Probit Estimation in gretl

Quantitative Microeconomics

R. Mora

Department of Economics
Universidad Carlos III de Madrid

Outline

The Probit Model and ML Estimation

The Probit Model

- $U_m = \beta_m x_m + \varepsilon_m$
- $U_h = \beta_h x_h + \varepsilon_h$
- $\varepsilon_h, \varepsilon_m \sim N(0, \Sigma)$ such that $\varepsilon \sim N(0,1)$
- $Pr(y=1) = \Phi(\beta x)$ where Φ is the cdf of the standard normal

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

- in gretl, a quasi-Newton algorithm is used (the BFGS algorithm)

Basic Commands in gretl for Probit Estimation

- probit: computes Maximum Likelihood probit estimation
- omit/add: tests joint significance
- $yhat: returns probability estimates
- $lnl: returns the log-likelihood for the last estimated model
- logit: computes Maximum Likelihood logit estimation

- in this Session, we are going to learn how to use probit, $yhat, and logit

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

- *depvar* must be binary $\{0,1\}$ (otherwise a different model is estimated or an error message is given)
- slopes are computed at the means
- by default, standard errors are computed using the negative inverse of the Hessian
- output shows χ^2_q statistic test for null that all slopes are zero
- options:
  1. `--robust`: covariance matrix robust to model misspecification
  2. `--p-values`: shows p-values instead of slope estimates
  3. `--verbose`: shows information from all numerical iterations

Example: Simulated Data

The Probit Model

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

- education brings utility if you work, dissutility if you don't
- having a kid brings more utility if you don't work
- $\beta x = -0.5 + 0.07 * educ - 1.5 * kids$

probit Output

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
```

Predicting the Probabilities

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

```
genr p_hat=$yhat
```

- for each observation, if $\hat{\Pr}(y_i = 1 | x_i) > 0.5$ then $\hat{y}_i = 1$
- the percent correctly predicted is the % for which $\hat{y}_i$ matches y_i
- it is possible to get high percentages correctly predicted in useless models
  - suppose that $\Pr(y_i = 0) = 0.9$
  - always predicting $\hat{y}_i = 0$ will lead to 90% correctly predicted!

Understanding the Coefficients and the Slopes

- the column "`coefficient`" refers to the ML estimates $\hat{\beta}^{ML}$
- in contrast to the linear model, in the probit model the coefficients do not capture the marginal effect on output when a control changes
  - if control x_j is continuous, $\frac{\partial Pr(y=1)}{\partial x_j} = \phi(\beta x)\beta_j$
  - if control x_j is discrete, $\Delta Pr(work = 1) = \Phi(\beta x_1) - \Phi(\beta x_0)$
- since the model is non-linear, marginal effects depend on the values of the other controls
- the column "`slopes`" refers to marginal effects computed at the sample average values for all controls

Individual Marginal Effects: Discrete Change

we want to estimate the change in probability when x changes from x_0 to x_1

Discrete change

- after estimation of the model, store estimated coefficients $\hat{\beta}^{ML}$ in a vector
- generate a matrix with the controls under scenario 0, x_0, and another one with the controls under scenario 1, x_1
- predict index functions $\hat{\beta}^{ML}x_0$ and $\hat{\beta}^{ML}x_1$
- generate the individual marginal effects

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

Example: The Effect of Having A Kid

```
# 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
```

- although the index function is linear, the effect of having a kid changes with education
- higher education makes individuals more likely to have indexes βx closer to 0.5 (the probit slope is largest at 0.5)
- the model as it stands does not make the "kid" effect smaller with higher education
- how would you create that effect?

Individual Marginal Effects: Infinitessimal Change

Calculus approximation

- store estimated coefficients $\hat{\beta}^{ML}$ in a vector
- generate a matrix with the values for all controls, x
- predict the index function $\hat{\beta}^{ML}x$
- generate the calculus approximation: $\phi\left(\hat{\beta}^{ML}x\right)\hat{\beta}_j^{ML}$

Example of Calculus Approximation

```
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
```

The Logit Assumption

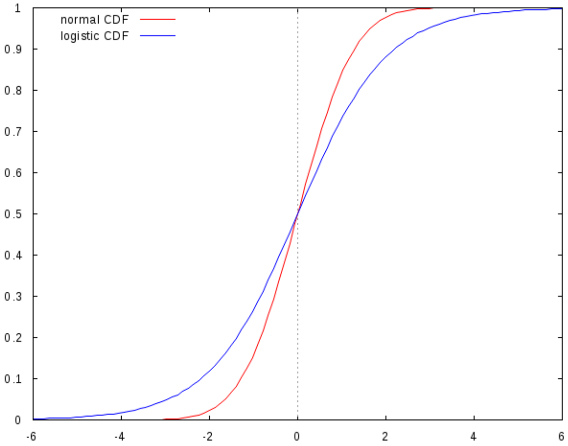
- $U_m = \beta_m^0 + \beta_m^e educ + \beta_m^k kids + \varepsilon_m$
- $U_h = \beta_h^0 + \beta_h^e educ + \beta_e^k kids + \varepsilon_h$

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

- $Pr\left(work = 1\right) = \frac{exp(\beta_0 + \beta_e educ + \beta_k kids)}{1 + exp(\beta_0 + \beta_e educ + \beta_k kids)}$
- Easy computation!

Logit vs. Probit

Tails are thicker in the logit



Logit & Probit Beta Estimates are not Directly Comparable...

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
```

but marginal effects, the “slope” columns, are

Summary

- gretl allows for probit estimation of the random utility model by ML
- not all parameters of the RUM can be estimated
- the Probit model identifies how each control affects the probability of $y = 1$
- logit estimation estimation of random utility model by ML can also be conducted in gretl