Rapport

Projet : Analyse Discriminante Linéaire

Aleksandra Kruchinina, Aymeric Delefosse, Mamadou Diallo

Table des matières

[Architecture 3](#_Toc61088860)

[Attributs 3](#_Toc61088861)

[Méthodes 4](#_Toc61088862)

[Fit 4](#_Toc61088863)

[Predict 5](#_Toc61088864)

[Discrim\_html\_output 5](#_Toc61088865)

[Create\_pdf 5](#_Toc61088866)

[Confusion\_matrix 5](#_Toc61088867)

[Accuracy\_score 5](#_Toc61088868)

[Pseudo-code 6](#_Toc61088869)

# Architecture

Pour faire notre application on a choisi de créer une classe LDA, LinearDiscriminantAnalysis, cette solution permet de créer des instances, mieux structurer et organiser le code. En plus cela a permis d’utiliser les variables calculées dans certaines fonctions (qui n’étaient pas utilisées dans leurs retours) à partir d’autres fonctions. Par exemple, l’attribut coef\_ qui est construit dans la fonction fit() et qui est utilisé ultérieurement dans la fonction predict().

**Fonctions supplémentaires** qui permettent de faire des calculs mais qui ne sont pas dedans la classe elle-même : verification\_NA(), freq\_relat(), means\_class(), cov\_matrix(), pooled\_cov\_matrix(), wilks,), wilks\_log(), wilks\_decay(), p\_value(), variables\_exlicatives(), createWebFile().

**Classe PDF(FPDF)** est une classe supplémentaire qui facilite l'affichage du bas de page lors d’une création d’un fichier pdf. Elle hérite la classe FPDF du package fpdf. Inspiré par le tutoriel : https://pyfpdf.readthedocs.io/en/latest/Tutorial/index.html

# Attributs

dataset : le jeu de données

classEtiquette : le nom de la variable cible

classNames : les noms des valeurs prises pour la variable cible

varNames : les noms des variables explicatives

n : la taille de l'échantillon

p : le nombre de variables explicatives

K : le nombre de classes

V : les matrices de covariance totale

Vb : les matrices de covariance biaisée

W : la matrice de covariance intra-classe

Wb : la matrice de covariance intra-classe biaisée

infoDataset : l’informations de bases sur le jeu de données

infoCovMatrix : les statistiques de la matrice de covariance intra-classe

infoClasses : les effectifs et fréquences relatives des classes

infoWilksStats : les statistiques du Lambda de Wilks

intercept\_ : l’intercept calculé par le modèle

coef\_ : les coefficients du modèle

infoFuncClassement : les valeurs de la fonction de classement

confusionMatrix : la matrice de confusion

confusionMatrixGraph : la matrice de confusion, version « graphique »

accuracy : le taux de précision

# Méthodes

fit()

predict()

confusion\_matrix(self,y\_true, y\_pred)

accuracy\_score(self,y\_true, y\_pred)

create\_HTML(self) / discrim\_html\_output()

create\_pdf(self) / discrim\_pdf\_output()

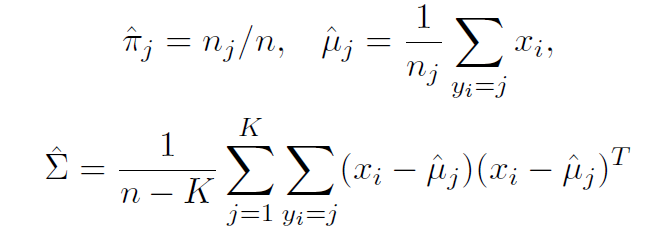
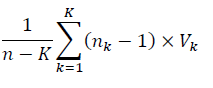
stepdisc(self, DATA, CLASS, SLENTRY, METHOD, VAR=None, BIGDATA=False, CONSOLELOG=True, HTMLFILE=None)

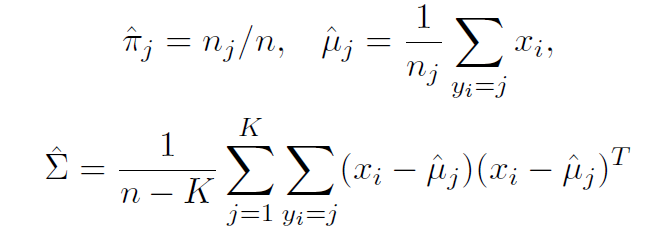
## Fit

Les valeurs nulles ne sont pas prises en compte et sont supprimées. Le message affiche le nombre d’observations supprimées.

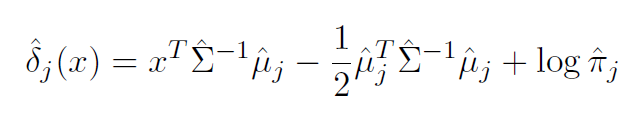
La fonction renvoie les coefficients et l’intercept du modèle.

D’habord on calcule les myennes conditionelles, les frequences relatives et la matrice de variance co-variance intra-classes[[1]](#footnote-1) :

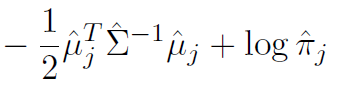




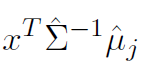
Apres on estime la fonction de classement par la formule :



Où l’intercept :

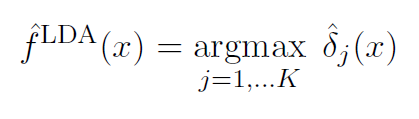


Et les coefficients :



## Predict

Pour calculer les prédictions il faut prendre le maximum parmi les scores d’appartenance à chaque classe.

Ainsi on a finalement la formule pour calculer les prédictions : 

La fonction retourne le vecteur des classes prédites.

## Discrim\_html\_output

## Create\_pdf

Cette fonction créée un fichier pdf, dont l’utilisateur donne le nom. Il contient des sorties ressemblant aux sorties de la procédure DISCRIM de SAS : la description générale de l’échantillon (Taille d'échantillon totale, le nombre des variables et des classes, les DDL), l’information au niveau classe (les fréquences et proportions des classes), l’informations sur la matrice de covariance combinée, les statistiques de lambda de Wilks, les coefficients et l’intercept construits par le modèle.

La matrice de confusion et le taux de précision ne sont pas présentes dans cette méthode car on peut les invoquer par des méthodes séparées - confusion\_matrix() et accuracy\_score().

## Confusion\_matrix

Avoir les prédictions c’est très bien, mais comment on peut savoir si ces prédictions sont correctes ou pas ? La matrice de confusion et le taux de précision y aident. La première « permet d’obtenir une évaluation non biaisée des performances du modèle en déploiement ». Elle prend le vecteur des vraies valeurs de Y qui n’étaient pas utilisées lors de l’apprentissage, et le vecteur des valeurs cibles prédites. Ainsi on peut estimer si la fonction est capable de bien prédire.

« Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée. La cellule ligne L, colonne C contient le nombre d'éléments de la classe réelle L qui ont été estimés comme appartenant à la classe C »[[2]](#footnote-2).

Cette matrice permet de mesurer plusieurs d’autres indices. Mais comme cela n’est pas l’objectif de notre projet, on a retenue que la matrice et le taux de précision.

La fonction renvoie la matrice numérique.

# TO BE CORRECTED

En analyse des données c’est toujours utile de construire des graphiques ou d’avoir les données en format plus lisible. Ce que heatmap de la librairie seaborn permet de faire pour la matrice de confusion.

## Accuracy\_score

La fonction prend en entrée les vraies valeurs de la variable cible et les prédictions et renvoie la proportion des prédictions correctes.

Accuracy = (True Positif + True Negatif) / (Positives + Negatives)

# Pseudo-code

#BASED ON LDA.V1 -> TBC

Fonction freq\_relat(y, n; taille de l'échantillon)

Fin Fonction

Fonction means\_class(X,y)

Fin Fonction

Fonction cov\_matrix(DATA)

Fin Fonction

Fonction pooled\_cov\_matrix(DATA, CLASS)

Fin Fonction

Fonction wilks(Vb, Wb)

Fin Fonction

Fonction wilks\_log(Vb, Wb)

Fin Fonction

Fonction wilks\_decay(VAR, Vb, Wb)

Fin Fonction

Fonction p\_value(F, ddl1, ddl2)

Fin Fonction

Fonction createWebFile(filename)

Fin Fonction

**class LDA()**

Fonction variables\_explicatives(x)

Fin Fonction

Fonction fit (X : matrice, y : vecteur)

Vérification et suppression des valeurs nulles de Y et X

Calcul et stockage des descriptions : taille de l’echantillon, nombre de variables, nombre de classe, DDL Total, "DDL in classes", "DDL between classes"

              Calcul des fréquences de classes (modalités de Y)

Stockage des fréquences et les proportions des classes

Calcul des moyennes des X conditionnellement à Y

Calcul de la matrice de covariance totale Vt

Calcul de la matrice de cov. totale biaisée Vb

Calcul de la matrice de var-covar intra-classes W et de la matrice de covariance intra-classes biaisée Wb, des déterminants respectifs et log(det)

Calcul de la matrice inverse de W (invW)

Calcul et stockage du rang (RangW), du déterminant (DetW) et du log(det) de la matrice (LogDetW)

Calcul des coefficients et de l’intercept

Calcul et stockage du lambda de Wilks, d’une statistique du test F Rao, de la p-value

              Renvoyer (coefficients, intercept)

Fin Fonction

Fonction predict(XP : matrice des valeurs à prédire) :

Création d’une liste vide pour la remplir avec les prédictions

Parcours la matrice XP

Calcul des probabilités d’appartenir aux classes

Stockage de ces valeurs dans une liste

Prédictions des classes en choisissant le maximum parmi les probabilités

Fin Fonction

Fonction confusion\_matrix(y\_true : vecteur des vrais valeurs de la cible,

y\_pred : vecteur des valeurs de la cible à prédire)

Conversion des noms des classes en numérique

Création d’une matrice zéro de taille K x K

Calcul des valeurs de la matrice de confusion

# conf\_mat[i,j]=somme( (y\_pred==i) ET (y\_true==j) )

Renvoyer la matrice de confusion

#TBC

Affichage plus lisible de la matrice de confusion en utilisant le package heatmap

Fin Fonction

Fonction accuracy\_score(y\_true : vecteur des vrais valeurs de la cible,

y\_pred : vecteur des valeurs de la cible à prédire):

Conversion des noms des classes en numérique

Calcul du taux de précision

# Accuracy = (TP + TN) / (P + N)

Renvoyer le taux de précision

Fin Fonction

Fonction create\_HTML()

Fin Fonction

Fonction create\_pdf()

Création d’une instance de la classe PDF

Ajout des cellules de pdf contenant les informations sur le model

Sauvegarde en demandant l’utilisateur le nom du fichier

Fin Fonction

1. https://www.stat.cmu.edu/~ryantibs/datamining/lectures/21-clas2.pdf [↑](#footnote-ref-1)
2. https://fr.wikipedia.org/wiki/Matrice\_de\_confusion [↑](#footnote-ref-2)