Analyse de données
M1 Statistique et économétrie
V. Monbet

Analyse Discriminante 2

Les données étudiées (Jobson 1992) sont issues d'une enquête réalisée auprès de 200 femmes mariées du Michigan. Les variables qualitatives sont les suivantes :

- THISYR : la variable à expliquer, (oui) si la femme travaille l'année en cours (non) sinon ;
- LASTYR : (oui) si la femme travaille l'année précédente (non) sinon ;
- CHILD1 : code la présence (oui) ou l'absence (non) d'un enfant de moins de 2 ans ;
- CHILD2 : présence (oui) ou absence (non) d'un enfant entre 2 et 6 ans ;
- ASCEND : ascendance noire (noi) ou blanche (bla).

Les autres variables, âge (AGE), nombre d'années d'études (EDUC), revenu du mari (HUBINC) sont quantitatives.

Les données sont disponibles dans le fichier
`https://www.math.univ-toulouse.fr/~besse/Wikistat/data/jobpanel.dat`.

Sous R, recoder les données

```
type=c("character","character","numeric","numeric","numeric",
     "character","character","character")
panel=read.table("jobpanel.dat",colClasses=type,header=TRUE)
summary(panel)
```

Recodage explicite des facteurs

```
# Codage explicite des facteurs
panel[,"THISYR"]=factor(panel[,1],levels=c("0","1"),labels=c("Wnon","Woui"))
panel[,"LASTYR"]=factor(panel[,2],levels=c("0","1"),labels=c("Lnon","Loui"))
panel[,"HUBINC"]=panel[,3]
panel[,"AGE"]=panel[,4]
panel[,"EDUC"]=panel[,5]
panel[,"CHILD1"]=factor(panel[,6],levels=c("0","1"),labels=c("Bnon","Boui"))
panel[,"CHILD2"]=factor(panel[,7],levels=c("0","1"),labels=c("Enon","Eoui"))
panel[,"ASCEND"]=factor(panel[,8],levels=c("0","1"),labels=c("Abla","Anoi"))
panel=panel[,-c(1:8)]
summary(panel)
```

Sous Python, on écrit par exemple

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cross_validation import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.metrics import confusion_matrix,roc_curve, auc

D = pd.read_table( \
    'https://www.math.univ-toulouse.fr/~besse/Wikistat/data/jobpanel.dat',sep='\s+')

D['THISYR'] = D['THISYR'].astype('category')
D['LASTYR'] = D['LASTYR'].astype('category')
D['CHILD1'] = D['CHILD1'].astype('category')
D['CHILD2'] = D['CHILD2'].astype('category')
D['BLACK'] = D['BLACK'].astype('category')
```

Ajuster les modèles suivants sous R puis sous Python puis les comparer en estimant l'erreur de classement et des courbes ROC. Vous pouvez sélectionner les variables les plus discriminantes pour améliorer les modèles.

- Analyse discriminante linéaire
- Analyse discriminante quadratique
- Régression logistique
- Méthode des plus proches voisins
- Arbre de décision
- Forêts aléatoires

Un exemple, sous Python,

```
X_train, X_test, y_train, y_test =  \
    train_test_split( X, y, test_size = 0.3, random_state = 100)

# lda
lda1 = LinearDiscriminantAnalysis(solver='lsqr', shrinkage=None).fit(X_train, y_train)
score_lda1 = lda1.score(X_test, y_test)
lda1_cm = confusion_matrix(y_test, y_pred)

# ROC curve
fpr, tpr, _ = roc_curve(y_test, y_pred,drop_intermediate=False)
roc_auc = auc(fpr, tpr)

plt.figure()
lw = 2
plt.plot(fpr, tpr, color='darkorange',
```

```
         lw=lw, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
```