REVUE DE STATISTIQUE APPLIQUÉE

J.-P. ASSELIN DE BEAUVILLE
A. DOLLA

Une méthode de protection du modèle linéaire

Revue de statistique appliquée, tome 28, n° 2 (1980), p. 25-43
<http://www.numdam.org/item?id=RSA_1980__28_2_25_0>

UNE MÉTHODE DE PROTECTION DU MODÈLE LINÉAIRE

J.-P. ASSELIN DE BEAUVILLE et A. DOLLA

Laboratoire d'informatique appliquée – U.E.R. A.G.I.

Parc de Grandmont – 37200 TOURS

1. INTRODUCTION

Dans un précédent article [1] nous avons présenté un algorithme itératif permettant d'utiliser l'estimateur de Huber pour la protection de la régression linéaire simple et polynomiale. Une des conclusions de cette note était relative à la difficulté de choisir un point de départ convenable pour les itérations. En effet nous avons illustré à l'aide d'un exemple (repris au paragraphe II) que le choix d'un mauvais point de départ peut conduire l'algorithme à converger vers un optimum local très éloigné de la solution robuste cherchée. D'autre part, on sait qu'un point de départ plus robuste que la solution des moindres carrés classique entraîne une perte d'efficacité de l'algorithme lorsqu'il n'y a pas de points aberrants dans l'échantillon.

Dans cette note on commence (paragraphe II) par présenter quelques exemples, pris parmi les nombreux articles consacrés au problème, pour lesquels ni la méthode des moindres carrés classique, ni la méthode de Huber classique exposée dans [1] ne conduisent à une solution satisfaisante.

Dans le paragraphe III on décrit un algorithme capable de résoudre correctement les exemples précédents et donner des résultats satisfaisants pour les simulations qui ont été effectuées. On notera que cet algorithme permet de traiter des cas multi-dimensionnels, alors que dans [1] seule la régression linéaire simple avait été abordée.

Suivant l'échantillon étudié cet algorithme fournira une solution choisie parmi les six possibilités suivantes (dont l'exposé détaillé est fait dans le même paragraphe) :

a) méthode des moindres carrés classique (MCO)

b) méthode des plus proches voisins (PPV)

c) solution des moindres carré (MCO) comme point de départ des itérations associées à la méthode de Huber avec la fonction $\psi_1(.)$

d) solution MCO comme point de départ des itérations associée à la méthode de Huber avec la fonction $\psi_2(.)$

Mots-Clés : **Régression, moindres carrés, plus proches voisins, points aberrants ("outliers"), Robustesse.**

e) solution PPV comme point de départ des itérations associée à la méthode de Huber avec la fonction $\psi_1(.)$

f) solution PPV comme point de départ des itérations associée à la méthode de Huber avec la fonction $\psi_2(.)$

(Les fonctions ψ_1 et ψ_2 seront définies au pargraphe III).

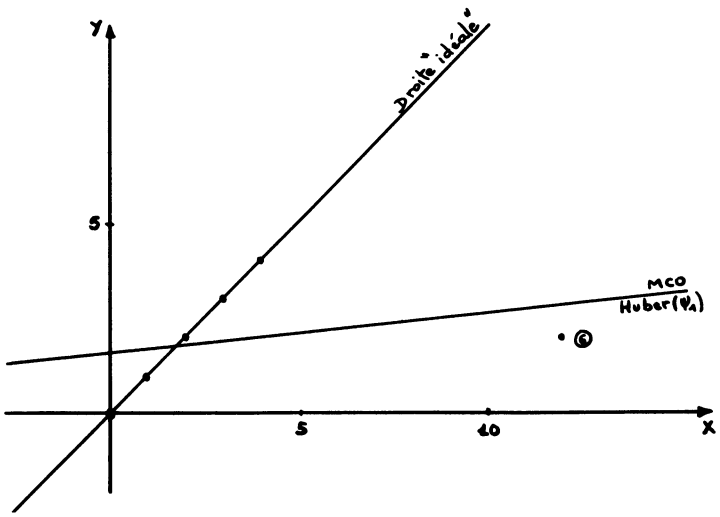
Dans le paragraphe IV on donne les résultats de l'application de cet algorithme d'une part, aux exemples présentés au paragraphe II et, d'autre part à plusieurs suites d'échantillons artificiels.

Enfin le paragraphe V est réservé à la conclusion. Les sous-programme FORTRAN (REGLIM, POIDS, PTIPOI, COND, SOMME, SOMFON et ECAROB) peuvent être obtenus en s'adressant aux auteurs.

2. – LES EXEMPLES TRAITES ET LES SIMULATIONS

L'algorithme décrit au paragraphe III a été appliqué à cinq cas particuliers considérés comme représentatifs de situations difficiles pour les méthodes connues. Pour tous ces exemples la méthode MCO ainsi que la méthode de Huber (ψ_1) ne donnent pas des résultats satisfaisants. (Dans la suite on désignera par méthode de Huber (ψ_1) la solution c) ci-dessus).

1) **Exemple n° 1** : donné par Harvey A.C. [2] de taille N = 6 à 2 dimensions déjà étudié lors d'un précédent article [1].

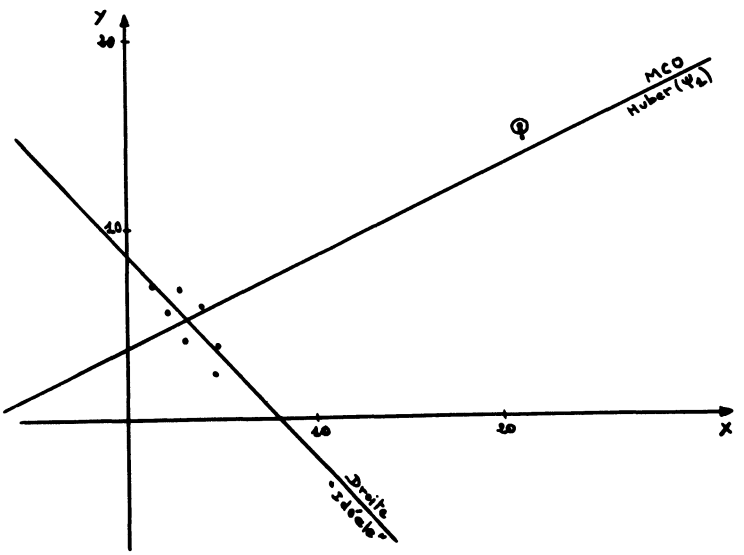


Exemple n° 1

La droite MCO d'équation $y = 0{,}107\,x + 1{,}607$ est très mauvaise ainsi que la droite obtenue par la méthode de Huber (ψ_1) puisque confondue avec la précédente.

La droite MCO obtenue en supprimant le point aberrant n° 6 et qui constitue la bonne solution a pour équation $y = x$.

2) **Exemple n° 2** : donné par Yale et Forsythe [3] de taille N = 8 à 2 dimensions.



Exemple n° 2

La droite MCO a pour équation $y = 0{,}477\,x + 3{,}760$. Celle obtenue par la méthode de Huber (ψ_1) est très voisine : $y = 0{,}475\,x + 3{,}868$.

La droite MCO obtenue en supprimant le point aberrant n° 8 a pour équation $y = -1{,}043\,x + 8{,}472$.

On peut ainsi se rendre compte de l'énorme influence du point aberrant sur les solutions MCO et Huber (ψ_1).

3) **Exemple n° 3** : donné par Daniel et Wood (1971) et étudié par Andrews D.F. [4] de taille N = 21 à 4 dimensions.

Il s'agit de 21 observations obtenues lors de l'oxydation de l'ammoniac par l'acide nitrique en fonction de 3 variables explicatives où 4 observations (n° 1, 3, 4, 21) sont considérées comme aberrantes par les auteurs après une analyse sérieuse.

La méthode MCO donne les coefficients de régression suivants :

–39,93 ; 0,72 ; 1,30 ; –0,15.

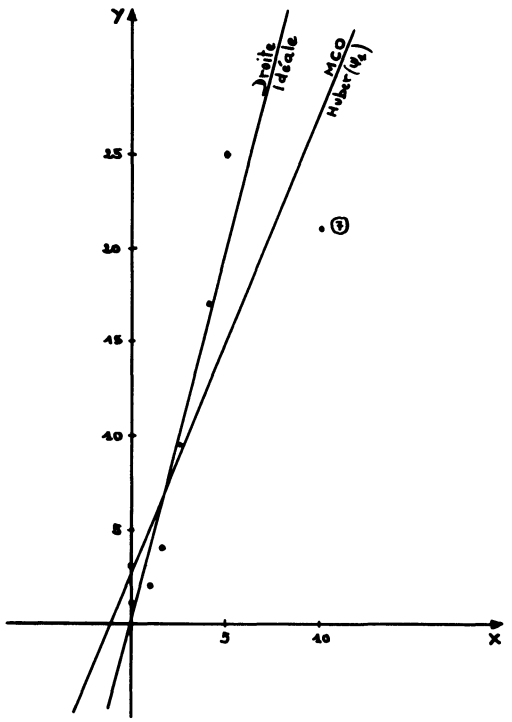
Celle de Huber (ψ_1) donne :

–40,23 ; 0,73 ; 1,26 ; –015 c'est-à-dire des coefficients très proches des précédents.

En supprimant les 4 points aberrants et en calculant la solution MCO sur les 17 points restants on obtient :

–37,67 ; 0,80 ; 0,58 ; –0,07.

4) **Exemple n° 4** : Afin de tester notre algorithme dans le cas d'une pente forte (donc de grande variations des résidus pour de faibles variations de la pente de la droite de régression) on étudie l'exemple suivant à 2 dimensions et de taille N = 8.



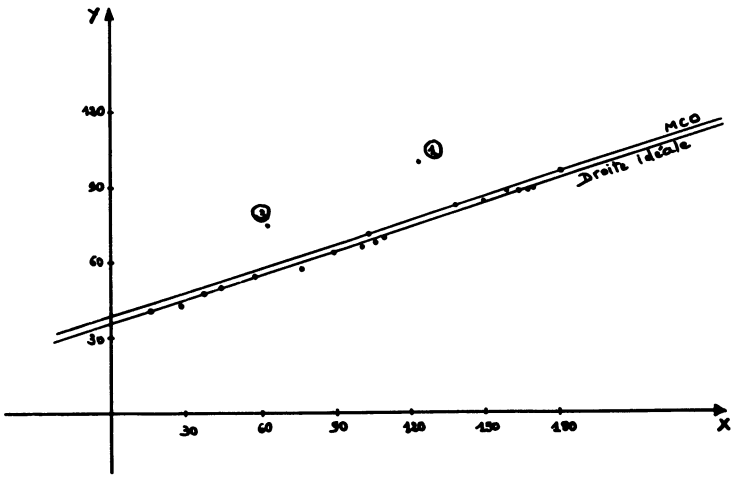
Exemple n° 4

La droite MCO a pour équation y = 2,422 x +2,744.

La droite de Huber (ψ_1) a pour équation y = 2,333 x + 2,674.

Celle obtenue par MCO, le point aberrant n° 7 étant supprimé, a pour équation : y = 4,018 x + 0,175.

5) **Exemple n° 5** : donné par Daniel et Wood (1972) et étudié par Yale et Forsythe [3] de taille N = 20 où les points n° 1 et 3 ont subi une translation de leur ordonnée afin de les rendre aberrants.



Exemple n° 5

La droite MCO a pour équation $y = 0{,}315\ x + 38{,}377$.

La méthode de Huber (ψ_1) donne : $y = 0{,}319\ x + 36{,}076$.

La droite MCO obtenue avec les points n° 1 et 3 dans leurs positions initiales (non aberrants) s'écrit : $y = 0{,}322\ x + 35{,}458$.

6) Simulations

Pour vérifier de manière plus générale la qualité de l'algorithme proposé, celui-ci a été appliqué à des échantillons artificiels à 2 et 3 dimensions, de taille N = 10, 20, 50. Dans tous les cas on a utilisé des erreurs distribuées, soit normalement, soit suivant une loi normale contaminée par une loi normale de variance supérieure.

3. – METHODE ET ALGORITHME

1) Rappels sur la méthode des moindres carrés pondérés

Le modèle linéaire multiple peut s'exprimer par l'équation matricielle :

$$\underline{y} = \underline{X}\,\underline{\beta} + \underline{e} \quad \text{où}$$

$\underline{y}' = (y_1\ y_2 \ldots y_n)$ est la transposée du vecteur $\underline{y}$ qui contient les n réalisations de la variable aléatoire à expliquer.

$$\underline{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{1k} \\ 1 & & & \\ . & & & \\ . & & & \\ . & & & \\ 1 & & & \\ 1 & x_{n1} & x_{n2} & x_{nk} \end{pmatrix}$$ est une matrice

de coefficients connus relative aux k variables explicatives, supposées linéairement indépendantes.

$\underline{\beta}' = (\beta_0\ \beta_1 \ldots \beta_k)$ est la transposée du vecteur $\underline{\beta}$ des coefficients de régression que l'on désire estimer.

$\underline{e}' = (e_1\ e_2 \ldots e_n)$ est la transposée du vecteur aléatoire $\underline{e}$ d'espérance mathématique nulle et de matrice des covariances $\sigma^2\ \underline{I}$ (erreur aléatoire).

On sait que la solution des moindres carrés est donnée par :

$$\underline{\hat{\beta}} = (\underline{X}'\ \underline{X})^{-1}\ \underline{X}'\ \underline{y}\ .$$

Dans le cas plus général où les n observations sont affectées de poids w_i positifs il est pratique de récrire le modèle sous la forme :

$$\underline{z} = \underline{Q}\ \underline{\beta} + \underline{f} \quad \text{avec}$$

- $\underline{z} = \underline{\sqrt{W}}\ \underline{y}$ où

$$\sqrt{W} = \begin{pmatrix} \sqrt{w_1} & & 0 \\ 0 & \sqrt{w_2} & \\ . & & \\ . & & \\ . & & \\ 0 & & \sqrt{w_n} \end{pmatrix}$$ (matrice diagonale non singulière)

- $\underline{Q} = \underline{\sqrt{W}}\ \underline{X}$
- $\underline{f} = \underline{\sqrt{W}}\ \underline{e}$

Le vecteur aléatoire $\underline{f}$ étant d'espérance nulle et de matrice des covariances $\sigma^2\ \underline{I}$ on peut alors appliquer la méthode des moindres carrés à ce nouveau modèle pour obtenir :

$$\beta = (\underline{X}'\ \underline{W}\ \underline{X})^{-1}\ \underline{X}'\ \underline{W}\ \underline{y} \quad \text{où} \quad \underline{W} = \underline{\sqrt{W}}\ .\ \underline{\sqrt{W}}\ .$$

En utilisant différentes définitions pour les poids w_i, la méthode des moindres carrés pondérés fournira donc des estimations différentes des coefficients β_i, $i = 1, \ldots, k$.

a) Méthode des moindres carrés non pondérés (MCO) :

C'est la méthode classique pour laquelle on a $w_i = 1$ pour tout

$$i = 1, 2, \ldots, n\ .$$

b) Méthode de Huber h-windsorisée (Huber ψ_1) :

Elle appartient à la famille des M-estimateurs : voir [7] et [8] pour les aspects théoriques et numériques.

C'est une méthode itérative : partant de la solution des moindres carrés non pondérés (MCO) on définit le poids w_i de chaque points par :

$$w_i = \begin{cases} 1 & \text{si} \quad i \in E_1 \\ r_1 / |e_i| & \text{si} \quad i \in E_2 \end{cases} \quad \text{avec}$$

E_1 : ensemble des indices $i\,(i = 1, 2, \ldots, n)$ pour lesquels on a $|e_i| < r_1$.

E_2 : ensemble des indices i pour lesquels $|e_i| \geqslant r_1$.

$r_1 = h.S.$ où h est un paramètre positif fixé au départ (en général $h = 2$) et S une estimation robuste de la dispersion des résidus e_i (la médiane de la valeur absolue des résidus pour les petits échantillons ou l'écart normalisé entre les percentiles 0,72 et 0,28 pour les grands échantillons :

$$S = (e_{0,72} - e_{0,28})/1\,.\,166).$$

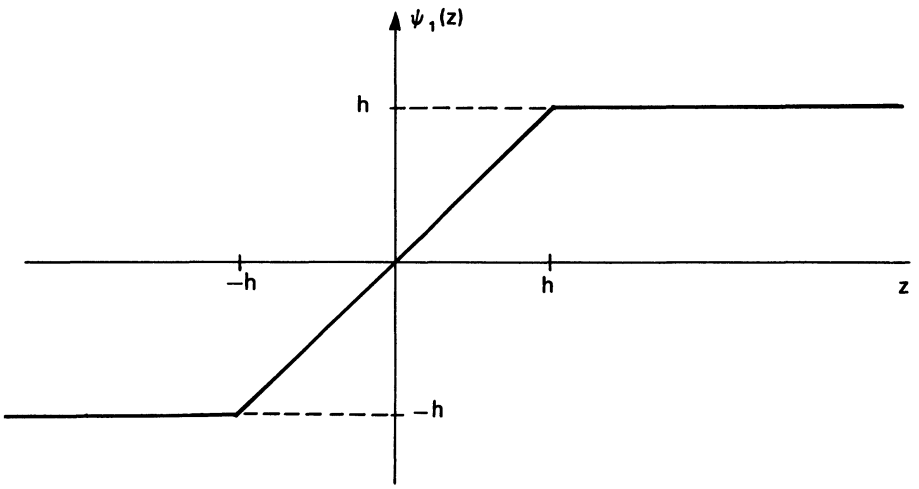
On calcule ensuite la solution des moindres carrés pondérés puis on recommence le processus à partir de cette dernière solution et ceci jusqu'à ce qu'un certain critère d'arrêt des itérations soit satisfait. On pourra consulter [1] à titre d'exemple d'application de cette procédure.

On peut d'ailleurs montrer que le choix de ces poids w_i est identique à la solution du système de $(k + 1)$ équations suivant :

$$\sum_{i=1}^{n} \psi_1(e_i/S) = 0 \qquad \sum_{i=1}^{n} x_{i1}\, \psi_1(e_i/S) = 0 \qquad \sum_{i=1}^{n} x_{ik}\, \psi_1(e_i/S) = 0$$

où la fonction $\psi_1(.)$ est définie par :

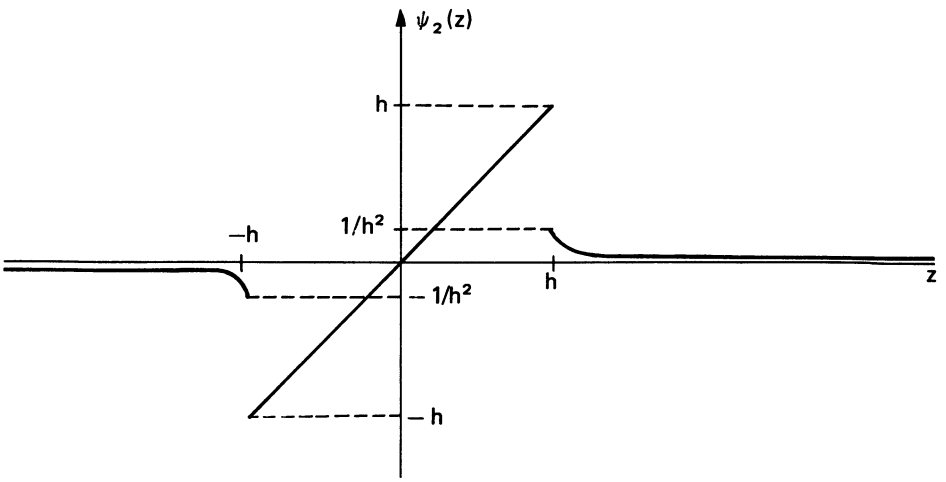
$$\psi_1(z) = \max(-h, \min(h, z)) \quad \text{soit} \quad \psi_1(z) = \begin{cases} z & \text{pour } |z| < h \\ h\,.\,\text{sign}(z) & \text{pour } |z| \geqslant h. \end{cases}$$



c) Méthode de Huber h-windsorisée modifiée (Huber ψ_2) :

Afin de diminuer plus fortement encore les poids affectés aux points aberrante de l'échantillon on définit un nouvelle fonction $\psi(.)$ pour laquelle on pose :

$$\psi_2(z) = \begin{cases} z & \text{pour } |z| < h \\ h\,\text{sign}(z)/|z|^3 & \text{pour } |z| \geqslant h \end{cases}$$



On remarque que par rapport à la fonction $\psi_1(.)$ précédente, cette fonction revient très vite vers zéro lorsque z devient grand. On trouvera dans [4] et [5] d'autres fonctions $\psi(.)$ qui présentent, d'une façon moins accentuée cette décroissance rapide.

L'expression des poids définis à partir de ψ_2 s'écrit :

$$w_i = \begin{cases} 1 & \text{si} \quad i \in E_1 \\ \dfrac{r_1}{e_i^4} & \text{si} \quad i \in E_2 \end{cases}$$

d) Méthodes des plus proches voisins (PPV) :

Comme on l'a souligné plus haut, l'algorithme de Huber (quelle que soit la fonction $\psi(.)$ utilisée) nécessite une première estimation du vecteur $\underline{\beta}$ afin de pouvoir procéder aux itérations successives. En général on choisit comme point de départ la solution fournie par les moindres carrés non pondérés (MCO). En pratique ce choix peut parfois conduire à de très mauvais résultats. C'est le cas, en particulier, lorsqu'il existe une ou plusieurs observations, très éloignées de l'ensemble des données, qui viennent perturber fortement la solution des moindres carrés classique. Il peut arriver alors que les itérations successives ne parviennent pas à "rattraper" le mauvais résultat qui constitue le point de départ et que l'on se déplace vers un optimum local qui ne serait pas la solution robuste cherchée. C'est pour cette raison que nous proposons dans cette note de choisir un point de départ basé sur les plus proches voisins de façon à éliminer l'influence des points aberrants éventuels. Si l'on convient pour chaque observation de retenir les k plus proches voisins, les poids sont alors définis par :

$$w_i = k/\left(\sum_{j \in E_3} d_{ij}^2\right)^2$$

où

E_3 : ensemble des indices j (j = 1 , 2 , . . . , n) des points faisant partie des k plus proches voisins de la $i^{ème}$ observation.

d_{ij} : distance euclidienne entre la $i^{ème}$ observation et la $j^{ème}$.

On remarquera que le poids ainsi affecté sera d'autant plus faible que l'observation considérée sera éloignée de ses k plus proches voisins (PPV).

D'autre part afin d'éviter les problèmes liés à l'inadaptation de la métrique euclidienne au cas où les unités sur les axes seraient très différentes on calcule les distances d_{ij} sur les observations centrées et réduites.

Ces poids w_i sont ensuite utilisés conjointement avec la méthode des moindres carrés pondérés.

2) Méthodologie

Etant donné un échantillon on peut se trouver dans l'un des deux cas suivants :

a) L'échantillon est "propre" c'est-à-dire qu'il ne possède pas de valeurs aberrantes et les erreurs aléatoires sont au moins approximativement normales ;

b) L'échantillon est entaché de valeurs aberrantes ou bien la loi des erreurs est une loi à queues plus importantes que celles de la loi normale.

On sait que le cas a) est résolu de façon satisfaisante par la méthode des moindres carrés ordinaire (MCO). Par contre pour b) cette méthode ne fournit pas de bons résultats. Il convient alors de choisir un procédé plus robuste que MCO [9].

Dans cette note on a tout d'abord opté pour l'algorithme de Huber h-windsorisé (Huber ψ_1) qui a été bien étudié sur le plan théorique. Toutefois il reste des cas (illustrés dans le §II) où celui-ci ne fournit pas de bons résultats.

Devant ces difficultés on a essayé de trouver un meilleur point de départ des itérations : c'est le but de la méthode des plus proches voisins (PPV).

Mais ce nouveau point de départ associé à la fonction ψ_1 ne suffit pas toujours pour fournir une solution satifaisante, aussi on a alors tenté d'améliorer la méthode en utilisant une fonction ψ_2 qui, par rapport à ψ_1, attribue des poids encore plus faibles aux observations éloignées.

Dans une étude réelle (notamment dans un exemple multidimensionnel) il est évident que l'on ne saura pas a priori si on se trouve dans la situation a) ou b). Il devient alors hasardeux d'adopter une méthode particulière plutôt qu'une autre. L'avantage de l'algorithme proposé ici est précisément qu'il effectue lui-même ce choix à partir des données (algorithme auto-adaptatif). En fait ce choix est double :

- d'une part on détermine le point de départ des itérations (PPV ou MCO) ;
- d'autre part on choisit entre les fonctions ψ_1 et ψ_2.

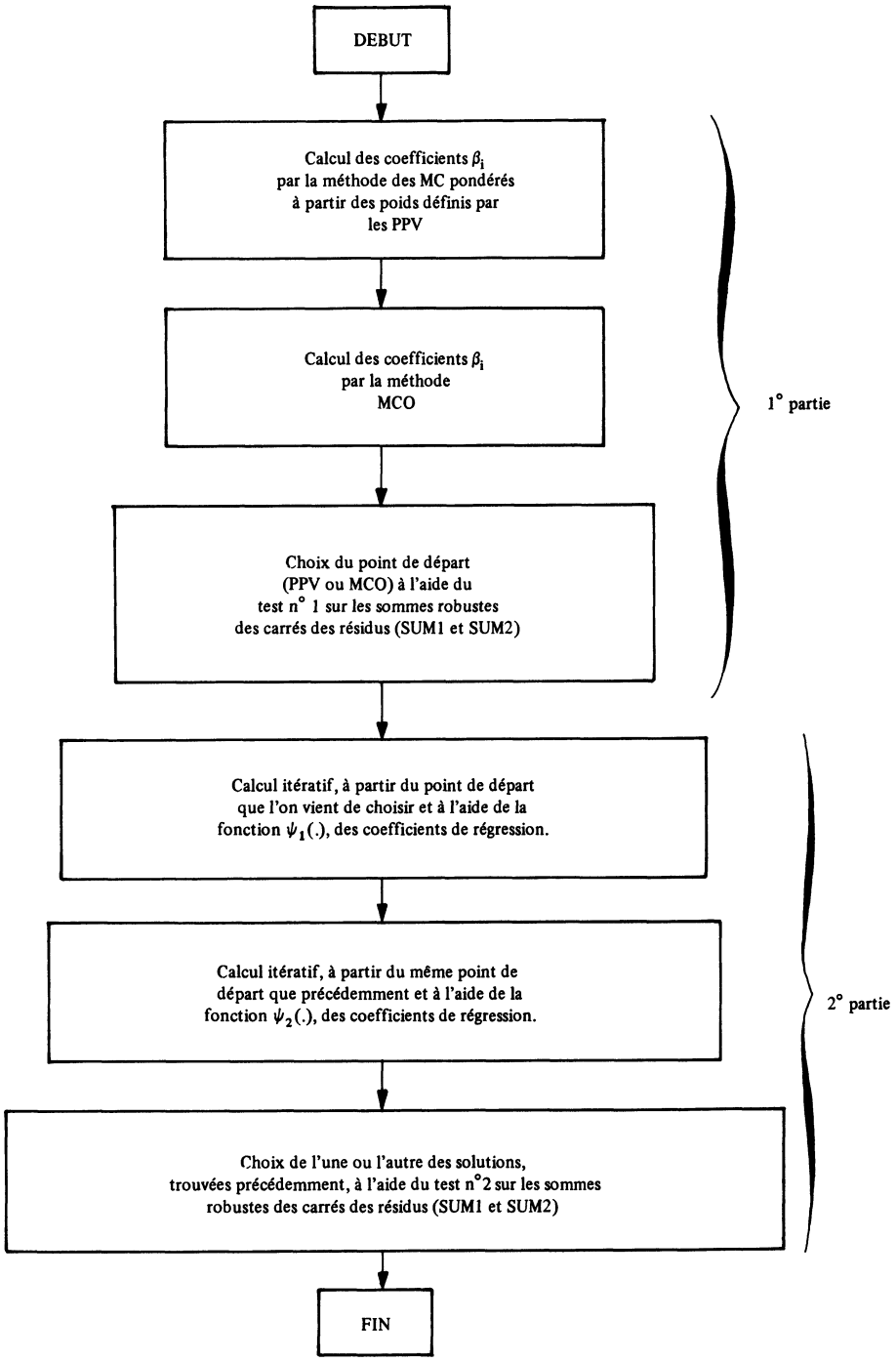
Ainsi on peut espérer que parmi l'ensemble des combinaisons de méthodes possibles (voir § I) on en trouvera une qui donnera des résultats corrects.

Le problème du choix entre les deux fonctions possibles de même que le choix du point de départ des itérations a été résolu de la façon suivante : dans chaque cas on calcule une moyenne tronquée des carrés des résidus ; les résidus supprimés étant toujours pris parmi les plus grands de façon à éliminer l'influence des éventuelles données aberrantes.

La méthode retenue comme étant la meilleure sera celle qui donnera la moyenne tronquée la plus faible. On peut penser en effet que la droite ainsi

obtenue ajustera au mieux la partie la plus dense des observations, les points situés sur la frontière du nuage ayant été exclus du critère de choix.

La méthodologie suivie peut être résumée par l'organigramme simplifié suivant :



Il convient de préciser la manière dont sont calculées les sommes robustes des carrés des résidus qui interviennent à la fin de la première et de la deuxième partie de l'algorithme.

Méthode n° 1 (sous-programme SOMME) pour le test n° 1

Cette méthode permet de calculer les sommes robustes (SUM_1 et SUM_2) à partir desquelles est effectué le choix entre PPV et MCO (Test n° 1).

Les modules des résidus étant classés en ordre non décroissant (O.N.D.) on effectue la somme de (N – 1) premiers termes que l'on multiplie par un facteur variable avec la taille de l'échantillon : PROSUM (défini ci-dessous ; compris entre 0 et 0,75). On obtient SUM que l'on compare avec le résidu dont le module est le plus grand (de rang N) et qui n'a pas été pris en compte précédemment.

Deux cas peuvent se présenter :

1) |résidu de rang N| < SUM

On calcule SUM1 (ou SUM_2 suivant l'état d'avancement de l'algorithme) tel que :

$$SUM_{1 \text{ ou } 2} = \frac{\sum_{i=1}^{N} |\text{résidus}_{1 \text{ ou } 2}|^2}{N} \quad \text{avec}$$

- résidus_1 : résidus obtenus à partir des PPV
- résidus_2 : résidus obtenus à partir de MCO
- N : nombre de points de l'échantillon

2) |résidu de rang N| ⩾ SUM

Ce résidu est écarté. On calcule à nouveau SUM non pas à partir des (N – 1) premiers résidus mais à partir des (N – 2) premiers et en conservant la même valeur pour PROSUM.

Deux cas peuvent encore se présenter :

a) |résidu de rang (N – 1)| < SUM

On calcule alors SUM1 (ou SUM_2) tel que :

$$SUM_{1 \text{ ou } 2} = \frac{\sum_{i=1}^{N-1} |\text{résidus}_{1 \text{ ou } 2}|^2}{N - 1}$$

b) |résidu de rang (N – 1) | ⩾ SUM

On calcule à nouveau SUM avec les (N – 3) premiers résidus . . .

On arrête le processus si le rang du résidu testé est ⩽ NTOUR (défini plus bas : nombre minimum de points conservés).

Méthode n° 2 (sous-programme SOMFON) pour le test n° 2

Elle permet de calculer SUM_1 et SUM_2 pour choisir entre les fonctions ψ_1 et ψ_2 (Test n° 2).

On calcule SUM_1 (ou SUM_2) tel que :

$$SUM_{1 \text{ ou } 2} = \frac{\sum_{i=1}^{n'} |\text{résidus}_{1 \text{ ou } 2}|^2}{n'} \quad \text{avec}$$

- résidus_1 : résidus obtenus à partir de ψ_1
- résidus_2 : résidus obtenus à partir de ψ_2
- n' : nombre de points dont les $|\text{résidus}| \leqslant r_2$ tel que

$$r_2 = \begin{cases} 5S \text{ grands échantillons} \\ 7S \text{ petits échantillons} \end{cases}$$

3) Algorithme

Soit un N-échantillon $(x_{1i}, x_{2i}, \ldots, x_{ki}, y_i)$ où Y représente la variable à expliquer. Dans l'algorithme on utilise divers paramètres définis ci-dessous :

$$MCA = \begin{cases} 0 : \text{si on effectue les calculs par la méthode des moindres carrés pondérés, les poids étant définis à partir des PPV.} \\ 1 : \text{si on effectue les calculs par la méthode des poindres carrés classiques.} \end{cases}$$

$$IEXP = \begin{cases} 1 : \text{exposant intervenant dans l'expression des poids définis à partir de } \psi_1. \\ 4 : \text{exposant intervenant dans l'expression des poids définis à partir de } \psi_2. \end{cases}$$

On a donc :

$$w_i = \begin{cases} 1 \quad \text{si} \quad i \in E_1 \\ \dfrac{r_1}{|e_i|^{IEXP}} \quad \text{si} \quad i \in E_2 \end{cases}$$

h : paramètre utilisé dans la méthode de Huber h-windsorisée (Huber ψ_1) servant à calculer r_1 : $r_1 = h.S$ (§ III 1b).

ALP et DIFCOE : paramètres fixes permettant l'arrêt des itérations pour le calcul des coefficients β^j , j = 1 , M.

Pour tous j , j = 1 , M , on teste la variation relative :

$\dfrac{\beta_i^j - \beta_{i-1}^j}{\beta_{i-1}^j}$ par rapport à ALP et éventuellement la variation absolue $(\beta_i^j - \beta_{i-1}^j)$ par rapport à DIFCOE où

β_i^j : valeur de β^j à la $i^{\text{ème}}$ itération

β_{i-1}^j : valeur de β^j à la $(i-1)^{\text{ème}}$ itération.

Pour tous les coefficients β^j on teste la variation relative du coefficient par rapport à ALP : si pour tous j la variation est inférieure ou égale à ALP on arrête les itérations ; si cette variation est supérieure à ALP pour au moins une valeur de j on teste la variation absolue correspondante. Si celle-ci est supérieure à DIFCOE on continue les itérations. Sinon on les arrête.

H1 : paramètre utilisé dans le sous-programme SOMFON pour le calcul de $r_2 = H1 \, . \, r_1$.

Pour $N \leqslant 10$ $r_2 = 5S$

$N > 10$ $r_2 = 7S$

Les différentes étapes de l'algorithme sont les suivantes :

(1) Définition de différentes constantes :

MCA = 0 ; IEXP = 1 ; h = 2 ; ALP = 0,02 ; DIFCOE = 0,01

(2) Définition de différents paramètres suivant la taille de l'échantillon :

a) $N > 10$: H1 = 2,5 ; PROSUM = 3/N ; NTOUR = N-Partie entière (N x 0,2) (on postule qu'a priori l'échantillon ne contient pas plus de 20 % de données aberrantes).

b) $N \leqslant 10$: H1 = 3,5 ; PROSUM = 0,75 ; NTOUR = N – 2

(3) ITER = 1

(4) Calcul des valeurs moyennes, des écarts-types, des valeurs centrées réduites pour les variables et les observations.

(5) Initialisation de la matrice des poids calculés avec PPV.

(6) Calcul des coefficients de régression à l'aide de la méthode des moindres carrés pondérés ou classiques.

(7) Réinitialisation de la matrice des poids :

$$w_{ij} = 1 \ \forall \, i = j \quad \text{et} \quad w_{ij} = 0 \ \forall \, i \neq j \, .$$

(8) Calcul des résidus des points à partir des coefficients de régression trouvés en (6).

(9) Test sur la valeur de N :

a) $N > 10$

1. Classement en OND des résidus
2. Calcul de l'écart robuste des résidus par appel au sous-programme ECAROB. On déduit $r_1 = h \times S$
3. Si ITER $\neq$ 1 aller en (12)
4. Classement en OND des modules des résidus
5. Calcul de SUM1 ou SUM2 par la méthode n° 1 exposée plus haut et réalisée par l'appel au sous-programme SOMME ; ensuite aller en (10)

b) $N \leqslant 10$

1. Classement en OND des modules des résidus
2. Calcul de l'écart robuste des résidus par la médiane puis de $r_1 = h \times S$
3. Si ITER $\neq$ 1 aller en (12)
4. Aller en (9) a) 5.

(10) Test sur MCA :

- Si MCA = 0 (PPV vient d'être exécuté) on pose MCA = 1 (afin d'effectuer MCO) et aller en (6)
- Si MCA = 1 (PPV et MCO ont été faits) on continue en (11)

(11) Comparaison de SUM1 et SUM2 :

On garde pour point de départ la solution fournie par MCO si SUM1 $\geqslant$ SUM2 ou par PPV si SUM1 $<$ SUM2.

(12) Calcul des poids par appel aux sous-programmes POIDS ou PTIPOI (suivant la taille de l'échantillon).

(13) Si ITER = 1 et KONT = N : aller en (17).

KONT représente le nombre de points dont les modules des résidus sont inférieurs à r_1 ; on arrête donc les itérations.

- Si ITER $\neq$ 1 et KONT = N aller en (15)
- Si ITER $\neq$ 1 et KONT $\neq$ N aller en (14)
- Si ITER = 1 et KONT $\neq$ N faire ITER = ITER + 1 et aller en (6)

(14) Tester les variations relatives et les variations absolues de tous les coefficients de régression (par rapport à ALP et DIFCOE).

- Si pour chaque coefficient au moins l'un des deux tests est vérifié aller en (15)
- Si non faire ITER = ITER + 1 et aller en (6)

(15) Calcul des sommes robustes pour le choix entre les fonctions ψ_1 et ψ_2 par la méthode n° 2 (exposée plus haut) ; appel du sous-programme SOMFON : on obtient SUM1 avec ψ_1 ou SUM_2 avec ψ_2.

- Si on a SUM1 : poser ITER = 2, IEXP = 4 et appel de POIDS ou PTIPOI puis aller en (6)

  En effet on continue les calculs à partir des coefficients déjà trouvés à la fin de la première partie (PPV ou MCO) mais pour la fonction ψ_2.
- Si on a SUM2 on continue

(16) Comparaison de SUM1 et SUM2 :

On retient la solution qui correspond à la plus petite somme.

(17) FIN : arrêt de la procédure.

4) Les sous-programmes utilisés

a) Le SP "COND" :

SUBROUTINE COND (RES, N, ITR)

RES : vecteur de taille N contenant les valeurs à classer au début de l'exécution du SP puis les observations classées après l'exécution.

ITR : vecteur de taille N contenant les indices initiaux des observations une fois le classement effectué.

b) Le SP "ECAROB" :

SUBROUTINE ECAROB (RES, N, S)

Il calcule l'écart robuste S des N résidus classés (RES) à partir des écarts normalisés entre les percentiles 0,28 et 0,72.

c) Le SP "SOMME" :

SUBROUTINE SOMME (YTR, N, NTOUR, PROSUM, SUMROB)

YTR : vecteur contenant les N modules des résidus classés en OND.

NTOUR : nombre minimum de points conservés pour le calcul de SUMROB.

PROSUM = 3/N pour les échantillons de taille supérieure à 10.

= 0,75 pour les échantillons de taille inférieure ou égale à 10.

Ce sous-programme calcule $\text{SUMROB} = \dfrac{\sum_{i=1}^{n_1} (\text{résidus})^2}{n_1}$ avec

n_1 : nombre de points conservés après application de la méthode n° 1.

SUMROB est ensuite appelé SUM1 ou SUM2 dans le programme appelant REGLIM.

d) LE SP "SOMFON" :

SUBROUTINE SOMFON (YTR, N, R2, SUM)

YTR : vecteur contenant les N modules des résidus classés en OND.

R_2 : correspond à :

$$r_2 = \begin{cases} 5S & \text{pour les échantillons de taille supérieure à 10.} \\ 7S & \text{pour les échantillons de taille inférieure ou égale à 10.} \end{cases}$$

Ce sous-programme calcule $\text{SUM} = \dfrac{\sum_{i=1}^{n'} (\text{résidus})^2}{n'}$ avec

n' : nombre de points dont les modules des résidus sont inférieurs ou égaux à r_2.
SUM est ensuite appelé SUM1 ou SUM2 dans REGLIM.

Ainsi on élimine du calcul de cette valeur moyenne les résidus trop grands (points aberrants).

e) Le SP "POIDS" :

SUBROUTINE POIDS (RES, ITR, N, R1, IEXP, N2, KONT, W, WW).

RES : vecteur classé des N résidus.

ITR : vecteur de taille N contenant les indices initiaux des observations une fois classées.

IEXP = 1 ou 4 suivant qu'on utilise la fonction ψ_1 ou la fonction ψ_2.

R1 : correspond à $r_1 = 2S$

KONT (argument de retour) : nombre de points dont le module du résidu est inférieur à r_1.

Ce sous-programme utilisé quand $N > 10$ calcule les poids des points en utilisant la fonction ψ_2 $\left(\text{poids en } \frac{r_1}{e_i^4}\right)$ ou la fonction ψ_1 $\left(\text{poids en } \frac{r_1}{|e_i|}\right)$.

f) Le SP "PTIPOI" :

SUBROUTINE PTIPOI (RES, N, R1, IEXP, N2, KONT, W, WW)

RES : vecteur non classé des N résidus.

Ce sous-programme utilisé si $N \leqslant 10$, effectue les mêmes calculs que "POIDS".

4. RESULTATS

1) Exemple n° 1

On trouve l'équation $y = x$. Ce résultat est obtenu par le choix du couple [PPV, $\psi_2(.)$] c'est-à-dire par la méthode PPV à la fin de la première partie comme point de départ et par la fonction $\psi_2(.)$ à la fin de la deuxième partie, après itérations.

2) Exemple n° 2

On trouve l'équation $y = -1{,}043x + 8{,}471$ par le choix du couple [PPV, $\psi_2(.)$].

3) Exemple n° 3

On obtient les coefficients de régression suivants :

$-37{,}67$; 0,80 ; 0,58 ; $-0{,}07$ une nouvelle fois par le choix du couple [PPV, $\psi_2(.)$].

4) Exemple n° 4

L'équation s'écrit $y = 4{,}018x + 0{,}176$. Cette solution est obtenue par le choix du couple [MCO + ψ_2].

5) Exemple n° 5

L'équation trouvée s'écrit $y = 0{,}321\,x + 35{,}504$ par le choix du couple [PPV + ψ_2].

On notera que l'algorithme décrit au paragraphe III donne pour ces cinq exemples exactement les mêmes résultats que la méthode des moindres carrés appliquée alors que les points aberrants ont été éliminés. Cet algorithme "ignore" donc totalement les points aberrants pour fournir la bonne valeur des coefficients de régression.

De plus il faut remarquer que dans quatre exemples sur cinq le couple choisi est [PPV, ψ_2] ce qui confirme la meilleure qualité du point de départ par les PPV, et de la fonction ψ_2 pour les itérations dans la plupart des cas où l'échantillon présente des points aberrants.

6) Résultats des simulations

L'efficacité de l'ajustement au modèle théorique fourni par l'algorithme est mesuré grâce à l'erreur quadratique moyenne relative :

$$R_\gamma = \sum_{j=1}^{n''} (\hat{\gamma}_j - \gamma)^2 / \sum_{j=1}^{n''} (\gamma_j^* - \gamma)^2 \quad \text{où :}$$

n'' : nombre d'échantillons artificiels (100).

γ : coefficient de régression pour la population.

$\hat{\gamma}_j$: coefficient de régression donné par la méthode des moindres carrés pour le $j^{ème}$ échantillon.

γ_j^* : coefficient de régression fourni par l'algorithme étudié pour le $j^{ème}$ échantillon.

Une valeur de R_γ supérieure à un indiquera que l'algorithme proposé fourni une estimation meilleure que la méthode des moindres carrés puisque dans cette hypothèse, les valeurs de γ_j^* seront statistiquement plus proches de γ que les valeurs de $\hat{\gamma}_j$.

Les résultats sont consignés dans les tableaux ci-dessous.

a) *Régression à 2 dimensions* (y = a + bx)

	N = 10	N = 20	N = 50
R_a	0,85	0,93	0,96
R_b	0,87	0,92	0,94

Tableau 1 : erreurs distribuées suivant N(0,1)

	N = 10	N = 20	N = 50
R_a	1,33	1,56	1,79
R_b	1,43	1,69	1,99

Tableau 2 : erreurs distribuées suivant 0,8 N(0,1) + 0,2 N(0,4)

b) *Régression à 3 dimensions* ($y = a + bx_1 + cx_2$)

	N = 10	N = 20
R_a	0,95	0,93
R_b	0,91	0,91
R_c	0,93	0,93

Tableau 3 : erreurs distribuées suivant N(0,1)

	N = 10	N = 20
R_a	1,16	1,53
R_b	1,15	1,40
R_c	1,22	1,60

Tableau 4 : erreurs distribuées suivant 0,8 N(0,1) + 0,2 N(0,4)

L'examen de ces tableaux montre l'excellente qualité de l'algorithme dans le cas des distributions normales contaminées ; celle-ci augmente d'ailleurs avec la taille de l'échantillon comme pour l'estimateur de Huber (ψ_1) mais en lui restant toujours supérieure comme on peut le constater par comparaison avec les résultats donnés dans [1].

Par contre l'estimateur de Huber (ψ_1) se montre supérieur dans le cas des distributions normales non contaminées.

Cet algorithme obtient son meilleur "rendement" pour des échantillons qui possèdent des points aberrants et en conséquence on peut particulièrement recommander son emploi dans tous les cas où l'on a quelques doutes sur la qualité des données.

Enfin indiquons que la taille de l'ensemble des sous-programmes utilisés : REGLIM, COND, ECAROB, SOMME, SOMFON, POIDS et PTIPOI est d'environ 8K pour un échantillon de taille N = 20.

Les temps de calcul (exécution seulement) sur un ordinateur CII 10070 sont pour des échantillons à deux dimensions de l'ordre de 0,5'' ($N \leqslant 10$), 1''40 (N = 20) et 8''50 (N = 50).

Ce temps d'exécution croît évidemment avec le nombre de dimensions de l'échantillon : pour N = 20 il passe de 1''40 à 2'' pour trois dimensions et à 3'' pour quatre dimensions.

On notera que ces temps de calcul sont en N^2.

5. CONCLUSION

L'algorithme étudié est en fait essentiellement un moyen de choisir parmi plusieurs solutions plus ou moins sensibles à la présence d'observations erronées dans l'échantillon. Cette technique donne d'excellents résultats lorsqu'elle est appliquée aux exemples, réputés difficiles, du paragraphe II. En outre, les quelques simulations effectuées soulignent sa robustesse puisqu'en l'absence de points aberrants, et dès que l'échantillon est de taille supérieure ou égale à vingt, son efficacité dépasse 90 % .

On notera enfin que la pondération des observations à partir de leurs k plus proches voisins constitue un bon point de départ pour la recherche d'une solution itérative du problème de la régression linéaire. Cette façon de procéder, semble-t-il originale, est à rapprocher de la régression par boules de Benzécri [6] pour le traitement de la régression fonctionnelle. Une des différences importante est que l'approximation donnée par cette dernière n'est en général pas linéaire.

BIBLIOGRAPHIE

[1] ASSELIN DE BEAUVILLE J.P. et DOLLA A. – Etude par simulation d'un estimateur de Huber pour la protection de la régression linéaire et polynômiale. *Les Cahiers de l'Analyse des Données,* Vol. IV, 1979, n° 2, p. 147-158.

[2] HARVEY A.C. – A comparison of preliminary estimators for robust regression *J.A.S.A.*, Décembre 1977, Vol. 72, Nombre 360, p. 910-913.

[3] YALE C. et FORSYTHE A.B. – Winsorized regression. *Technométrics,* Vol. 18, n° 3, 1976, p. 291-300.

[4] ANDREWS D.F. – A robust method for multiple linear regression. *Technometrics,* Vol. 16, n° 4, 1974, p. 523-531.

[5] HAMPEL et autres (1972). – Robust Estimates of Location, Survey and Advances. Princeton University Press.

[6] BENZECRI J.P. – Méthodes statistiques de la taxinomie ; 1ère partie, Laboratoire de Statistiques Mathématiques (Université de Paris VI) Paris 1974.

[7] HUBER P.J. – Robust regression : asymptotics, conjectures and Monte-Carlo. *The annals of Statistics,* Vol. 1, 1973, p. 799-821.

[8] DUTTER R. – *J. Stat. Comput. Simul.*, Vol. 5, 1977, p. 207-238.

[9] MORINEAU A. – Regressions robustes. Méthodes d'ajustement et de validation ; *Revue de Statistique Appliquée,* Vol. 25, n° 3, 1978, p. 5-28.