The Probit model

August 8, 2008

The problem

Suppose we are faced with a situation where the variable of interest, that we choose to explain, takes values of 0 or 1. This is called a discrete variable, a categorical variable, or a 'factor'. We seek to write a model where a set of explanatory variables determine this outcome.

Examples of this situation include:

- The outcome of an election (did the incumbent win or not?)
- Whether a firm has built a factory outside its home country or not
- Whether a country has further liberalised the capital account in a given year or not.

In each of these cases, we seek to write a model where the outcome is explained using a set of explanatory variables. Ordinary OLS is not appropriate for this because in OLS, the variable that we seek to explain must have real values and can run from ∞ to ∞. There is a theorem which teaches us that if OLS is inappropriately applied in this situation, the estimates from this 'linear probability model' are inconsistent.

The model

We assume there is a latent, or unobserved, variable y^* which is generated from a familiar looking model:

$$y^* = \beta' x + e$$

where β is a K-vector of parameters, x is a vector of explanatory variables and $e \sim N(0,1)$ is a random shock. We observe $y = 1$ if $y^* > 0$ and $y = 0$ otherwise.

Note that in the model, the standard error of e is 1. If we sought to write $e \sim N(0, \sigma^2)$, this σ is not identified. To see this, divide both left and right hand sides by σ.

Estimation

It is easy to show that $\Pr(y = 1) = \Phi(\beta' x)$. This gives us the likelihood for both cases $y = 0$ and $y = 1$. Assuming the observations are i.i.d. it is easy to construct the sample log likelihood. This can be maximised using standard nonlinear maximisation algorithms. The standard MLE inference procedures give us the variance-covariance matrix of $\hat{\beta}$.

Monte carlo example in R

Let us start by simulating a dataset containing x_1 and x_2, and a true model where $y^* = 7 + 3x_1 - 4x_2 + e$. Thus, the true $\beta = (7, 3, -4)$.

```
R version 2.7.1 (2008-06-23)

> # Simulate from probit model --
> simulate <- function(N) {
   x1 <- 2*runif(N)
   x2 <- 5*runif(N)
   ystar <- 7 + 3*x1 - 4*x2 + rnorm(N)
   y <- as.numeric(ystar>0)
   print(table(y))
   data.frame(y, x1, x2, ystar)
 }
```

The function `simulate(N)` will give us a data frame with N observations from this model.

Let's go on to estimation. We'll use the glm() function in R where probit is a special case of Generalised Linear Models.

```
> # A simple example of estimation --
> D <- simulate(200)
y
  0   1
 97 103
> m <- glm(y~x1+x2, family=binomial(probit), data=D)
> summary(m)

Call:
glm(formula = y ~ x1 + x2, family = binomial(probit), data = D)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-2.125e+00  -7.332e-04   2.107e-08   7.812e-04   2.326e+00

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   8.0810     1.6723   4.832 1.35e-06 ***
x1            3.1877     0.6948   4.588 4.47e-06 ***
x2           -4.4544     0.8488  -5.248 1.54e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 277.079  on 199  degrees of freedom
Residual deviance:  40.845  on 197  degrees of freedom
AIC: 46.845

Number of Fisher Scoring iterations: 10
```

For this one dataset of 200 observations, we got back a $\hat{\beta} = (8.08, 3.19, -4.45)$ which is quite close to the true β.

Let us look at how well this model predicts in-sample:

```
> # Predictions using the model --
> predictions <- predict(m)
> summary(predictions)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-13.0800  -4.1720   0.1780   0.3705   5.1250  12.7400
> yhat <- ifelse(predictions>0, 1, 0)
> table(yhat, D$y)
yhat  0  1
   0 93  5
   1  4 98
```

It does pretty well. For 93 cases, the truth is 0 and the predicted value is 0. Similarly, for 98 cases, the truth is 1 and the predicted value is 1. There are only 9 cases which are misclassified.

Do we have consistency?

```
> # For very large N do we recover the true parameters?
> D <- simulate(10000)
y
   0    1
4996 5004
> m <- glm(y~x1+x2, family=binomial(probit), data=D)
> coef(m)
(Intercept)          x1          x2
   6.878840    2.983776   -3.945550
```

Compared with a true $\beta = (7, 3, -4)$, we have recovered a $\hat{\beta} = (6.878, 2.983, -3.945)$.

Further reading

For an example of an applied paper that utilises the probit model, see: *The economic determinants of the choice of an exchange rate regime: A probit analysis* by J. M. Rizzo, *Economics Letters*, 1998,
http://linkinghub.elsevier.com/retrieve/pii/S0165176598000561

History

The probit model was first proposed by Chester Ittner Bliss in 1935. Estimation of the model only became practical in the 1970s with the availability of mainframe computers which could solve nonlinear maximisation problems. It has become the workhorse of analysing discrete choice problems.

Immediate extensions

Some discrete variables take on more than one discrete outcome. Sometimes there is a clear ordering, e.g. if we define 0 for no education, 1 for school education and 2 for college or beyond, we have a ordered factor with values 0,1,2. The ordered probit model is a natural extension of the probit model where instead of defining 0 or 1 based on whether $y^* > 0$ or not, we look at where y^* falls with respect to a vector of cutoffs τ.

When a discrete variable cannot be ordered, the multinomial probit is useful.

Alternative distributions can be considered. In the discrete $0/1$ case, this leads to the logit model instead of the probit model.

Generalised linear models (GLMs) are a powerful framework which encompasses a wide variety of these models.