Université PARIS-XII — Faculté de Sciences économiques et de gestion
Master *Sciences économiques*
Mention *Économie quantitative et expertise internationale*
Économétrie III — *F. LEGENDRE* — année universitaire 2005-06

Fiche de travail n°7
Modèles *logit* et *probit*

Je vous propose dans cette septième fiche de travail de prendre connaissance avec les modélisations *logit* et *probit*.

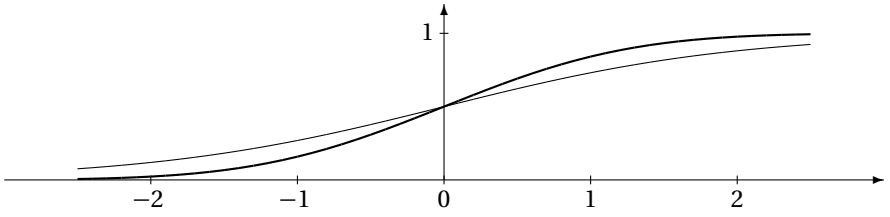
1 Les fonctions de répartition des lois normale et logistique

On note traditionnellement $\Phi(\cdot)$ la fonction de répartition de la loi normale centrée réduite et $\Lambda(\cdot)$ la fonction de répartition de la loi logistique «standard». Ces deux fonctions sont respectivement égales à

$$\begin{aligned}\Phi(z) &= \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-t^2}{2}\right) dt \\ \Lambda(z) &= \frac{1}{1+\exp(-z)}\end{aligned}$$

Ces deux fonctions sont portées sur la figure 1. On voit notamment, comme toutes fonctions de répartition, que $\lim_{z\to-\infty}\Phi(z)=\lim_{z\to-\infty}\Lambda(z)=0$, que $\lim_{z\to\infty}\Phi(z)=\lim_{z\to\infty}\Lambda(z)=1$ et que ces deux fonctions sont monotones croissantes.

FIG. 1 – Fonctions de répartition de la loi normale centrée réduite (trait gras) et de la loi logistique «standard» (trait maigre)



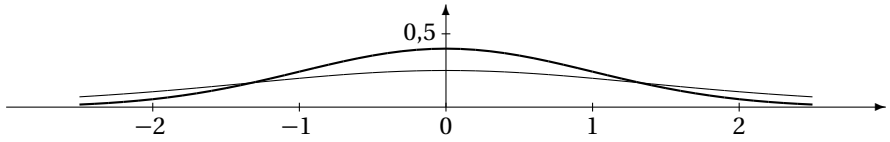
Les fonctions de densité, représentées sur la figure 2, sont respectivement égales à

$$\Phi'(z) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-z^2}{2}\right)$$

$$\Lambda'(z) = \frac{\exp(-z)}{[1+\exp(-z)]^2} = \Lambda(z)[1-\Lambda(z)]$$

Ces fonctions sont paires — $\Phi'(-z)=\Phi'(z)$ et $\Lambda'(-z)=\Lambda(z)$. Aussi les fonctions de répartition vérifient-elles la propriété suivante : $\Phi(-z)=1-\Phi(z)$ et $\Lambda(-z)=1-\Lambda(z)$. On en déduit notamment que $\Phi(0)=0{,}5$ et $\Lambda(0)=0{,}5$. Enfin, il faut noter que la densité de la loi logistique peut s'exprimer à partir de sa fonction de répartition. Ceci conduit à simplifier un grand nombre de calculs et constitue la raison principale de l'utilisation de cette fonction de répartition.

FIG. 2 – Fonctions de densité de la loi normale centrée réduite (trait gras) et de la loi logistique «standard» (trait maigre)



Ces distributions sont très différentes. Les queues de la logistique sont particulièrement épaisses ; sa densité pour les valeurs centrales est beaucoup plus faible que la densité de la loi normale. On a $\Lambda'(0)=\Lambda(0)(1-\Lambda(0))=0{,}5(1-0{,}5)=0{,}25$; en revanche, on a $\Phi'(0)\approx 0{,}4$. Cette comparaison n'est pas très rigoureuse parce que la variance de la loi logistique «standard» est égale à $\pi^2/3\approx 3{,}29$; elle est donc beaucoup plus grande que la variance de la loi normale centrée réduite.

2 Engendrer artificiellement les données d'une modélisation *logit*

Comme dans la fiche de travail n°5, on engendre artificiellement les données en utilisant le générateur de nombres pseudo-aléatoires de SAS. Plus précisément, on va coder l'étape `DATA` suivante :

```
DATA table ;
  DO i = 1 TO 100 ;
    x = NORMAL(123456) ;
    u = RANUNI(123456) ;
    DO WHILE ( (u EQ 0) OR (u EQ 1) ) ;
      u = RANUNI(123456) ;
    END ;
    y = ( 2*x + LOG(u/(1-u)) >= 0 ) ;
```

```
    y_tilde = 4 * (y-.5) ;
    OUTPUT ;
  END ;
RUN ;
```

Le modèle est donc :

$$y_i^\star = \alpha x_i + u_i \quad i = 1,\dots,100 \quad \text{et} \quad y_i = \begin{cases} 1 & \text{si} \quad y_i^\star \geq 0 \\ 0 & \text{si} \quad y_i^\star < 0 \end{cases}$$

où $y_i^\star$ est la variable latente non observable et où u_i suit une loi logistique «standard».

Les valeurs prises par la variable explicative sont tirées, pseudo-aléatoirement, dans une loi normale centrée réduite. Pour obtenir une pseudo réalisation de la loi logistique, on utilise le fait que si la variable aléatoire v suit une loi uniforme comprise entre 0 et 1, alors $\Lambda^{-1}(v)$ suit une loi logistique «standard». Cette propriété n'est pas particulière à la loi logistique ; elle est vérifiée pour toutes fonctions de répartition bien élevées.

La documentation de SAS, pour ce qui a trait à la fonction `RANUNI`, dispose que «The RANUNI function returns a number that is generated from the uniform distribution on the interval (0,1)». Il nous faut donc exclure les cas où le résultat serait 0 ou 1 puisque $\Lambda^{-1}(\cdot)$ n'est pas définie dans ces deux cas. C'est l'objet de la boucle `DO WHILE ( (u EQ 0) OR (u EQ 1) ) ;` : tant que le résultat est égal à 0 ou à 1, on appelle une nouvelle fois la fonction `RANUNI` pour obtenir une autre valeur pour la variable `u`.

Pour le coefficient α, on prend la valeur 2. Aussi l'expression `2*x + LOG(u/(1-u))` est-elle égale à $y_i^\star$. Il faut ensuite bien comprendre que l'expression `2*x + LOG(u/(1-u)) >= 0` est une expression logique qui est égale soit à «vrai» soit à «faux». En SAS — mais plus généralement en informatique — «vrai» se code 1 et «faux» se code 0. Cette expression est donc égale à y_i.

3 La résolution numérique de la maximisation de la vraisemblance

On a

$$\text{Prob}(y=1) = \text{Prob}(y^\star \geq 0) = \text{Prob}(\alpha x + u \geq 0) = \text{Prob}(u \geq -\alpha x),$$

comme la distribution logistique est symétrique, on en déduit que

$$\text{Prob}(u \geq -\alpha x) = \text{Prob}(u < \alpha x) = \Lambda(\alpha x)$$

De même, *mutatis mutandis*, on trouve

$$\text{Prob}(y=0) = \Lambda(-\alpha x) = 1 - \Lambda(\alpha x)$$

La log-vraisemblance de l'échantillon est

$$\sum_{i\ /\ y_i=0} \ln[\text{Prob}(y_i=0)] \ + \sum_{i\ /\ y_i=1} \ln[\text{Prob}(y_i=1)]$$

Cette expression se réécrit, de façon commode mais dépendante du codage utilisé, comme suit :

$$\sum_{i=1}^{\text{N}} \{y_i \ln[\Lambda(\alpha x_i)] + (1-y_i)\ln[1-\Lambda(\alpha x_i)]\}$$

Par rapport à la fiche de travail n°6, la log-vraisemblance doit être identifiée à la fonction $f(\cdot)$ dont on cherche le maximum et le coefficient α à la variable θ. La suite de l'algorithme de NEWTON est donc définie par

$$\alpha_j = \alpha_{j-1} - \frac{f'(\alpha_{j-1})}{f''(\alpha_{j-1})} \quad j = 1,2,\dots \quad \text{et} \quad \alpha_0 \text{ fixé}$$

La dérivée de la log-vraisemblance est *a priori* compliquée :

$$f'(\alpha) = \sum_{i=1}^{\text{N}} \left[y_i \frac{x_i \Lambda'(\alpha x_i)}{\Lambda(\alpha x_i)} + (1-y_i) \frac{-x_i \Lambda'(\alpha x_i)}{1-\Lambda(\alpha x_i)} \right]$$

En remplaçant $\Lambda'(\cdot)$ par son expression en fonction de $\Lambda(\cdot)$ et après quelques manipulations algébriques, on trouve quelque chose de beaucoup plus simple :

$$f'(\alpha) = \sum_{i=1}^{\text{N}} [y_i - \Lambda(\alpha x_i)] x_i$$

La dérivée deuxième est alors égale à

$$f''(\alpha) = -\sum_{i=1}^{\text{N}} \Lambda(\alpha x_i)[1-\Lambda(\alpha x_i)] x_i^2$$

La suite de l'algorithme de NEWTON s'exprime ainsi

$$\alpha_j = \alpha_{j-1} + \frac{\sum_{i=1}^{\text{N}} [y_i - \Lambda(\alpha_{j-1} x_i)] x_i}{\sum_{i=1}^{\text{N}} \Lambda(\alpha_{j-1} x_i)[1-\Lambda(\alpha_{j-1} x_i)] x_i^2} \quad j = 1,2,\dots \quad \text{et} \quad \alpha_0 \text{ fixé}$$

SAS prend $\alpha_0 = 0$. La première itération de l'algorithme devient

$$\alpha_1 = \frac{\sum_{i=1}^{\text{N}} (y_i - 0{,}5) x_i}{\sum_{i=1}^{\text{N}} 0{,}5(1-0{,}5) x_i^2}$$

En définissant le vecteur $\underline{\widetilde{y}}$ comme $4(\underline{y} - 0{,}5\underline{e}_N)$, α_1 s'écrit $\underline{x}'\underline{\widetilde{y}}/\underline{x}'\underline{x}$ et apparaît comme l'estimation des MCO du modèle $\underline{\widetilde{y}} = \alpha\underline{x} + \underline{u}$.

La documentation de SAS explique que « PROC PROBIT always models the probability of response levels at the beginning of the ordering » et que le mot-clé `ORDER`, quand il est égal à `DATA`, spécifie que « [the ordering is the] order of appearance in the input data set ». Notre codage est donc, *a priori*, à l'inverse de la convention que retient SAS. Il faut, pour contourner cela, *i)* trier la table en fonction de y en ordre décroissant *ii)* coder `ORDER = DATA`.

Le programme SAS, en définitive, est le suivant.

```
PROC sort DATA = table OUT = table ;
  BY DESCENDING y ;
RUN ;
PROC probit DATA = table ORDER = DATA ;
  CLASS y ;
  MODEL y = x / NOINT DISTRIBUTION = LOGISTIC ITPRINT ;
RUN ;
PROC reg DATA = table ;
  MODEL y_tilde = x / NOINT ;
RUN ;
```

Les points suivants peuvent être notés.

1. le mot-clé `DESCENDING` de l'instruction `BY` de la `PROC sort` permet d'indiquer que le tri, pour la variable qui suit, doit être effectué en ordre décroissant ;
2. l'instruction `CLASS` dans la `PROC probit` permet de désigner les variables qui représentent des caractéristiques discrètes, ces variables sont soit la variable à modéliser soit des variables explicatives qui s'utilisent comme dans la `PROC glm` ;
3. l'option `LOGISTIC` du mot-clé `DISTRIBUTION` permet de mettre en œuvre une régression *logit* ;
4. le mot-clé `ITPRINT` commande l'impression des itérations de l'algorithme de NEWTON de maximisation numérique de la vraisemblance ;
5. la `PROC reg` finale a pour objet d'illustrer la proposition relative à la première itération de l'algorithme de NEWTON.