TP Advanced Machine Learning - M2 ISF

Université Paris Dauphine

Régression ACE

D. Jeannel

8 Janvier 2026

Problème et objectif

Le modèle de la régression linéaire possède de nombreux avantages (interprétation, tests statistiques...) mais il suppose qu'il existe une linéarité entre la variable à expliquer Y et la variable explicative X. Mathématiquement, le modèle de régression linéaire s'écrit : $Y = a_0 + a_1 * X$. L'appréciation d'une telle relation linéaire s'effectue en observant la valeur du coefficient de corrélation linéaire. Plus ce dernier est proche de 1 en valeur absolue, plus l'hypothèse de linéarité est forte entre Y et X.

En pratique, quand la linéarité n'est pas observée, des transformations sont effectuées sur la variable X comme par exemple $Log(x)$. Il existe différentes algorithmes pour essayer d'augmenter la linéarité entre Y et X.

L'objectif de l'examen est d'implémenter un des ces algorithmes et de l'appliquer sur des jeux de données. L'algorithme à développer est l'algorithme ACE (Alternating Conditional Expectations) qui a pour but d'identifier des transformations (θ, ϕ) respectivement sur la variable explicative Y et sur la variable X pour augmenter la corrélation du modèle linéaire sur les données transformées. Le modèle transformé s'écrit $\theta(Y) = b_0 + b_1 * \phi(x)$ au lieu de $Y = a_0 + a_1 * X$.

Données

Les jeux de données utilisées pour l'examen sont les fichiers suivants :

- TEST_LISSAGE.CSV : 285 couples (X, Y) ;
- PERF_CIRCLE.CSV : 100 couples (Y, X).

Description de l'Algorithme ACE

La figure 1 énonce l'algorithme Alternating Conditional Expectations (ACE).

ACE algorithm:

- Set $f_0(x) = (x - \bar{x}\mathbb{1})/\|x - \bar{x}\mathbb{1}\|_2$
- For $k = 1, 2, 3, \ldots$
  1. Let $G(y) = \mathcal{S}(f_{k-1}(x)|y)$, and center and scale, $g_k(y) = (G(y) - \overline{G(y)}\mathbb{1})/\|G(y) - \overline{G(y)}\mathbb{1}\|_2$
  2. Let $F(x) = \mathcal{S}(g_k(y)|x)$, and center and scale, $f_k(x) = (F(x) - \overline{F(x)}\mathbb{1})/\|F(x) - \overline{F(x)}\mathbb{1}\|_2$
  3. Stop if $|f_k(x)^T g_k(y) - f_{k-1}(x)^T g_{k-1}(y)|_2$ is small
- Upon convergence, define $\mathrm{mcor}(x, y) = f_k(x)^T g_k(y)$

Figure 1: *Étapes de l'algorithme ACE.*

x et y désignent des vecteurs de dimensions $(N, 1)$. Le paramètre *mcor* désigne la corrélation maximale entre les données transformées de x et y c'est à dire $\phi(x)$ et $\theta(y)$.

La variable $S(y \mid x)$ représente une estimation de l'espérance conditionnelle $E(y \mid x)$par une fonction lissage.

Question 1

Avant d'implémenter l'algorithme ACE, il est nécessaire de considérer la fonction de lissage $S(u, v)$ qui est une approximation de $E(u \mid v)$.

1. Construire la fonction de lissage S en utilisant par exemple la fonction non paramétrique KernelReg de Python (from statsmodels.nonparametric.kernel_regression import KernelReg) ou la fonction smooth.spline de R ou toute autre fonction de lissage mais pas d'interpolation. Attention le lissage est une technique qui consiste à réduire les irrégularités et singularités d'une courbe, il ne s'agit pas d'une fonction d'interpolation ;

2. Appliquer le lissage $S(y, x)$ sur le jeu de données TEST_LISSAGE.CSV. Représenter sur un même graphique les données brutes et le lissage obtenu. Évaluez la qualité du lissage obtenu ? Les objectifs du lissage (estimation de la tendance de la courbe) sont-ils obtenus ?

Question 2

Implémenter l'algorithme ACE décrit sur la figure 1 par une fonction `my_ace(x, y, tol=`$1e-6$`, maxiter = 500)` qui prend comme arguments les variables x et y et les paramètres de convergence `tol` et `maxiter`. Ces deux arguments utilisent des valeurs par défaut. La fonction devra renvoyer les valeurs suivantes : $\theta(Y) = g(y)$, $\phi(X) = f(x)$, *mcor* et le nombre d'itérations obtenues.

1. Appliquer l'algorithme sur le jeu de données PERF_CIRCLE.CSV qui contient une variable explicative X et une variable à expliquer Y. Précisez la valeur de *mcor* et le nombre d'itérations obtenus ;

2. Afficher des graphiques montrant les transformations obtenues sur X et Y : un graphe $f(X)$ versus X, un graphe $g(Y)$ versus Y, un graphe $g(Y)$ versus $f(X)$. Comparez le dernier graphe avec Y versus X. Que remarquez-vous en particulier au niveau de la linéarité ?

3. Comparer le coefficient de la corrélation linéaire issu de la régression linéaire de Y sur X (coefficient de corrélation) et celui de la régression de $g(Y)$ sur $f(X)$.

Question 3

Dans cette question, on effectue une simulation de 500 couples de points (X_i, Y_i) issus d'une balle de rayon 1. La variable X_i sera issue d'une loi uniforme sur $[-1, 1]$ et la variable Y_i sera simulée uniformément sur le support $[-\sqrt{1-x^2}, \sqrt{1-x^2}]$.

1. Afficher les couples de points simulées sur un graphique ;

2. Appliquez l'algorithme *ACE*. Quelle valeur obtenez-vous pour *mcor* (corrélation maximale) ? En Python, la simulation d'une loi uniforme se fait à l'aide de la bibliothèque numpy et de la fonction numpy.random.uniform et en R en utilisant la fonction de base runif ;

3. Réitérez un grand nombre de fois (1000) la simulation des couples (X_i, Y_i) et appliquez l'algorithme *ACE*. Tracez un histogramme des valeurs de *mcor* obtenues. Que remarquez-vous ? Que concluez-vous sur l'algorithme *ACE* ?