Régression PLS pour le classement - Création d'un package pour R

M2 SISE - Nawres Dhiflaoui - Chrystelle Grosso - Hugo Urbaniak

Décembre 2022

Contents

1 INTRODUCTION

1.1 Régression PLS : de quoi s'agit-il ?

La régression PLS (Partial Lesat Squares) est une méthode développée par Herman Wold, basée sur l'algorithme NIPALS (Non Iterative Partial Least Squares), que le chercheur a à l'origine conçu pour l'Analyse en Composantes Principales (ACP).

La régression PLS, comme l'ACP, cherche à trouver des composantes qui maximisent la variabilité, c'est-à-dire la variance des prédicteurs, mais diffère de l'ACP dans la mesure où la PLS exige que les composantes aient une corrélation maximale avec une ou plusieurs variables à expliquer. En ce sens, la PLS est une procédure supervisée (on choisit une ou plusieurs variables à expliquer) tandis que l'ACP est non supervisée[1].

Afin d'expliquer q variables Y à partir de p variables explicatives X, on réalise une régression de Y sur X en projetant les variables dans un nouvel espace. Au cours de ce processus de modélisation linéaire, la régression PLS va construire la série de facteurs (uh, th) afin que leur covariance soit maximale.
Les descripteurs sont résumés en une série de 2X2 facteurs orthogonaux th (axes factoriels, variables latentes, X-scores).
Le nombre de facteurs ne peut excéder le nombre de variables explicatives.
Les Y variables/cibles à expliquer et à prédire sont résumées en une série de composantes uh (scores Y).

A la fois méthode factorielle et méthode prédictive, la régression PLS présente l'avantage de bien interpréter les résultats, à l'inverse, par exemple, de la régression Ridge qui ne contient pas de procédure de réduction de dimensions[2]. Les observations étant projetées dans un nouvel espace, ce dernier permet d'expliciter les relations entre les variables, de mieux situer les proximités entre les individus.

Enfin, il est d'usage de parler de régression "PLS1" lorsqu'il y a une seule variable Y à prédire, et de "PLS2" s'il y en a plus de 2. Le professeur M.Tenenhaus, spécialiste français de la régression PLS, a notamment formalisé la PLS1 et la PLS2 dans son ouvrage de référence de 1998[3].

[1] H.Wold : - Estimation of principal components and related models by iterative least squares - In Multivariate Analysis (Ed., P.R. Krishnaiah), Academic Press, NY, 1966, pp. 391-420.
- Path models with latent variables: The NIPALS approach, In Quantitative Sociology: International perspectives on mathematical and statistical model building (Ed.s, H.M. Blalock et al.). Academic Press, NY, 1975, pp. 307-357.
- Nonlinear Iterative Partial Least Squares (NIPALS) Modelling: Some Current Developments. Department of statistics, university of Göteborg, Sweden, 1973, pp. 383-407.
- Model Construction and Evaluation When Theoretical Knowledge Is Scarce: Theory and Application of Partial Least Squares Evaluation of Econometric Models. Academic press, NY,1980, pp.47-74.

[2] S. Vancolen - La régression PLS – Diplôme Postgrade en Statistique- Université de Neuchâtel - 22 juin 2004. https://core.ac.uk/download/pdf/20641325.pdf.

[3] M.Tenenhaus - La régression PLS. Editions Technip, Paris, 1998.

1.2 Analyse discriminante des moindres carrés partiels (PLS-DA) : quelle différence avec la régression PLS ?

L'analyse discriminante des moindres carrés partiels (PLS-DA) est une variante de la régression PLS.
A l'origine conçue pour traiter des variables Y continues (variables-cibles / variables à prédire et à expliquer), la régression PLS a été adaptée pour réaliser l'analyse discriminante de variables Y nominales, discrètes. On parle de "PLSDA" : Partial Least Squares Discriminant Analysis, soit en français, Analyse discriminante des moindres carrés partiels[4].

1.3 Choix du dataset dans le cadre de ce projet : dans quelle mesure est-il adapté ?

La PLS-DA est particulièrement adaptée lorsque le nombre de descripteurs (variables X explicatives de Y) est élevé par rapport au nombre d'observations.
Ainsi, dans le cadre de ce projet, nous avons opté pour le fichier breast cancer : 9 variables explicatives, 1 variable qualitative à 2 modalités à expliquer et prédire, pour seulement 699 observations. Ce jeu de données est de plus assez simple *vs.* un jeu de plus grande largeur, ce qui offre une présentation plus accessible des résultats.

[4]M.Tenenhaus - La régression PLS. Editions Technip, Paris, 1998

2 NOTRE PROGRAMME PLSDA : FONCTIONS CRÉÉES, AVEC EXPLICITATION DES MÉTHODES STATISTIQUES UTILISÉES

2.1 Avant-propos : une visualisation simple grâce à notre tutoriel WIKI-GitHub

Les différentes fonctions introduites ci-après sont également présentées dans notre WIKI GitHub afin d'offrir la meilleure prise en main possible de notre programme : les fonctions et modules sont volontairement déroulés étape-par-étape afin de visualiser les étapes de traitement inhérentes à la PLS-DA. Les codes sont entièrement affichés, ainsi que les sorties console[5].

2.2 Fonction fit( ) et ses modules associés

Les fonctions centrales de notre programme PLS-DA sont les fonctions plsda.predict( ) et plsda.fit( ) par choix du nombre de composantes ou par validation croisée, avec surcharge du print( ) et du summary( )), encapsulant des fonctions / modules que nous avons rédigés et surchargés sur le print( ). Tout au long de cette partie, les fonctions créées sont facilement repérables (textes en bleu).

Présentation générale :
- La fonction d'apprentissage fit( ) observe le prototype suivant : fit(formula, data, ncomp, cv).
- L'objet en sortie est de type PLSDA en S3 dont les fonctions génériques print( ) et summary( ) ont été surchargées.
- La fonction print( ) renvoie la classification des individus.
- La fonction summary( ) renvoie la matrice de confusion[6].

Par "fonction fit( )", nous désignons, en référence à la régression PLS, le processus de construction de la série de facteurs (uh, th) afin que leur covariance soit maximale :

uh = les Y variables/cibles à expliquer et à prévoir sont résumées en une série de composantes uh (scores Y).
th = les descripteurs - c'est-à-dire les variables explicatives X - sont résumés en une série de 2X2 facteurs orthogonaux th (axes factoriels, variables latentes, X-scores).

- La régression PLS produit une **matrice des poids W** reflétant les structures de covariance entre les prédicteurs et les réponses. Les colonnes de W sont des vecteurs de poids pour les colonnes de X produisant la **matrice de facteurs de scores T** correspondante.

Le modèle obtenu à l'issue est de type linéaire :

$$Y = XB + E, \text{ où } B = WQ, \text{ avec :}$$

E = terme du bruit pour le modèle ou Erreur,
W = matrice des poids,

[5] https://github.com/HugoUrba/PLS-DA-projet-R/wiki/GithHub-Tutorial-plsda-Use-case.
[6] Définition Wikipedia : "En apprentissage automatique supervisé, la matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée" https://en.wikipedia.org/wiki/Confusionmatrix

Q = matrice des coefficients de régression sur T.

Processus de construction pas-à-pas :

(1) Dummies. Pour la construction du modèle de prédiction, la variable cible est remplacée par K variables indicatrices définies de la manière suivante :

$$\mathbf{Z_k = 1, si Y = y_k}$$
$$0, \text{ sinon}$$

Nous avons rédigé la fonction plsda.dummies( ) surchargée sur le print( ) (affichage automatique des valeurs affectées à l'individu dans la classe Y. Exemple : si l'individu 1 était classé en "begnin", il s'affiche désormais 2 colonnes avec "begnin"= 1 et "malignant"=0".)

(2) Centrage-réduction. Il existe un effet de taille lorsque les variables explicatives X sont exprimées dans des unités de mesure différentes. Pour évacuer ce biais, les données sont préalablement centrées-réduites.

Nous avons rédigé la fonction plsda.scale( ) surchargée sur le print( ). Le print.scale( ) affiche les contrôles suivants: moyennes des colonnes bien égales à zéro ? Ecarts-types des colonnes bien égaux à 1 ?

A noter que nous avons utilisé cette fonction sur le jeu de données Breast à fin d'exemplarité. En effet, dans ce cas de figure, le centrage-réduction n'est pas essentiel puisque l'ensemble des variables X sont exprimées sur la même échelle de 0 à 10.

(3) Sélection des axes/ composantes principales :

Elle représente un choix délicat dans la mesure où nous disponsons pas des outils de la statistique inférentielle[7]. Or, l'objectif de la régression PLS est de prendre les variables latentes les plus corrélées avec Y et de les projeter ; choisir le bon nombre de composantes est donc essentiel.

En pratique, pour travailler l'ajustement des données à partir du choix du **nombre de composantes**, une approche possible consiste à suivre l'évolution des coefficients de régression en fonction du nombre de composantes choisies. Plus ce dernier est important, plus les coefficients des prédicteurs les plus fortement corrélés paraissent exploser. On s'arrête alors au nombre en-delà duquel le phénomène apparaît[8].

En cela, la prédiction interne ou **validation croisée** optimise les critères de prédiction en permettant d'estimer la qualité du modèle[9].

Notre fonction fit( ) propose soit la sélection du nombre de composantes (par défaut, 2 sont prises), soit la validation croisée avec calcul du RESS(REsidual Sum of Squares) pour chaque composante.

En raison d'une problématique de programmation, le PRESS, erreur quadratique en validation croisée avec la variable Y, n'est pas fourni. Cependant, nous en parlons ici car son utilisation est recommandée pour décider des composantes à retenir, soit *via* une réduction - on regarde si elle est supérieure ou non

[7] R.Rakotomalala – Régression PLS - Sélection du nombre d'axes – 04-04-2008 - https://tutoriels-data-science.blogspot.com/p/tutoriels-en-francais.html.

[8] S. Vancolen - La régression PLS – Diplôme Postgrade en Statistique- Université de Neuchâtel - 22 juin 2004 - https://core.ac.uk/download/pdf/20641325.pdf

[9] M.Tenenhaus - La régression PLS. Editions Technip, Paris, 1998.

à un seuil choisi par l'utilisateur [10], soit *via* le calcul du Q^2. [11]

Sur le plan théorique, le PRESS fonctionne selon cette méthode : soit A, le nombre d'axes ou de composantes. Après avoir enlevé un individu i aux matrices X et Y, on calcule $\beta^{(}\mathbf{i}_{\mathbf{A}}^{)}$ qui représente la matrice des coefficients du modèle construit avec les individus restants. Puis, on calcule l'erreur de prédiction faite sur l'individu i,

$$\mathbf{E}^{(}\mathbf{i}_{\mathbf{A}}^{)} = \mathbf{Y_i} - \mathbf{X_i}\beta^{(}\mathbf{i}_{\mathbf{A}}^{)}$$

L'erreur de prédiction est définie par le PRESS, PREDiction Sum of Squares :

$$PRESS(A) = \frac{1}{n}\sum_{i=1}^{n} ||E^{(}i_A^{)}||^2$$

Concernant le RESS que nous fournissons (REsidual Sum of Squares). La différence est que l'on ne retire pas préalablement un individu mais la règle de prise de décisions est la même (privilégier les composantes avec un faible niveau de résidus).

(4)Validation externe.

Il est possible de recourir à la validation externe grâce à l'utilisation d'un échantillon "train" (échantillon d'apprentissage qui, comme son nom l'indique, permet de faire apprendre, d'entraîner le modèle) et d'un échantillon "test", qui permet de calculer les taux d'erreur, de rappel et de précision du modèle à partir de la matrice de confusion[12].

Nous avons programmé la fonction pls.split.sample( ) avec surcharge du print
(affiche automatiquement le nombre d'individus dans l'échantillon d'entraînement et dans l'échantillon de test.)

Nous avons programmé également la fonction plsda.predict( ), surchargée sur le summary( ) qui retourne un vecteur-colonne contenant les valeurs prédites pour la variable-cible, sur l'échantillon test.

2.3 Fonction predict( )

Le prototype de la fonction predict est : plsda.predict(objetPLSDA, newdata, type) :

. objet PLSDA est un objet S3 fourni par notre fonction plsda.fit( )
. newdata sont les données à traiter. Un contrôle de cohérence est réalisé : s'agit-il bien d'un dataframe ?
. type indique le type de prédiction "class", "posterior" et renvoie en fonction, respectivement, l'objet "classe prédite" ou l'objet "probabilité d'appartenance aux classes".

Nous avons rédigé la fonction "softmax" (fonction exponentielle normalisée) pour normaliser les scores

[10] R.Rakotomalala – Régression PLS-Sélection du nombre d'axes – 04-04-2008.
[11] PL. Gonzalez, M.Tenenhaus – Les méthodes PLS – Groupe HEC - CNAM – publication post an 2000 (sans plus de précisions), P.12.
[12] Définition Wikipedia : "En apprentissage automatique supervisé, la matrice de confusion est une matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée" https://en.wikipedia.org/wiki/Confusionmatrix

d'appartenances, et affecter à la classe la plus probable.

Le softmax est une généralisation de la fonction logistique qui prend en entrée le vecteur "Ypred" de K nombre réels et qui en sort un vecteur "Ysoftmax" de K nombres réels strictement positifs et de somme 1.

2.4 Fonctions graphiques

A partir de l'objet PLS-DA généré par la fonction fit ( ) :

Nous avons créé un ensemble de fonctions graphiques (plsda.graphs( )), reprises sur l'application Rshiny dont l'éboulis des valeurs-propres, la carte des individus dans les espaces factoriels, et la carte des variables.

3 SYNTHÈSE DE L'ARCHITECTURE DU PROGRAMME

3.1 Liste des fonctions et modules R

plsda.fit( )
plsda.predict( )
plsda.split.sample( )
plsda.dummies( )
plsda.scale ( )
plsda.graphs ( )

3.2 Détail des objets générés

CLASSIFICATION

MATRICE DE CONFUSION générée avec le summary.plsda ( ).

OBJET PLSDA : Les principaux attributs de cet objet sont :

1. X la matrice des variables explicatives ;
2. Y la variable à expliquer ;
3. Modalities le vecteur des modalités de Y ;
4. Xmeans les moyennes des prédicteurs X.
5. Xweights la matrice des vecteurs des poids pour les colonnes de X.
6. pls.ncomp le nombre de composantes utilisées ;
7. coef les coefficients finaux de régression ;

Les noms, le nombre de composantes, ... sont également renvoyés. Au final, la liste exhaustive des éléments générés est :
"X" = X,
"Y" = Yb,
"Xname" = Xname,
"Yname" = Yname,
"modalities" = levels(Y),
"Xmeans" = Xmeans,
"Xweights" = Xweights,
"Yweigths" = Yweights,
"Xscores" = Xscores,
"Yscores" = Yscores,
"Xloadings" = Xloadings,
"Yloadings" = Yloadings,
"coef" = coef,
"intercept" = intercept,
"ncomp" = ncomp,
"components" = compnames,
"n iter" = n iter.

OBJETS PREDICT :

1.l'objet "classe predite" (if type=='class').
2.l'objet "probabilité d'appartenance aux classes (if type=='posterior').

4 BIBLIOGRAPHIE PAR ORDRE DE CITATION

H.Wold :
-Estimation of principal components and related models by iterative least squares - In Multivariate Analysis (Ed., P.R. Krishnaiah), Academic Press, NY, 1966, pp. 391-420.
- Path models with latent variables: The NIPALS approach, In Quantitative Sociology: International perspectives on mathematical and statistical model building (Ed.s, H.M. Blalock et al.). Academic Press, NY, 1975, pp. 307-357.
- Nonlinear Iterative Partial Least Squares (NIPALS) Modelling: Some Current Developments. Department of statistics, university of Göteborg, Sweden, 1973, pp. 383-407.
-Model Construction and Evaluation When Theoretical Knowledge Is Scarce: Theory and Application of Partial Least Squares Evaluation of Econometric Models. Academic press, NY,1980 pp.47-74.

M.Tenenhaus La régression PLS. Editions Technip, Paris, 1998.

S. Vancolen La régression PLS. Diplôme Postgrade en Statistique- Université de Neuchâtel. 22 juin 2004. https://core.ac.uk/download/pdf/20641325.pd

R.Rakotomalala Régression PLS-Sélection du nombre d'axes. 04-04-2008
https://tutoriels-data-science.blogspot.com/p/tutoriels-en-francais.html 756344325800812232

https://en.wikipedia.org/wiki/Confusionmatrix

PL. Gonzalez, M.Tenenhaus Les méthodes PLS. Groupe HEC - CNAM – Publication post an 2000 (sans plus de précisions), P.12.