Estimación Probit en gretl

Microeconomía Cuantitativa

R. Mora

Departmento de Economía
Universidad Carlos III de Madrid

Esquema

1. Introducción
2. Probit en gretl
3. El modelo Logit

El modelo Probit y su estimación MV

El modelo Probit

- $U_m = \beta_m x_m + \varepsilon_m$
- $U_h = \beta_h x_h + \varepsilon_h$
- $\varepsilon_h, \varepsilon_m \sim N(0, \Sigma)$ tal que $\varepsilon \sim N(0,1)$
- $Pr(y=1) = \Phi(\beta x)$ donde Φ es la fda de la normal estándar

$$\hat{\beta}^{ML} = \arg\max \sum_i \{ y_i \log(\Phi(\beta x_i)) + (1 - y_i) \log(1 - \Phi(\beta x_i)) \}$$

- en `gretl`, se utiliza un algoritmo quasi-Newton (el algoritmo BFGS)

Comandos básicos en `gretl` para la estimación Probit

- `probit`: computa Máxima Verosimilitud para el modelo Probit
- `omit/add`: constrastes de exclusión
- `$yhat`: estima la probabilidad del evento
- `$lnl`: proporciona la log-verosimilitud del último modelo estimado
- `logit`: computa Máxima Verosimilitud para el modelo alternativo logit

- in esta Sesión, aprenderemos a usar `probit`, `$yhat` y `logit`

probit *depvar indvars* --robust --verbose --p-values

- *depvar* debe ser binaria $\{0,1\}$ (de lo contrario, o aparece un error o se estima otro modelo)
- las pendientes (“`slopes`”) se computan en las medias muestrales
- las desviaciones típicas se obtienen con la inversa del hessiano
- el output muestra el estadístico χ^2_q de la nula de que todos los coeficientes son cero salvo la constante
- opciones:
  1. `--robust`: matriz de covarianzas robusta a error de especificación
  2. `--p-values`: muestra los p-valores en vez de las “`slopes`”
  3. `--verbose`: muestra información sobre todas las iteraciones

Ejemplo: Datos Simulados

El Modelo Probit

- $U_m = 0.3 + 0.05 * educ + 0.5 * kids + \varepsilon_m$
- $U_h = 0.8 - 0.02 * educ + 2 * kids + \varepsilon_h$
- $\varepsilon_h, \varepsilon_m \sim N(0, \Sigma)$ such that $\varepsilon \sim N(0,1)$

- la educación proporciona utilidad si trabajas y desutilidad si no
- tener un niño proporciona más utilidad si no trabajas
- $\beta x = -0.5 + 0.07 * educ - 1.5 * kids$

La salida del comando probit

probit *work const educ kids*

```
Convergence achieved after 6 iterations

Model 1: Probit, using observations 1-5000
Dependent variable: work

              coefficient   std. error   t-ratio     slope
  ---------------------------------------------------------
  const       -0.434462     0.0812490     -5.347
  educ         0.0659247    0.00576068    11.44      0.0240325
  kids        -1.47598      0.0407604    -36.21     -0.521270

Mean dependent var   0.366800   S.D. dependent var   0.364545
McFadden R-squared   0.233290   Adjusted R-squared   0.232378
Log-likelihood      -2519.525   Akaike criterion     5045.049
Schwarz criterion    5064.601   Hannan-Quinn         5051.902

Number of cases 'correctly predicted' = 3859 (77.2%)
f(beta'x) at mean of independent vars = 0.365
Likelihood ratio test: Chi-square(2) = 1533.26 [0.0000]

           Predicted
               0      1
  Actual 0  2495    671
         1   470   1364
```

Prediciendo probabilidades

Computando $\hat{\Pr}(y_i = 1 | x_i)$

genr p_hat =$yhat

- para cada observación, si $\hat{\Pr}(y_i = 1 | x_i) > 0.5$ entonces $\hat{y}_i = 1$
- "`percent correctly predicted`" es el % para el que $\hat{y}_i$ iguala y_i
- es posible obtener altos % en modelos inservibles
  - supón que $\Pr(y_i = 0) = 0.9$
  - predecir siempre $\hat{y}_i = 0$ conllevará $\approx 90\%$ de predicciones correctas!

Entendiendo los coeficientes y las pendientes

- la columna "`coefficient`" incluye las estimaciones ML $\hat{\beta}^{ML}$
- a diferencia del modelo lineal, en el modelo Probit los coeficientes no capturan el efecto marginal sobre el output de una variación del control
  - si el control x_j es continuo, $\frac{\partial Pr(y=1)}{\partial x_j} = \phi(\beta x)\beta_j$
  - si el control x_j es discreto, $\Delta Pr(work = 1) = \Phi(\beta x_1) - \Phi(\beta x_0)$
- puesto que el modelo no es lineal, los efectos marginales dependen de los valores de otros controles
- la columna "`slopes`" incluye los efectos marginales computados en las medias muestrales de los controles

Efectos marginales individuales: cambios discretos

queremos estimar el cambio de la probabilidad cuando x pasa de valer x_0 a x_1

Cambio discreto

- tras estimar el modelo, guarda el vector $\hat{\beta}^{ML}$ en un vector
- genera una matriz con los controles bajo el escenario 0, x_0, y otra con los controles bajo el escenario 1, x_1
- computa las funciones índices $\hat{\beta}^{ML} x_0$ y $\hat{\beta}^{ML} x_1$
- genera los efectos individuales

$$\Phi\left(\hat{\beta}^{ML} x_1\right) - \Phi\left(\hat{\beta}^{ML} x_0\right)$$

Ejemplo: el efecto de tener un niño

```
# marginal effects of having a kid
 genr beta=$coeff
 series kids0=0
 matrix x0={const,educ,kids0}
 series kids1=1
 matrix x1={const,educ,kids1}
 series x1b = x1*beta
 series x0b = x0*beta
 series Mg_kid = cdf(N,x1b)-cdf(N,x0b)
 summary Mg_kid --by=educ --simple
 summary Mg_kid --simple
```

summary *Mg_kid* --by=*educ* --simple

```
educ = 8  (n = 759) :    -0.45370
educ = 12 (n = 2279):    -0.50782
educ = 16 (n = 1499):    -0.53638
educ = 21 (n = 463) :    -0.52950
```

- aunque la función índice es lineal, el efecto de tener un niño cambia con la educación
- mayor educación hace que los individuos tiendan a tener índices βx más cercanos a 0.5 (la pendiente probit es máxima en 0.5)
- tal y como está, el modelo no hace que tener un niño afecte más a las mujeres con menor educación
- ¿cómo podríamos crear tal efecto?

Efectos marginales individuales: cambios infinitesimales

Aproximación por cálculo infinitesimal

- guarda las estimaciones $\hat{\beta}^{ML}$ en un vector
- genera una matriz con los valores de todos los controles, x
- computa la función índice $\hat{\beta}^{ML}x$
- genera la aproximación: $\phi\left(\hat{\beta}^{ML}x\right)\hat{\beta}_j^{ML}$

Ejemplo de cálculo infinitesimal

```
genr beta=$coeff
matrix x={const,educ,kids}
series xb=x*beta
genr meanXb=mean(xb)
series Mg_educ_slope=pdf(N,meanXb)*$coeff(educ)       # this is the slope in gretl output
series Mg_educ_cal=pdf(N,xb)*$coeff(educ)             # this is the individual's marginal effect
summary Mg_educ_slope Mg_educ_cal --by=kids --simple
```

```
kids = 0 (n = 2035):
                          Mean        Minimum        Maximum       Std. Dev.
Mg_educ_slope         0.025214       0.025214       0.025214          0.0000
Mg_educ_cal           0.024004       0.016797       0.027284       0.0028914

kids = 1 (n = 2965):
                          Mean        Minimum        Maximum       Std. Dev.
Mg_educ_slope         0.025214       0.025214       0.025214          0.0000
Mg_educ_cal           0.016362       0.010516       0.024298       0.0038762
```

El supuesto Logit

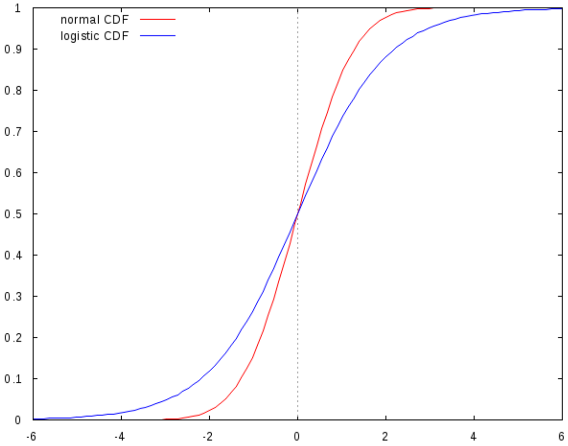
- $U_m = \beta_m^0 + \beta_m^e educ + \beta_m^k kids + \varepsilon_m$
- $U_h = \beta_h^0 + \beta_h^e educ + \beta_e^k kids + \varepsilon_h$

Supuesto Logit: $\varepsilon_h - \varepsilon_m = \varepsilon \sim$Logistic

- $Pr\left(work = 1\right) = \frac{exp(\beta_0 + \beta_e educ + \beta_k kids)}{1 + exp(\beta_0 + \beta_e educ + \beta_k kids)}$
- Esta probabilidad es muy fácil de computar

Logit vs. Probit

Las colas del Logit son más gruesas



Las estimaciones de Logit & Probit no son comparables directamente...

probit *const educ kids*

```
Convergence achieved after 6 iterations

Model 1: Probit, using observations 1-5000
Dependent variable: work

             coefficient   std. error   t-ratio     slope
  -----------------------------------------------------------
  const      -0.434462     0.0812490     -5.347
  educ        0.0659247    0.00576068    11.44     0.0240325
  kids       -1.47598      0.0407604    -36.21    -0.521270

Mean dependent var   0.366800   S.D. dependent var   0.364545
McFadden R-squared   0.233290   Adjusted R-squared   0.232378
Log-likelihood      -2519.525   Akaike criterion     5045.049
Schwarz criterion    5064.601   Hannan-Quinn         5051.902

Number of cases 'correctly predicted' = 3859 (77.2%)
f(beta'x) at mean of independent vars = 0.365
Likelihood ratio test: Chi-square(2) = 1533.26 [0.0000]

           Predicted
              0      1
  Actual 0  2495    671
         1   470   1364
```

logit *const educ kids*

```
Convergence achieved after 5 iterations

Model 3: Logit, using observations 1-5000
Dependent variable: work

             coefficient   std. error   t-ratio     slope
  -----------------------------------------------------------
  const      -0.785736     0.137875      -5.699
  educ        0.113003     0.00995293    11.35     0.0248565
  kids       -2.45293      0.0712432    -34.43    -0.523436

Mean dependent var   0.366800   S.D. dependent var   0.219963
McFadden R-squared   0.233181   Adjusted R-squared   0.232268
Log-likelihood      -2519.884   Akaike criterion     5045.768
Schwarz criterion    5065.319   Hannan-Quinn         5052.620

Number of cases 'correctly predicted' = 3859 (77.2%)
f(beta'x) at mean of independent vars = 0.220
Likelihood ratio test: Chi-square(2) = 1532.54 [0.0000]

           Predicted
              0      1
  Actual 0  2495    671
         1   470   1364
```

pero los efectos marginales, las columnas “`slope`”, lo son

Resumen

- gretl permite la estimación del modelo Probit de la utilidad aleatoria por MV
- no todos los parámetros del modelo de utilidad aleatoria pueden ser estimados
- el modelo Probit identifica cómo cada control afecta la probabilidad de $y = 1$
- la estimación MV del modelo Logit también es posible en gretl