Régression linéaire multiple

Vijayakulanathan Thanusan
Numéro étudiant : 13400963
Email : tison75@live.fr

Résumé

Cet article a pour but de présenter une évaluation des différente cœurs lors des différentes analyses de la régression linéaire multiple pour N variables.
La régression linéaire multiple est une analyse statistique dans lequel une variable quantitative Y est expliquée, modélisée, par plusieurs variables quantitatives $Xj \ (j = 1, \dots, p)$.
Le processeur utilisés pour les différentes analyses est le *Intel Core i5 2,6 GHz.*
Le programme séquentiel permet de faire une analyse sur N variables.
Cependant, aucune version parallèle n'est disponible.

INTRODUCTION

Plusieurs méthodes permettent de faire une analyse de régression linéaire multiple. La méthode que j'ai choisi est celle avec des matrices.
L'étude ici, est de mettre en évidence la liaison qui existe entre une variable qui sera une variable expliquée Y et les variables permettant d'expliquer x1,x2,…,xn. Ceci est possible par le calcul des moindres carrés . Nous ferons ces calculs en utilisant les matrices.

La première section présente comment fonctionne une analyse de régression linéaire multiple en utilisant les matrices, puis la deuxième section présente mon idée de parallèlisation en utilisant la bibliothèque OPEN MPI. La section suivante détaille et commente les résultats et enfin la dernière section présente une conclusion.

ANALYSE D'UNE REGRESSION LINEAIRE MULTIPLE EN UTILISANT LES MATRICES

La formule permettant de faire une analyse d'une régression linéaire multiple est l'équation matricielle =

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$$

L'analyse d'une régression linéaire multiple en utilisant les matrices est défini comme suit :
Pour n qui est le nombre de donnée d'une variable.
Pour k le nombre de variables.
Y est une matrice de taille(n,1) telle que =

$$\mathbf{Y}_{nx1} = [Y_1, Y_2, \dots, Y_n\ ]^T$$

X est une matrice des prédicteurs de taille(k,n) telle que =

$$[1\ X_{11}\ X_{12}\ \dots X_{1k}\ ]$$

$$\mathbf{X}_{nx(k+1)} = \ . \ . \ .$$

$$[\ 1\ X_{21}\ X_{22}\ .\ .\ .\ X_{2k}\ ]$$

$$[\ 1\ X_{n1}\ X_{n2}\ .\ .\ .\ X_{nk}\ ]$$

$\mathbf{X}^T$ est une matrice de taille (n,k) qui est la transpose de notre matrice X

$(\mathbf{X}^T\mathbf{X})^{-1}$ est une matrice de taille (k,k) qui est l'inverse de la matrice résultant de la multiplication de la transposé de X par la matrice X.

$(\mathbf{X}^T\mathbf{Y})$ est une matrice de taille (n,1) qui est le résultat de la multiplication matricielle de la matrice transposé de X par la matrice Y.

Le résultat de la formule $\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$ est en effet une matrice de taille (k,1).

la fonction multipliermatrice permet de multiplier deux matrices.
La fonction determinant permet de trouver le déterminant d'une matrice.
La fonction co-facteur permet de trouver les différents mineurs d'une matrice.

ALGORITHME 1 *co-facteur*(matrice, taille de la matrice)

1. Créer une matrice m1(taille de la matrice-1, taille de la matrice -1)
2. Créer une matrice résultat(tailledelamatrice, tailledelamatrice)
3. For(i = 0 ; i < taille de la matrice ; i++)
4. For(j = 0 ; j < taille de la matrice ; j++)
5. a)Placer dans la matrice m1 dans l'ordre respectif toutes les
6. valeurs de la matrice en excluant les valeurs qui sont dans la
7. même ligne et qui sont dans la même colonne.
8. B)appeler la fonction *determinant* qui permet de déterminer le
9. déterminant de cette matrice m1.
10. C)placer ce déterminant dans la case résultat[i][j]
11.
12. Envoyer la matrice résultat.

MON IDEE DE PARALLELISATION

Voici mon pseudo-code qui permet d'indiquer mon idée de parallélisation

1. Algorithme version parallele
2. Dans le main
3. Si (rank = maitre)
4. A) M1 est une matrice de taille (k,k) qui est en effet le
5. résultat de la multiplication du transposé d'une matrice X
6. avec la matrice X.
7. B) envoyer cette matrice M1 à tous les autres processus en
8. utilisant MPI_Bcast
9. C)créer un tableau tab dans le quel est indiqué le processus qui
10. doit travailler sur une ligne qui correspond au rang du tableau
11. D)envoyer ce tableau tab à tous les processus
12. E) for(j = 0 ; j < k ; j++)
13. trouver les déterminant de chaque colonne sur les lignes sur lequel le processus 0 doit travailler avec la fonction déterminant après avoir créer une matrice de taille (k-1,k-1) qui contient dans l'ordre respectif toutes les
valeurs de la matrice en excluant les valeurs qui sont dans la
même ligne et qui sont dans la même colonne (j)

F) Et placer ces résultats dans une matrice de taille(k,k) à la ligne et colonne respectif

12. G) Recevoir les tableaux obtenus par les autres matrices

13. et les placer à la ligne indiqué dans MPI_TAG.

15.(si rank != maitre)

16. a) recevoir le tableau tab

14. 17. b) trouver les déterminant de chaque colonne sur les lignes sur lequel le processus 0 doit travailler avec la fonction déterminant après avoir créer une matrice de taille (k-1,k-1) qui contient dans l'ordre respectif toutes les valeurs de la matrice en excluant les valeurs qui sont dans la même ligne et qui sont dans la même colonne (j)

18.c) et envoyer ces résultat au processus 0 ;