

TP 9 : MIP

MLA, Stéphane Canu

17 novembre 2025, ITI, INSA Rouen

Le but du TP est d'étudier l'usage de la programmation entière mixte *mixed integer programming* (MIP), dans le cadre de la régression. Vous êtes supposé avoir déjà installé CVX.

Ex. 1 — Régression linéaire : les moindres déviations absolues (*least absolute deviations*)

1. Écrire une fonction python $(\hat{\beta}, \hat{\beta}_0) = \text{Estime_beta}(X, y)$ permettant d'estimer la valeur des paramètres d'un modèle linéaire à partir d'un échantillon de n observations y et d'une matrice $x \in \mathbb{R}^{n \times p}$ en minimisant

$$\sum_{i=1}^n |X_i^\top \hat{\beta} + \hat{\beta}_0 - y_i|$$

Estimez les paramètres du modèle à partir des 13 observations suivantes.

```
import numpy as np
X = np.array([ [-1.6000, -0.8095, -0.1649, 1.1174],
[ 0.4600, -2.9443, 0.6277, -1.0891],
[ 1.3549, 1.4384, 1.0933, 0.0326],
[ 0.5519, 0.3252, 1.1093, 0.5525],
[ 0.8152, -0.7549, -0.8637, 1.1006],
[-1.4367, 1.3703, 0.0774, 1.5442],
[ 0.8371, -1.7115, -1.2141, 0.0859],
[ 0.0246, -0.1022, -1.1135, -1.4916],
[-1.5218, -0.2414, -0.0068, -0.7423],
[-0.9986, 0.3192, 1.5326, -1.0616],
[-0.2226, 0.3129, -0.7697, 2.3505],
[ 0.9647, -0.8649, 0.3714, -0.6156],
[ 0.9861, -0.0301, -0.2256, 0.7481] ])
y = np.array([ 2.1250, -0.8500, -1.0410, 1.1469, 2.7257, -0.2190, 1.8182, 1.2390,
-0.5833, -0.0358, 1.2243, 2.3874, 2.4112 ])
```

2. Donnez une estimation de l'espérance de la valeur absolue des erreurs

$$\mathbb{E}(|X^\top \hat{\beta} + \hat{\beta}_0 - Y|)$$

où $X \in \mathbb{R}^4$, $Y \in \mathbb{R}$ deux variables aléatoires et $(\hat{\beta}, \hat{\beta}_0)$ les estimations des paramètres β et β_0 .

3. a) Écrire une fonction python $y_p = \text{predlin}(x, \hat{\beta}, \hat{\beta}_0)$ permettant de prédire la valeur de la variable Y connaissant $x \in \mathbb{R}^4$ à l'aide d'un modèle linéaire $Y = X^\top \beta + \varepsilon$. Donnez une prédiction pour la valeur de la variable y à partir des 27 observations suivantes.

```
Xt = np.array([ [-1.2504, 0.1825, -0.2730, -0.7519],
[-1.0300, -1.5651, 1.5763, 1.5163],
[-0.8261, -0.0845, -0.4809, -0.0326],
[-0.1811, 1.6039, 0.3275, 1.6360],
[-1.5184, 0.0983, 0.6647, -0.4251],
[ 1.5848, 0.0414, 0.0852, 0.5894],
[ 1.7378, -0.7342, 0.8810, -0.0628],
[ 0.0866, -0.0308, 0.3232, -2.0220],
[ 0.0808, 0.2323, -0.7841, -0.9821],
[-0.4705, 0.4264, -1.8054, 0.6125],
```

```
[ 1.5751, -0.3728, 1.8586, -0.0549],
[-0.3558, -0.2365, -0.6045, -1.1187],
[-1.2945, 2.0237, 0.1034, -0.6264],
[ 1.1393, -2.2584, 0.5632, 0.2495],
[-0.2812, 2.2294, 0.1136, -0.9930],
[-0.8198, 0.3376, -0.9047, 0.9750],
[-0.2297, 1.0001, -0.4677, -0.6407],
[-1.3481, -1.6642, -0.1249, 1.8089],
[-1.2189, -0.5900, 1.4790, -1.0799],
[ 1.7279, -0.2781, -0.8608, 0.1992],
[ 1.7791, 0.4227, 0.7847, -1.5210],
[ 0.3934, -1.6702, 0.3086, -0.7236],
[-1.4815, 0.4716, -0.2339, -0.5933],
[-0.8449, -1.2128, -1.0570, 0.4013],
[-0.4143, 0.0662, -0.2841, 0.9421],
[ 1.2882, 0.6524, -0.0867, 0.3005],
[ 0.2908, 0.9248, 0.3501, 0.7596] ])
```

b) Calculez la moyenne des carrés des erreurs

$$\frac{1}{27} \sum_{i=1}^{27} |Xt_i^\top \hat{\beta} + \hat{\beta}_0 - yt_i|$$

avec

```
yt = np.array([-0.2504, -0.03 , 0.1739, 0.8189, -0.5184, 2.5848, 2.7378, 1.0866,
 1.0808, 0.5295, 2.5751, 0.6442, -0.2945, 2.1393, 0.7188, 0.1802, 0.7703,
-0.3481, -0.2189, 2.7279, 2.7791, 1.3934, -0.4815, 0.1551, 0.5857, 2.2882,
 1.2908])
```