étiquettes `zapp` associées, et retourner les paramètres du modèle (proportions, moyennes et matrices de covariance). On stockera de manière générique les matrices de covariance dans un tableau à trois dimensions (`array`).

La fonction `ad.val` calcule les probabilités a posteriori des classes pour un ensemble d'individus, ainsi que le classement associé : elle prend donc en compte les paramètres du modèle et l'ensemble de données `Xtst` à classer, et retourne une structure contenant les probabilités `prob` estimées et les classes prédites `pred`. On pourra s'appuyer sur la fonction `dmvnorm` [1], qui permet de calculer la densité d'une loi normale multivariée pour un tableau de données.

Vérification des fonctions

La Figure 1 montre les courbes de niveau des probabilités a posteriori $\widehat{\Pr}(\omega_k|\boldsymbol{x})$ estimées lorsque la totalité des données `Synth1-40` sont utilisées pour l'apprentissage du modèle. On pourra utiliser la fonction `prob.ad`, disponible sur le site de l'UV, pour afficher les courbes de niveau des probabilités a posteriori $\widehat{\Pr}(\omega_1|\boldsymbol{x})$ estimées par un modèle. On rappelle que la frontière de décision estimée entre les deux classes ω_1 et ω_2 correspond à la courbe de niveau $\widehat{\Pr}(\omega_1|\boldsymbol{x}) = \widehat{\Pr}(\omega_2|\boldsymbol{x})$.

Traitement des données

On utilisera le même protocole expérimental que précédemment (séparation des données en ensembles d'apprentissage et de test, apprentissage, puis classement des données et évaluation des performances), et on le répétera $N = 20$ fois pour chaque jeu de données.

1. Pour chaque jeu de données, calculer le taux d'erreur (de test) moyen sur les $N = 20$ séparations effectuées. On pourra s'appuyer sur les frontières de décision obtenues pour analyser les résultats. Comment peut-on les interpréter ?
2. Recommencer en ne sélectionnant que $n_{\text{app}} = 20$ exemples au total pour l'apprentissage ; comparer et interpréter.

1. Cette fonction, tirée de la bibliothèque `mvtnorm`, est mise à disposition sur le site de l'UV.

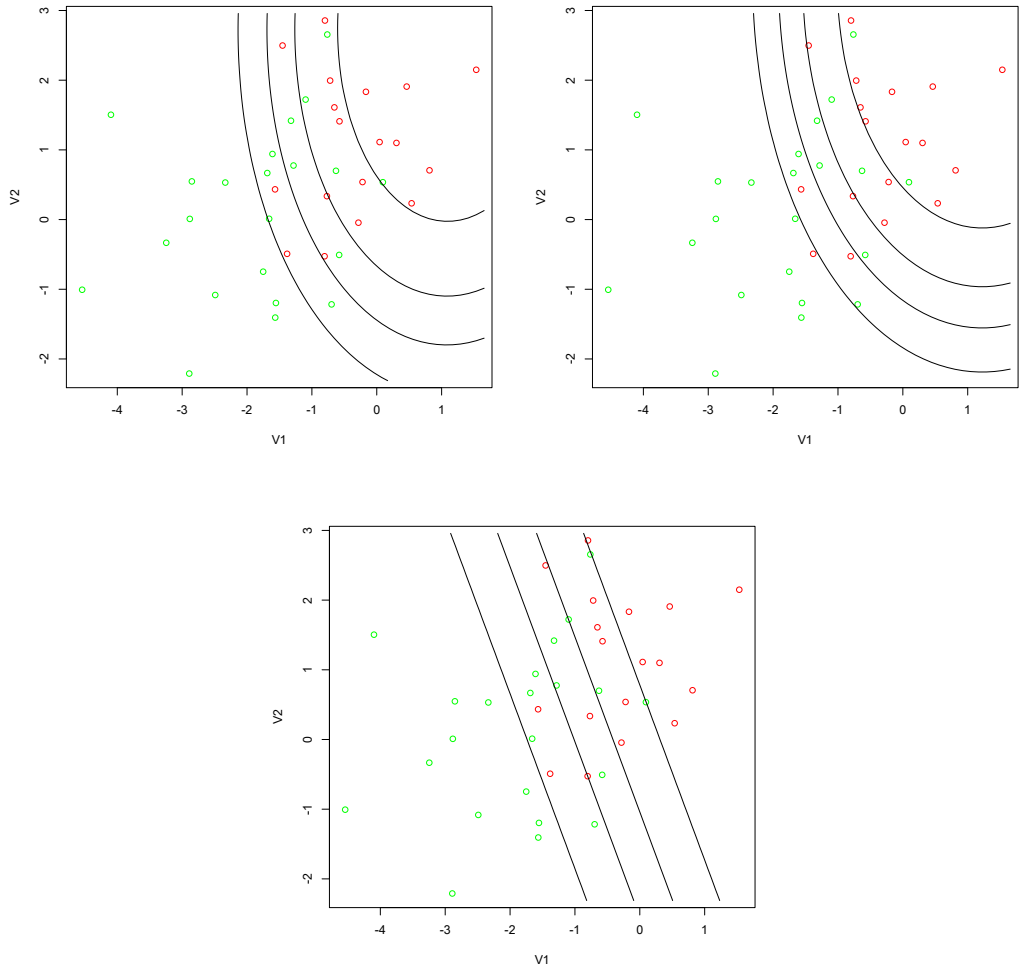

FIGURE 1 – Frontières de décision obtenues avec les données `Synth1-40` (TP6) en utilisant tous les exemples pour l'apprentissage ; haut : ADQ (gauche) et classifieur bayésien naïf (droite), bas : ADL.

1.2 Exercice théorique : règle de Bayes

Les jeux de données synthétiques utilisés au paragraphe 1.1 ont été obtenus par un processus génératif tel que décrit dans le TP précédent :

1. tout d'abord, l'effectif n_1 de la classe ω_1 a été déterminé par tirage aléatoire suivant une loi binomiale $\mathcal{B}(n, \pi_1)$, avec $\pi_1 = 0.5$;
2. n_1 individus ont ensuite été générés dans la classe ω_1 suivant une loi normale bivariée $\mathcal{N}(\mu_1, \Sigma_1)$, et $n_2 = n - n_1$ dans la classe ω_2 suivant une loi normale bivariée $\mathcal{N}(\mu_2, \Sigma_2)$.

Questions.

1. Quelles sont les distributions marginales des variables X^1 et X^2 dans chaque classe ?
2. Calculer l'expression des courbes d'iso-densité dans la classe ω_k, en fonction de μ_k et Σ_k. À quoi correspondent ces courbes dans le cas général (Σ_k quelconque) ? Si Σ_k est diagonale, sphérique ?
3. Calculer l'expression de la frontière de décision de la règle de Bayes $\delta^\star$ dans le cas général. Représenter cette frontière et comparer aux frontières obtenues avec les trois modèles d'analyse discriminante pour le jeu de données `Synth1-1000`, généré avec les paramètres suivants :

$$\mu_1 = \begin{pmatrix} -1 \\ -2 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \Sigma_1 = \begin{pmatrix} 3 & -1.5 \\ -1.5 & 2 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}.$$

4. Calculer l'expression de la frontière de décision de la règle de Bayes $\delta^\star$ lorsque Σ_1 et Σ_2 sont diagonales ($\Sigma_k = \mathrm{diag}(\sigma_{k1}^2, \sigma_{k2}^2)$, pour $k = 1, 2$).
   Représenter cette frontière et comparer aux frontières obtenues avec les trois modèles d'analyse discriminante pour le jeu de données `Synth2-1000`, généré avec les paramètres suivants :
$$\mu_1 = \begin{pmatrix} -1 \\ -2 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \Sigma_1 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 4.5 \end{pmatrix}.$$
5. Calculer la frontière de décision de la règle de Bayes $\delta^\star$ lorsque $\Sigma_1 = \Sigma_2 = \Sigma$.
   Représenter cette frontière et comparer aux frontières obtenues avec les trois modèles d'analyse discriminante pour le jeu de données `Synth3-1000`, généré avec les paramètres suivants :
$$\mu_1 = \begin{pmatrix} -1 \\ -2 \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad \Sigma_1 = \Sigma_2 = \begin{pmatrix} 3 & -1.5 \\ -1.5 & 4.5 \end{pmatrix}.$$

2 Détection de spams par analyse discriminante

On considère à présent les données contenues dans le fichier `spambase.csv`. On pourra les charger au moyen du code suivant :

```
> Spam <- read.csv("donnees/spambase.csv", header=T, row.names=1)
> X <- Spam[,-58]
> z <- as.factor(Spam[,58])
```

Les données sont des descripteurs calculés sur des courriers électroniques, et ont pour objectif de distinguer les spams des courriers réguliers. Les variables initiales sont :

1. 48 mesures de fréquences, dans le mail, de mots pré-définis,
2. 6 mesures de fréquences de caractères pré-définis,
3. la longueur moyenne des séquences ininterrompues de lettres majuscules dans le message,
4. la longueur de la plus longue séquence ininterrompue de lettres majuscules,
5. le nombre total de majuscules dans le message.

2.1 Utilisation des modèles précédents

Dans un premier temps, on utilisera « naïvement » les modèles d'analyse discriminante développés précédemment, que l'on sait présenter des garanties d'optimalité si les classes sont gaussiennes.

L'utilisation de ces modèles peut poser un certain nombre de problèmes d'ordre numérique. On pourra tenter de les résoudre en pré-traitant les données (centrage-réduction, sélection de variables [2]). Si s'avère que certains modèles ne sont pas utilisables malgré ces procédures, on veillera à apporter des éléments d'explication.

2.2 Classifieur bayésien naïf pour données binaires

Simplification des données

On considère à présent une simplification des données décrites ci-dessus ; plus particulièrement, elles ont été transformées en nouvelles variables *binaires* de la manière suivante :

1. les $48 + 6 = 54$ mesures de fréquence de mots ont été converties en indication de la présence (valeur `1`) ou absence (valeur `0`) du mot dans le message,
2. les autres mesures (longueur moyenne et maximale des séquences de lettres majuscules, nombre total de majuscules dans le message) ont été remplacées par `1` si la valeur originale était supérieure à la médiane des observations de la variable dans `spambase`, et `0` sinon.

Ces nouvelles données sont contenues dans le fichier `spambase2.csv`.

2. Dans un contexte d'apprentissage supervisé, *on exclura évidemment les données de test de la procédure de sélection de variables.*

Implémentation

Les probabilités a posteriori des classes sont obtenues par la règle de Bayes :

$$\Pr(Z_{ik} = 1|\boldsymbol{X} = \boldsymbol{x}) = \frac{\Pr(\boldsymbol{X} = \boldsymbol{x}|Z_{ik} = 1)\Pr(Z_{ik} = 1)}{\Pr(\boldsymbol{X} = \boldsymbol{x})};$$

sous l'hypothèse d'indépendance des attributs conditionnellement à la classe, on a donc

$$\Pr(Z_{ik} = 1|\boldsymbol{X} = \boldsymbol{x}) = \frac{\pi_k \prod_{j=1}^{p} (p_{kj})^{x_j}(1 - p_{kj})^{1-x_j}}{\sum_{\ell=1}^{g} \pi_\ell \prod_{j=1}^{p} (p_{\ell j})^{x_j}(1 - p_{\ell j})^{1-x_j}}.$$

On admettra pour l'instant que les estimateurs du maximum de vraisemblance des paramètres du modèle sont, pour tout $k = 1, \ldots, g$ et tout $j = 1, \ldots, p$,

$$\widehat{\pi}_k = \frac{1}{n}\sum_{i=1}^{n} z_{ik}, \quad \widehat{p}_{kj} = \frac{\sum_{i=1}^{n} z_{ik}x_{ij}}{\sum_{i=1}^{n} z_{ik}}.$$

On programmera deux fonctions pour implémenter le modèle décrit ci-dessus. On fera l'apprentissage du modèle avec une fonction `bin.app`, qui prendra comme arguments d'entrée un tableau individus-variables `Xapp` et un vecteur d'indicateurs de classe `zapp`, et retournera les estimations des paramètres du modèle (probabilités a priori `pik` et probabilités conditionnelles `pkj`).

On classera un ensemble de données de test au moyen d'une fonction `bin.val`, qui prendra en arguments d'entrée les estimations des paramètres du modèle (`pik` et `pkj`) de même que le tableau individus-variables des individus de test (`Xtst`), et retournera le vecteur `prob` des probabilités a posteriori calculées et les prédictions associées `pred`.

Test

Utiliser la procédure habituelle pour évaluer les performances du modèle sur les données `spambase2`. Comparer avec les autres modèles utilisés. Que peut-on remarquer ? Quelles performances obtient-on en comparaison de celles obtenues sur les données `spambase` ? Pourquoi ?

Formalisation (subsidiaire)

Les questions suivantes ont pour objectif de prouver l'expression des estimateurs du maximum de vraisemblance des paramètres donnés ci-dessus.

1. Quelle est la distribution conditionnelle d'un attribut X^j conditionnellement à la classe Z ? En déduire l'expression de $p_{kj} = \Pr(X^j = x_j|Z = \omega_k)$.
2. En supposant l'indépendance des variables $X^1, \ldots, X^p$ conditionnellement à la classe $Z = \omega_k$, en déduire la probabilité jointe du vecteur aléatoire $\boldsymbol{X}$ conditionnellement à la classe ω_k : $\Pr(\boldsymbol{X} = \boldsymbol{x}|Z = \omega_k)$, où $\boldsymbol{x} = (x_1, \ldots, x_p)^T$ est une réalisation du vecteur aléatoire $\boldsymbol{X}$.
3. En considérant que le i$^{\text{e}}$ exemple d'apprentissage consiste en un couple $(\boldsymbol{x}_i, \boldsymbol{z}_i)$ où $\boldsymbol{x}_i$ est une réalisation de $\boldsymbol{X}$ et $\boldsymbol{z}_i$ une réalisation de $\boldsymbol{Z}$ du vecteur de classe (avec $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{ig})^T$, et $z_{ik} = 1$ si $\boldsymbol{x}_i \in \omega_k$ et $z_{ik} = 0$ sinon), écrire la probabilité jointe $\Pr(\boldsymbol{X} = \boldsymbol{x}_i, \boldsymbol{Z} = \boldsymbol{z}_i)$.
4. En déduire la vraisemblance jointe des paramètres du modèle p_{kj} et $\pi_k = \Pr(Z = \omega_k)$ $(k = 1, \ldots, g,\ j = 1, \ldots, p)$ étant donné l'ensemble d'apprentissage $\{(\boldsymbol{x}_1, \boldsymbol{z}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{z}_n)\}$.
5. Montrer que l'EMV de chaque paramètre p_{kj} $(k = 1, \ldots, g,\ j = 1, \ldots, p)$ est

$$\widehat{p}_{kj} = \frac{1}{n_k}\sum_{i=1}^{n} z_{ik}x_{ij},$$

avec $n_k = \sum_{i=1}^{n} z_{ik}$ (on supposera pour cela que $0 < p_{kj} < 1$ pour tout $k = 1, \ldots, g$ et $j = 1, \ldots, p$), et que l'EMV de chaque probabilité a priori est

$$\widehat{\pi}_k = \frac{1}{n}\sum_{i=1}^{n} z_{ik} = \frac{n_k}{n}.$$