Projet tutoré de CN

Rapport d'analyse

Binôme : Yann CAUCHEPIN & Baptiste BOISSON

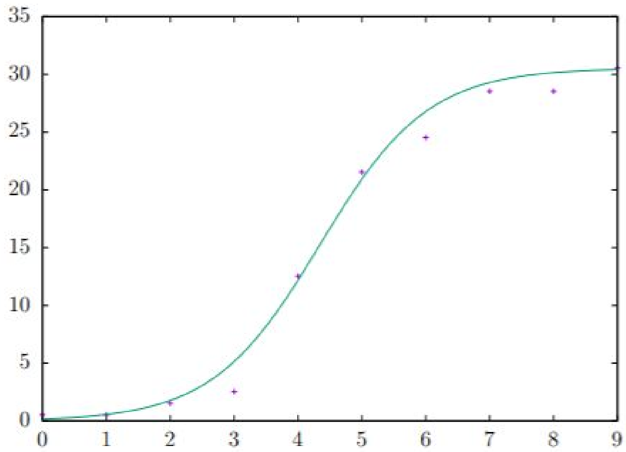
Tuteur : François BOULIER

GIS3 – 2018-2019

Sommaire

Introduction

Ce projet de calcul numérique vise à estimer la valeur de trois paramètres κ,α et ρ à partir d'un nuage de n points (x_i,y_i) par la méthode des moindres carrés. Nous résoudrons ce problème par le biais de deux méthodes différentes : Une première en résolvant le système des équations normales avec κ connu, puis une deuxième en passant par une variante de la méthode de Gauss Newton. Le langage de programmation utilisé sera le Fortran.



Courbe logistique obtenue pour κ, α, ρ = 30.54, 5.163, 1.188

nb. jours x_i	0	1	2	3	4	5	6	7	8	9
qté eau y_i	.53	.53	1.53	2.53	12.53	21.53	24.53	28.53	28.53	30.53

Coordonnées (x_i,y_i)

Avant de programmer

À partir du modèle statistique $Y_i = \kappa x_i^2 + \alpha x_i + \rho + E_i$, on cherche à déterminer la valeur du vecteur de paramètres (κ,α,ρ) pour laquelle la vraisemblance est maximale.

Le problème revient donc à trouver les valeurs de κ,α et ρ qui minimisent la somme de carrés :

$$\sum_{i=1}^{n} \left(y_i - \left(\kappa x_i^2 + \rho x_i + \alpha\right)\right)^2 = S$$

Obtenir le minimum de la somme de carrés revient à calculer le maximum de vraisemblance. La méthode pour trouver ce maximum consiste à résoudre le système d'équation des dérivées partielles formé par les paramètres κ,α et ρ :

$$\frac{\partial S}{\partial \kappa} = 0 \,, \frac{\partial S}{\partial \rho} = 0 \text{ et } \frac{\partial S}{\partial \alpha} = 0$$

En d'autres termes, résoudre le système des dérivées partielles par rapport aux paramètres revient à résoudre le système suivant :

$$\sum_{i=1}^{n} \left(y_i - \left(\kappa x_i^2 + \rho x_i + \alpha\right)\right) x_i^l = 0\%\%\% \, (1 \square l \square 3)$$

Pour faciliter les calculs, il est plus simple de passer en écriture matricielle. Résoudre le système précédent revient à résoudre le système $A^TAx = A^Tb$ avec :

$b = (y_0,y_1,y_2,...,y_n)^T$

$x = (\kappa,\alpha,\rho)^T$

$A = \{\{x_1^2, x_1, 1\}\{x_2^2, x_2, 1\}...\{x_n^2, x_n, 1\}\}$ $\{\}$ = une ligne de A

On sait que A^TA est une matrice carrée symétrique définie positive.

Cette matrice étant symétrique définie positive, elle respecte les conditions d'application de la méthode de Cholesky. Il devient alors possible de décomposer la matrice A^TA tel que $A^TA=LL^T$

Par la suite, afin de résoudre le système $LL^Tx = A^Tb$, il ne reste plus qu'à résoudre deux systèmes triangulaires en effectuant une remontée et une descente :

$Ly = A^Tb$ et $L^Tx = y$

Estimation par moindres carrés linéaires

Soit la fonction logistique : $y(x) = \kappa / (1 + e^{\alpha-\rho x})$

Dans cette partie, on possède un nuage de n points aux coordonnées (x_i,y_i). On veut déterminer les valeurs des paramètres qui font passer la sigmoïde au plus près des données. On considère ensuite κ connu et on estime par la méthode des moindres carrés les paramètres α et ρ.

On peut connaître κ visuellement en estimant l'asymptote horizontale vers laquelle tend la sinusoïde.

Cela nous permet de déterminer le vecteur Y avec la fonction logit(). On obtient alors l'équation :
$Y = \text{logit}(y/\kappa) = \rho x - \alpha$

On veut donc trouver α et ρ tel que $Y = \rho x - \alpha$ et $\kappa/y = 1 + e^{\alpha-\rho x}$

Cela revient à résoudre le système $A^TAx = A^Tb$ avec :

$b = (y_0,y_1,y_2,...,y_n)^T$
$x = (\rho,\alpha)^T$

$A = (\{x_1,-1\}\{x_2,-1\}...\{x_n,-1\})$ $\{\}$ = une ligne de A

Programmation

L'estimation par moindres carrées linéaires a été réalisé en Fortran 77. Le programme se distribue en différents fichiers dont le principale est *projet.f*. Les autres fichiers nous a été transmis par notre tuteur. Nous ne présenterons donc que celui que nous avons créé.

Le code du fichier *projet.f* se répartie en une fonction principale ainsi qu'en plusieurs fonctions annexes. Parmi les fonctions annexes, on distingue :

- TRANSP : fonction qui permet de faire la transposé d'une matrice.
- FLOGIT : fonction qui permet d'obtenir le vecteur Y à partir des résultats d'une fonction logistique.
- MULTI_MATT_VEC : fonction qui permet de faire le produit matriciel de la transposé d'une matrice par un vecteur.
- DESCENTE : fonction qui permet d'effectuer la méthode de la descente.
- REMONTE : fonction qui permet d'effectuer la méthode de la remontée.
- CHOLESKY : fonction qui permet d'effectuer la méthode de Cholesky.

Pour résoudre le système $A^TAx = A^Tb$, nous devons tout d'abord enregistrer les matrices A et b. Nous lisons donc les valeurs de x_i et y_i afin de créer les matrices A et B tel que précisé dans l'estimation par moindres carrées linéaires. Toutefois, il est à noter que la valeur de κ est donnée sous forme de variable locale car nous avons rencontré des difficultées pour la lecture de celle-ci lors de la redirection en entrée d'un fichier texte.

La résolution du programme commence par chercher tout d'abord une matrice L telle que $LL^T=A^TA$ grâce à la fonction de Cholesky. Ensuite, on pose :

1) $Ly = w$ où $w = A^Tb$ et $L^Tx = y$
2) $L^Tx = y$

La première équation se résout par une descente simple. On obtient donc y.
La deuxième équation se résout par une remontée simple. On obtient donc x.

Ainsi, on trouve l'estimation de ρ et α avec $x = (\rho,\alpha)^T$

Test

Au cours de l'élaboration du programme, nous avons effectué de nombreux tests. La première partie de notre programmation a consisté à faire fonctionner les fonctions CHOLESKY, DESCENTE et REMONTE pour une matrice A donnée. Nous avons donc fait le test avec les fichiers *exemple1.dat* et *exemple2.dat* fourni par notre tuteur.
Puis, à la suite de ces tests, nous nous sommes davantage rapprochés du problème en utilisant des matrices A et b qui dépendent des valeurs de x_i et y_i. Nous avons donc créer le fichier test *exemple.dat* où les vecteurs X et Y nous sont données à l'identique de l'énoncé. Nous savons donc que ce fichier test doit fournir :
$\kappa = 30{,}54$, $\alpha = 5{,}163$, $\rho = 1{,}188$
On indiquera donc en variable locale $\kappa = 30{,}54$.
En appliquant le programme en redirigeant en entrée le fichier *exemple.dat*, on obtient
$\alpha = 5.163224155$, $\rho = 1.187931589$
Bien sûr, le degré de précision ici est inadéquat mais en prenant quatre chiffres significatifs, on retrouve bien les mêmes valeurs que celles de l'énoncé.

Problèmes rencontrés

Nous avons fait preuve d'un manque d'efficacité dans la programmation, principalement en raison de la découverte du langage Fortran 77. Nous avons également perdu un temps important sur quelques fonctions des bibliothèques BLAS et LAPACK telle que pour la fonction DGEMM. C'est pour ces raisons que nous avons créé nous même nos propres fonctions annexes.

Estimation par moindres carrés non linéaires

Dans cette partie, κ n'est pas connu. C'est pour cela que l'estimation par moindres carrés n'est plus linéaire.

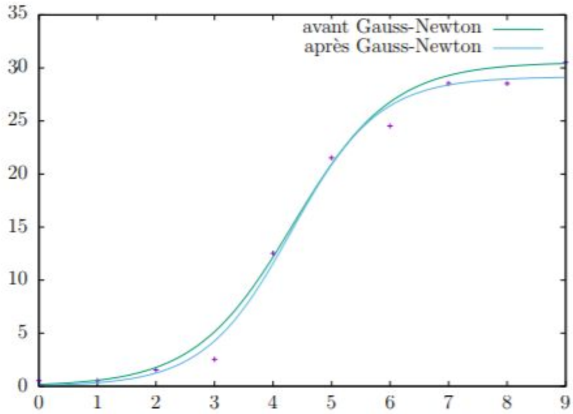
Le principe de la méthode itérative de Gauss-Newton est de construire une suite de vecteurs de taille nombre de nos paramètres qui converge vers le vecteur $v = (\kappa,\alpha,\rho)^T$.

Cette méthode consiste à établir le vecteur ***r*** et la matrice jacobienne ***J*** de ce vecteur, puis enfin à résoudre le système d'équation linéaire $(J^TJ)w = -J^Tr$

On retrouve alors un système d'équations normales et on peut alors résoudre de ce système par la méthode de Cholesky.

A noter que l'efficacité de cette méthode dépend fortement du choix de vecteur initial de la suite de vecteur. Dans le pire des cas, cette suite pourrait ne pas converger du tout.

Nous n'avons pas programmé cette partie.



Amélioration de Gauss-Newton

Conclusion

Ce projet nous a permis d'appliquer pour la première fois des calculs numériques sur machine et de mettre en place certaines fonctions incontournables comme l'algorithme de descente, de remontée ou de Cholesky. Nous avons eu l'opportunité de formater des connaissances mathématiques en code. De plus, nous avons pu approfondir certains concepts et appliquer pour la première fois une méthode de résolution aux moindres carrés.

Ce projet a également le mérite de tisser un lien entre plusieurs enseignements, de nous faire appréhender le langage Fortran 77 et donc d'approfondir notre culture informatique.