

Maitrise Econométrie

Université d'Orléans

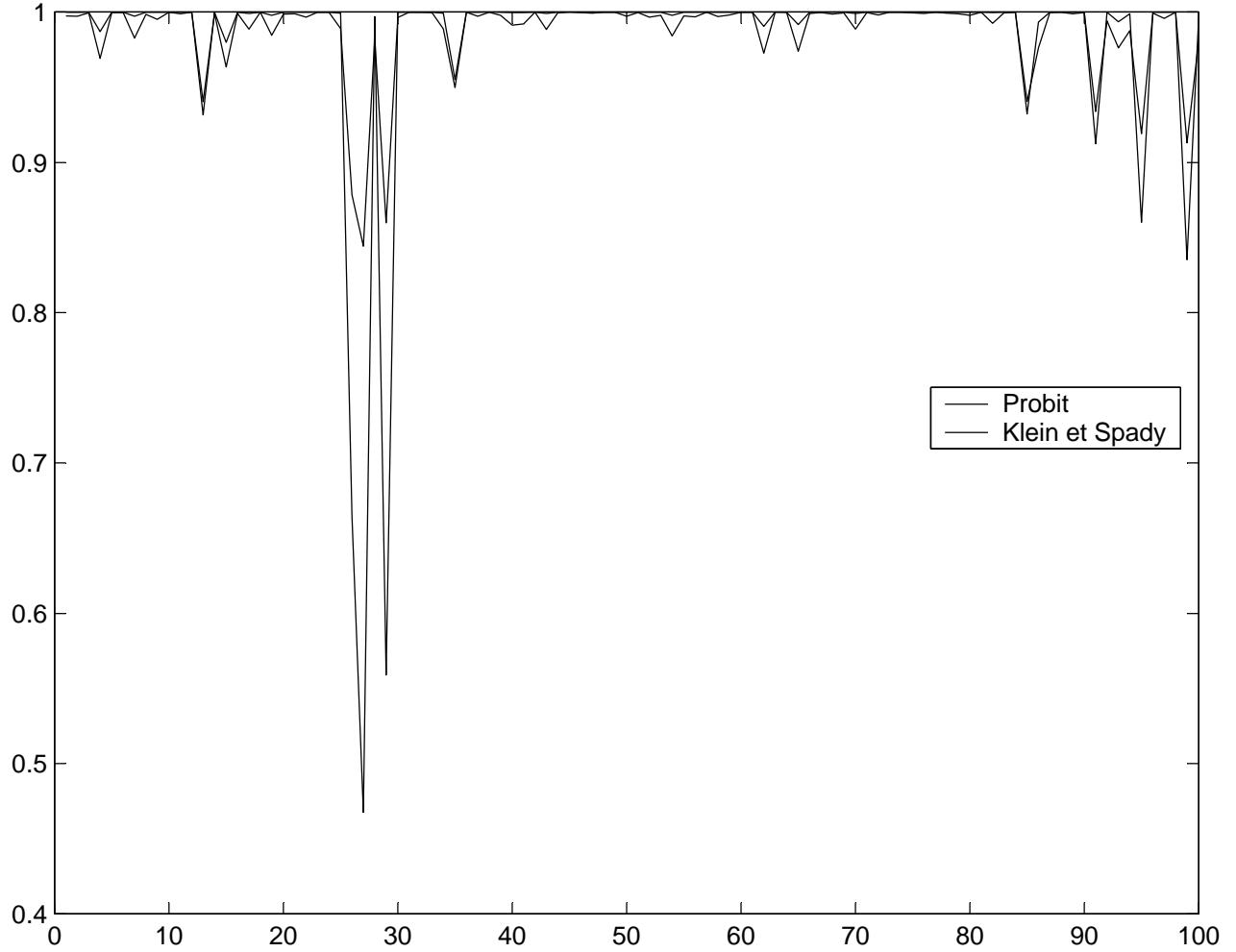
Annexes : Programmes Matlab

Estimations Non Paramétriques et Semi Paramétriques
d'un Modèle Dichotomique sur Données Simulées

Christophe HURLIN

Octobre 2003

Probabilités Estimées



Exemple de réalisation

Programme Prog1_Score

Probit Maximum Likelihood Estimates

McFadden R-squared = 0.0536

Estrella R-squared = 0.0674

LR-ratio, 2*(Lu-Lr) = 67.9215

LR p-value = 0.0000

Log-Likelihood = -600.2179

of iterations = 5

Convergence criterion = 4.5417526e-011

Nobs, Nvars = 1000, 2

of 0's, # of 1's = 330, 670

Variable	Coefficient	t-statistic	t-probability
variable 1	0.451701	10.713724	0.000000
variable 2	0.687519	8.007546	0.000000

Méthode du Score

Condition Initiale = 0.45 Condition Initiale = 0.69

Solution = 0.477 Solution = 0.672

Comparaison des Probabilités Estimées et Prévisions : individus de 1 à 10

1 1
1 1
0 0
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1

Nombre de Previsions Différentes = 22

>>

Exemple de réalisation

Programme Prog2_Klein_Spady

Modèle Probit : Maximum de Vraisemblance

Probit Maximum Likelihood Estimates

McFadden R-squared = 0.5411

Estrella R-squared = 0.1416

LR-ratio, 2*(Lu-Lr) = 10.6095

LR p-value = 0.0011

Log-Likelihood = -4.4991

of iterations = 11

Convergence criterion = 8.3049087e-010

Nobs, Nvars = 100, 2

of 0's, # of 1's = 2, 98

Variable	Coefficient	t-statistic	t-probability
variable 1	4.537603	2.159871	0.033222
variable 2	4.126800	1.749810	0.083281

Estimation Semi Paramétrique (Klein et Spady 1993)

Condition Initiale

4.5376 4.1268

Solution

4.6581 4.1204

Exemple Individu 27

Proba Estimée Probit 0.47

Proba Estimée Klein Spady 0.84

Realisation 1

```
function LogVrais_Esti = LogVrais_Esti(beta)
```

```
% PURPOSE: Estimateur Non Paramétrique de de Log Vraisemblance
%-----
% USAGE: results = LogVrais_Esti(beta)
% where: beta = Valeur des Paramètres
%         gloabl x y : Variables endogene et exogene
%-----
% RETURNS: a structure
%         LogVrais_Esti = Valeur estimé de la logVrais en beta
%-----
% written by:
% C Hurlin - Octobre 2002
```

```
global N x y
```

```
%-----
%--- Estimation de la fonction de lien ---
%-----
index=x*beta; % Creation des Index
h=0.15*(max(index)-min(index)); % Bandwitch Parameter
```

```
for i=1:N
```

```
    % Coefficient de Pondération : Kernel Gaussien
    w=1/sqrt(2*pi)*exp(-0.5*((index-index(i,1))/h).^2);

    % Estimateur fonction de lien
    r(i,1)=sum(w.*y)/sum(w);
```

```
end
```

```
%-----
%--- Estimateur de la Log Vraisemblance ---
%-----
LogVrais_Esti=sum(y.*log(r)+(1-y).*log(1-r));
LogVrais_Esti=-LogVrais_Esti;
```

```

%-----
% ESTIMATION par Semi Paramétrique d'un Modèle Dichotomique (Klein et Sapoz
y 1993)
% Octobre 2003
% C. Hurlin
%-----

clear all
close all
clc

global N x y

%-----
%--- Paramètres de la Simulation ---
%-----
N=100; % Taille d'Echantillon

%-----
%--- Simulation des vecteurs Xi ---
%--- Modèle à Variable Latente ---
%-----

%--- Vraie Valeur des Paramètres ---
beta0=[0.5 0.8]'; % Vecteur de Paramètres
sig=1; % Ecart Type du Choc

%--- Simulation des Séries ---
eps=randn(N,1)*sig; % Distribution Normale
eps=gamm_rnd(N,1,2); % Distribution Gamma
x=[ones(N,1) randn(N,1)*0.5]; % Matrice des Exogènes
ystar=x*beta0+eps; % Construction Variable Latente
y=(ystar>0); % Construction Observable endogène

%-----
%--- Estimation Probit ---
%-----
maxit = 100;
tol = 1e-6;
resup = probit(y,x,maxit,tol);

% Affichage Résultat
disp(' ')
disp(' Modèle Probit : Maximum de Vraisemblance')
disp(' ')

prt(resup);
proba_probit=resup.yhat;

```

```

%-----
%--- Estimation Non Paramétrique ---
%-----
disp(' ')
disp(' ')
disp(' -----')
disp(' ')
disp(' -----')
disp(' ')
disp(' Estimation Semi Paramétrique (Klein et Spady 1993)')
disp(' ')
disp(' -----')
disp(' ')
disp(' -----')
disp(' ')

% Choix des conditions initiales : OLS
condini=resup.beta;           % Guess probit
disp(' Condition Initiale')
disp(condini')

% Options de l'Optimisation
options = OPTIMSET('Display','on');
options.Diagnostics='on';
options.TolFun=1e-20;
options.TolX=1e-20;
options.MaxIter=1000;
options.MaxFunEvals=options.MaxIter;

% Optimisation
betasp=fminsearch('LogVrais_Esti',condini,options);

disp(' ')
disp(' Solution ')
disp(betasp)

%-----
%--- Estimation des Probabilités ---
%-----
index=x*betasp;                % Creation des Index
h=0.15*(max(index)-min(index)); % Bandwitch Parameter

for i=1:N

    % Coefficient de Pondération : Kernel Gaussien
    w=1/sqrt(2*pi)*exp(-0.5*((index-index(i,1))/h).^2);

```

```
% Estimateur fonction de lien
r(i,1)=sum(w.*y)/sum(w);

end

proba_sp=r; % Probas Estimés par KS
proba_probit=resup.yhat; % Probas Estimés par KS

plot([proba_probit proba_sp])
title ('Probabilités Estimées')
legend('Probit','Klein et Spady',0)

[i,j]=max(abs(proba_sp-proba_probit));
disp(' ')
disp( sprintf( 'Exemple Individu %1.0f ',j))
disp(' ')
disp( sprintf( ' Proba Estimée Probit %1.2f ',proba_probit(j)))
disp(' ')
disp( sprintf( ' Proba Estimée Klein Spady %1.2f ',proba_sp(j)))
disp(' ')
disp( sprintf( ' Realisation %1.0f ',y(j)))
```

```
function score = score(beta)
```

```
% PURPOSE: Fonction de Score Maximum : Estimation Non Paramétrique
```

```
%-----
```

```
% USAGE: results = Chapitre1_Score(beta)
```

```
% where: beta = Valeur des Paramètres du Score
```

```
%         gloabl  x y : Variables endogene et exogene
```

```
%-----
```

```
% RETURNS: a structure
```

```
%         score    = Valeur du Score
```

```
%-----
```

```
% written by:
```

```
% C Hurlin - Septembre 2002
```

```
global N x y
```

```
score = (1/N)*sum((y==1).*(x*beta>0)+(y==0).*(x*beta<0));
```

```
score=-score;
```

```
% Comparaison Estimation par MV et Estimation Non Paramétrique
```

```
%-----
```

```
% ESTIMATION par Score Maximum et MV
```

```
% Septembre 2003
```

```
% C. Hurlin
```

```
%-----
```

```
clear all
```

```
clc
```

```
global N x y
```

```
%-----
```

```
%--- Paramètres de la Simulation ---
```

```
%-----
```

```
N=1000; % Taille d'Echantillon
```

```
%-----
```

```
%--- Simulation des vecteurs Xi ---
```

```
%---- Modèle à Variable Latente ---
```

```
%-----
```

```
%--- Vraie Valeur des Paramètres ---
```

```
beta0=[0.5 0.8]'; % Vecteur de Paramètres
```

```
sig=1; % Ecart Type du Choc
```

```
%--- Simulation des Séries ---
```

```
eps=randn(N,1)*sig; % Distribution Normale
```

```
x=[ones(N,1) randn(N,1)*0.5]; % Matrice des Exogènes
```

```
ystar=x*beta0+eps; % Construction Variable Latente
```

```
y=(ystar>0); % Construction Observable endogène
```

```
%-----
```

```
%--- Estimation Probit ---
```

```
%-----
```

```
maxit = 100;
```

```
tol = 1e-6;
```

```
resup = probit(y,x,maxit,tol);
```

```
% Affichage Résultat
```

```
disp(' ')
```

```
disp(' Modèle Probit : Maximum de Vraisemblance')
```

```
disp(' ')
```

```
prt(resup);
```

```
proba_probit=resup.yhat;
```

```
%-----
```

```
%--- Estimation Non Paramétrique ---
```

```
%-----
```

```
disp(' -----')
disp(' -----')
disp(' ')
disp(' Méthode du Score')
disp(' ')

% Choix des conditions initiales : OLS
%condini=x\y;           % Guess ols
condini=resup.beta;    % Guess probit
disp(sprintf(' Condition Initiale = %1.2f',condini))

% Options de l'Optimisation
options = OPTIMSET('Display','on');
options.Diagnostics='on';
options.TolFun=1e-20;
options.TolX=1e-20;
options.MaxIter=10000;
options.MaxFunEvals=options.MaxIter;

% Optimisation du Score
betascore=fminsearch('Score',condini,options);

disp(' ')
disp(sprintf(' Solution = %1.3f',betascore))

disp(' -----')
disp(' -----')
disp(' ')
disp(' Comparaison des Probabilités Estimées et Prévisions : individus de
1 à 10 ')

prevision_probit=(resup.yhat>0.5);
prevision_score=(x*betascore>0);

prevision=[prevision_probit prevision_score];

disp(prevision(1:10,:))

disp(sprintf(' Nombre de Previsions Différentes = %1.0f',sum(prevision_pro
bit~=prevision_score)))
```