Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

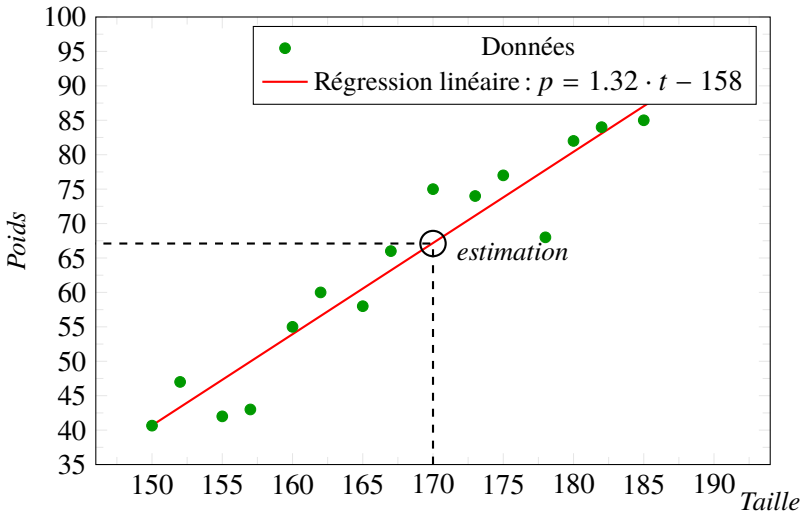
Parallélisme & Applications

Examen – mai 2021

Durée : 1h30 — Documents autorisés

Régression Linéaire

En ML, « *Machine Learning* », la **régresssion linéaire** est l'un des modèles statistiques les plus simples. Elle est utilisée pour réaliser des **estimations** ou des **prédictions** :



Sur cet exemple :

- ▷ les points représentent les couples de valeur (taille,poids) qui ont été relevés par un médecin ;
- ▷ la régression linéaire définie une droite passant « *au plus près* » de l'ensemble de ces points ;
  Ici la droite est définie par $y = 1,32 * x - 158$ *ce qui correspond à la définition d'une relation entre le poids et la taille :*
  $p = 1,32 * t - 158$

En utilisant la droite définie par régression linéaire, le poids estimé pour une taille de 170cm est de : $1,32 * 170 - 158 = 66,4kg$.

Réaliser la régression linéaire

Pour calculer les coefficients m et c de la droite de régression $y = mx + c$, on va utiliser la méthode des « *moindres carrés* » :

$$m = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sum (x-\bar{x})^2}$$

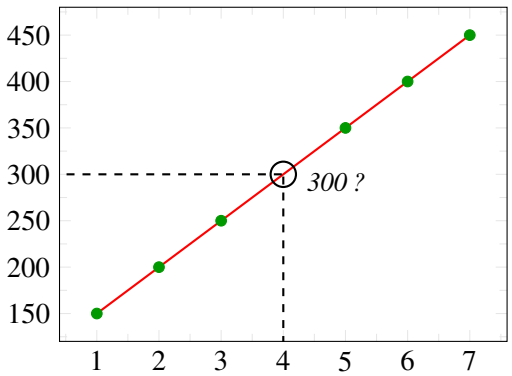
et

$$c = \bar{y} - (m * \bar{x})$$

où :

- ▷ $\bar{x}$ désigne la moyenne des valeurs x,
- ▷ $\bar{y}$ désigne la moyenne des valeurs y

Sur un **exemple simple** :

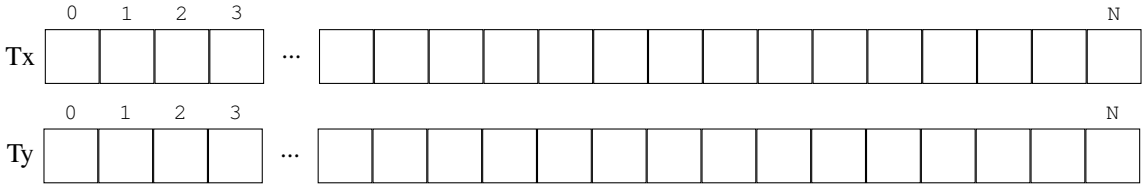


$x = (1, 2, 3, 5, 6, 7)$
$y = (150, 200, 250, 350, 400, 450)$
$\bar{x} = 4$ et $\bar{y} = 300$
$(x - \bar{x}) = (-3.0, -2.0, -1.0, 1.0, 2.0, 3.0)$
$(y-\bar{y}) = (-150.0, -100.0, -50.0, 50.0, 100.0, 150.0)$
$(x-\bar{x})(y-\bar{y}) = (450.0, 200.0, 50.0, 50.0, 200.0, 450.0)$
$\sum (x - \bar{x})(y - \bar{y}) = 1400.0$
$(x - \bar{x})^2 = (9.0, 4.0, 1.0, 1.0, 4.0, 9.0)$
$\sum (x - \bar{x})^2 = 28.0$
$m = \sum (x-\bar{x})(y-\bar{y}) / \sum (x-\bar{x})^2 = 1400/28 = 50$
$c = \bar{y} - (m * \bar{x}) = 300 - (50 * 4) = 100$

Avec $m = 50$ et $c = 100$, on prédit que pour la valeur $x = 4$, la valeur $y = m*x+c = 50*4+100 = 300$.

OpenMP — 13 points

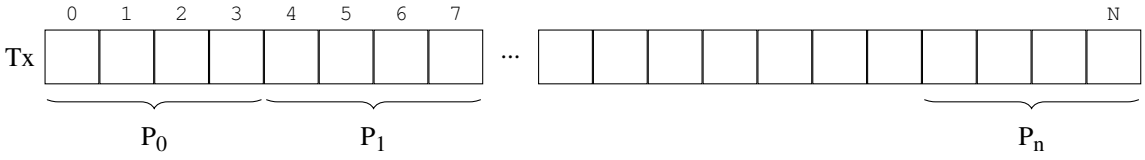
1 – 6pts Vous disposez de deux tableaux `Tx` et `Ty` de N éléments :



a. Présentez quelles opérations/méthodes d'openMP vous pouvez utiliser pour réaliser **de manière la plus efficace** la régression linéaire : calcul de m et c. *(2pts)*
Vous justifierez votre réponse.

b. Écrivez un programme utilisant openMP correspondant à votre présentation. *(4pts)*

2 – 7pts Le calcul de la moyenne $\bar{x}$ peut être fait **en parallèle** en répartissant les N données de `Tx` et `Ty` entre les différents n processeurs.



Exemple sur 6 valeurs réparties entre 3 processeurs P_0, P_1 et P_2, soient deux valeurs par processeur :
$x = (1, 2, 3, 5, 6, 7)$
$\bar{x} = \frac{\sum x}{N} = \frac{x_0+x_1+x_2+x_3+x_4+x_5}{N} = \frac{x_0+x_1}{N} + \frac{x_2+x_3}{N} + \frac{x_4+x_5}{N} = \boxed{❶}\frac{x_0+x_1}{N} + \boxed{❷}\frac{x_2+x_3}{N} + \boxed{❸}\frac{x_4+x_5}{N}$

$\Longrightarrow$*Le calcul* ❶ *peut être fait sur* P_0, ❷ *sur* P_1 *et* ❸ *sur* P_2.
P_0 peut calculer :
▷ sa partie de la moyenne $\bar{x}_{P_0}$ et $\bar{y}_{P_0}$ **ATTENTION : calculée avec N et pas avec N/n** ;
▷ $(x_{P_0} - \bar{x}_{P_0})$ et $(y_{P_0} - \bar{y}_{P_0})$ sur ses données x_{P_0} et y_{P_0} ;

Sur l'**exemple simple** précédent :

P_0		P_1		P_2	
x	y	x	y	x	y
1,2	150,200	3,5	250,350	6,7	400,450
$\bar{x}_{P_0}$	$\bar{y}_{P_0}$	$\bar{x}_{P_1}$	$\bar{y}_{P_1}$	$\bar{x}_{P_2}$	$\bar{y}_{P_2}$
0.5	58.34	1.33	100.0	2.17	141.67
$(x_{P_0} - \bar{x}_{P_0})$	$(y_{P_0} - \bar{y}_{P_0})$	$(x_{P_1} - \bar{x}_{P_1})$	$(y_{P_1} - \bar{y}_{P_1})$	$(x_{P_2} - \bar{x}_{P_2})$	$(y_{P_2} - \bar{y}_{P_2})$
(0.5, 1.5)	(91.67, 141.67)	(1.67, 3.67)	(150.0, 250.0)	(3.83, 4.83)	(258.33, 308.33)

a. Que reste-t-il à faire sur chaque processeur pour finir le calcul de $(x - \bar{x})$ et $(y - \bar{y})$? *(1pt)*

b. Comment garantir que l'on puisse calculer $(x - \bar{x})^2$ **correctement** sur chaque processeur ? *(1pt)*

c. Quelle forme de parallélisation openMP va-t-on réaliser pour réaliser ce travail si on dispose de $n = 4$ processeurs et de $N = 100000$ valeurs et en tenant compte des contraintes vues précédemment ? *(1pt)*

d. Donnez le **code openMP** correspondant à cette parallélisation et qui calcule les coefficients m et c. *(4pts)*

MPI – 7 points

3 – 7pts Chaque nœud de l'application :
▷ correspond à un processeur P_i différent ;
▷ reçoit la partie des tableaux `Tx` et `Ty` qui lui est associée ;
▷ calcule m et c de la manière présentée dans l'exercice 2).

a. Quelle **opération MPI** pourrait être utilisée pour distribuer les données de `Tx` et `Ty` entre les différents nœuds de l'application ? *(1pt)*
Comment est-il possible de tirer parti d'openMP au sein d'un nœud MPI dans le calcul de m et c ?

b. Écrivez un **programme MPI** réalisant uniquement le calcul de m et c (vous considérerez que les données associées au nœud sont déjà présentes dans le nœud). *(6pts)*
Vous pouvez utiliser openMP également.