Discrete Choice Modeling
William Greene
Stern School of Business, New York University

Lab Session 1 Assignment 1A
Binary Choice Modeling

This exercise will involve estimating and analyzing binary choice models. We will analyze the panel probit, manufacturing innovation data. The data set is PanelProbit.lpj. Use File → Open Project … to open PanelProbit.lpj. These data are a panel. The data set appears as follows:

```
+-----------------------------------------------------------------------------+
|****************************************************************             |
|Panel probit data: Stacked                                                   |
|N = 1270, T = 5                                                              |
|EMPLP  = Employment                                                          |
|IM = Industry employment                                                     |
|IP = dependent variable, innovation, binary                                  |
|IMUM = imports share                                                         |
|FDIUM = FDI share                                                            |
|SP = relative size                                                           |
|PROD = productivity                                                          |
|SALES = sales                                                                |
|LOGSALES = log sales                                                         |
|RAWMTL, INVGOOD, CONSGOOD, FOOD = sector dummies                             |
|T = period, T1,T2,T3,T4,T5 = period dummy variables                          |
|FIRM = firm ID                                                               |
|Authors Model = (one,logsales,sp,imum,fdium,prod,rawmtl,invgood)             |
|****************************************************************             |
|Panel Probit Data -  Wide form                                               |
|For observations with T=1 (ignore the others)                                |
|IP84...IP88 = 5 years of IP                                                  |
|EMPLP84...EMPLP88                                                            |
|IM84...IM88                                                                  |
|IMUM84...IMUM88                                                              |
|FDIUM84...FDIUM88                                                            |
|PROD84...PROD88                                                              |
|SALES84...SALES88                                                            |
|LSALES84...LSALES88                                                          |
+-----------------------------------------------------------------------------+
```

1. Different Functional forms.

As we saw in class, the different distributions chosen for the binary choice model each imply a scaling of the coefficients. Superficially, it appears that the model results depend heavily on the distribution. But, this is illusory. The differences essentially disppear when we examine the partial effects rather than the raw coefficients. The following will illustrate this effect for three specific functional forms. The commands compute and assemble the results in tables that enable convenient viewing.

```
Probit       ; Lhs = IP ; Rhs = x ; Table = Probit  $
Partials     ; effects: x ; summary(table=ProbitME) ; means $
Logit        ; Lhs = IP ; Rhs = x ; Table = Logit   $
Partials     ; effects: x ; summary(table=LogitME)  ; means $
Arctan       ; Lhs = IP ; Rhs = x ; Table = Arctan    $
Partials     ; effects: x ; summary(table=ArctanME) ; means $
Maketable    ; Probit,Logit,Arctan$
Maketable    ; ProbitME,LogitME,ArctanME $
```

2. The Linear Probability Model

Some recent applications have used linear regression to fit a 'linear probability' rather than employ the usual probit model. What does least squares do in a binary choice setting? As might be expected from the previous exercise, the coefficients one obtains are very different. Are the results? The following compares the results of the linear probability model to those of a logit model, both in terms of the coefficients and the partial effects. The results suggest what is actually happening when one uses a linear probability model. The coefficients are approximating the partial effects (at the means of the data) of the appropriate nonlinear binary choice model.

```
Regress     ; Lhs = IP ; Rhs = x ; Table = LinearPM $
Partials    ; effects: x ; summary(table=linearME) $
Maketable   ; Logit,LinearPM $
Maketable   ; LogitME,LinearME $
```

The success of the linear probability at mimicing the probit model is mixed. Notice the good result for IMUM and FDIUM, but the less favorable results for IM, SP, PROD and LOGSALES.

3. A Robust Covariance Matrix.

It is now common to compute a 'robust' sandwich type of estimator when fitting a binary choice model. As we discussed in class, there is not much in the way of failures of the model assumption to which the MLE could be robust. Nonetheless, it might be of interest how much difference it makes. The robust estimator is $\mathbf{H}^{-1}(\mathbf{G}'\mathbf{G})\mathbf{H}^{-1}$, where $\mathbf{H}$ is the negative of the Hessian of the log likelihood and G is the n×K matrix of first derivatives, by observation, of the log densities. The following computes the conventional estimator, $\mathbf{H}^{-1}$ and the robust estimator. We then report the two sets of results side by side.

```
Probit      ; Lhs = IP ; Rhs = X ; Table = standard $
Probit      ; Lhs = IP  ; Rhs = X ;  RobustVC ; Table = Robust $
Probit      ; Lhs = IP  ; Rhs = X ; Cluster = Firm ; Table = Cluster $
Maketable   ; Standard, Robust, Cluster $
```

With one notable exception, the so-called robust estimator doesn't matter much. But, the clustering seems to make a large difference. Again, this is to be expected.

4. Creating a Plot of Probabilities.

Once estimation is completed, there are a variety of useful post estimation computations that can be carried out with the estimated model. To begin, it is useful to display the predicted probabilities produced by the model. The following estimates a probit model for innovation, then simulates the probabilities over the range of logSales. The plot is generated by dividing the range into 20 parts from the sample minimum of logSales to the maximum. A listing of the probabilities averaged over the sample with all other variables taking their observed values is shown, followed by a plot with a confidence interval around the prediction.

```
Probit      ; if[t=1] ; Lhs = IP ; Rhs = one,IMUM,FDIUM,SP,logsales $
Calc        ; low = .5*Min(LogSales)  ; high = 1.5*Max(LogSales)
            ; incrmnt = .05*(high-low) $
Simulate    ; scenario: logsales & logsales = Low(incrmnt)high
            ; plot(ci)
            ; title=Simulation of Innovation Probabilities vs. Log Sales$
```

5. Fit Measures

The binary choice models are not fit by least squares, and there is no R squared-like statistic to measure the correlation between the predictions of the model and the observed data. Many ad hoc measures have been proposed. The most widely known is McFadden's pseudo R squared, which as discussed in class, does not actually measure anything like the fit of the model to the data. We examined a number of others in class. The following fits a probit model and stores the predicted probabilities. It then computes predictions by the rule 'Predict y = 1 if fitted probability is greater than T*.' The routine lets you choose T*. The usual choice is T* = .5, however, for these data, the best choice – the choice that produces the most frequent match (zero or one) between actual and prediction – is less than .5. Some authors label this statistic the 'count R squared.' Try different values in the second execute command to find that value. Note that this strategy, in general, is not optimal because the MLE is not chosen to maximize the number of correct predictions. Manski's maximum score estimator does just that. The last command computes the Mscore estimator and displays the fit obtained.

```
Probit          ; Lhs = IP ; Rhs = X ; Summarize ; Prob = Pfit $
Proc = Fit(limit) $
Create          ; ipfit = Pfit > limit ; Correct = (IPfit = ip) $
Crosstab        ; Lhs = IP ; Rhs = IPFIT $
Calc            ; List ; CountRsq = Sum(correct) / N $
EndProc $
Exec            ; Proc = fit(.4) $
Calc            ; MeanIP = xbr(ip) $
Exec            ; Proc = fit(MeanIP) $
Exec            ; Proc = fit(.42)$  Try different values.  Which is best?
MScore          ; if[t=1] ; Lhs = IP ; Rhs = X $
```

6. Partial Effects for a Quadratic and for Interaction Terms

Marginal effects in the binary choice models are complicated functions of the parameters and the data. They are more so when the index function contains complex functions of the data. Suppose, for example,

$$P = \Phi(\boldsymbol{\beta}'\mathbf{x} + \alpha_0\text{logSales} + \alpha_1\text{logSales}^2).$$

The marginal effect of logSales, which is the effect on the probability of a one percent change in sales is

$$\partial P/\partial \text{logSales} = \phi(\boldsymbol{\beta}'\mathbf{x} + \alpha_0\text{logSales} + \alpha_1\text{logSales}^2) \times (\alpha_0 + 2\alpha_1\text{logSales})$$

Computing these properly is a longstanding, widely discussed issue in modern software. The problem, in general, is in obtaining the right single effect for logSales rather than separate effects for the two parts, neither of which give the right answer. Recent versions of Stata (with 'Margins') and NLOGIT (with PARTIALS and SIMULATE) have automated the computation of these types of effects. The following does several computations around this formulation. The probit model contains the indicated quadratic term in logSales. The first command computes the average partial effects for logSales and fdium. The second computes the average partial effect for logSales while varying fdium from .05 to 1.0 in steps of .05, and plots the results. This calculation is done using the delta method. The next Partials command does the same thing, but uses the method of Krinsky and Robb. Since K&R involves a large amount of computation, we have speeded it up by using only the observations with T = 1, which is the first of 5 years of the data. The Wald command shows how to compute the partial effects another way, by actually programming the function. In this example, Wald is more complicated than necessary. In other applications, it might be preferred.

```
Namelist        ; X = One,IMUM,FDIUM,SP $
Probit          ; Lhs = ip ; rhs = x,logsales,logsales^2 $
Partials        ; effects: logsales / fdium $
Partials        ; effects: logsales & fdium = .05(.05)1 ; plot(ci) $
Partials        ; if[t = 1] ; effects: logsales & fdium = .05(.05)1 ; plot(ci)
                ; k&r ; pts = 50 $ Add this to the command to use K&R
? Use Wald instruction for Delta method or K&R. (Add ;K&R)
Namelist        ; FullX = x,logsales,logsales^2
Wald            ; Start  = b ; Var = Varb
                ; Labels = beta0,beta1,beta2,beta3,a0,a1
                ; Fn1 = ME_logS   = n01(beta0'fullX)*(a0+2*a1*Logsales)
                ; Fn2 = ME_fdium = n01(beta0'fullX)*beta2
                ; Average $
```

Computing partial effects for models with interaction terms presents the same challenges as nonlinearities, but yet more complex. The model below is

$$P = \Phi(\beta_1 + \beta_2\, imum + \beta_3 fdium + \beta_4 sp + \beta_5 sp{\times}imum + \beta_6\, sp{\times}fdium \\ + \beta_7\, logSales + \beta_8\, logSales^2).$$

It contains the same nonlinearities as the previous model, plus the interaction terms in sp with the other two variables. The following estimates the probit model then computes partial effects for sp, evaluated at the sample range of values. All terms are accounted for.

```
Namelist        ; x=one,imum,fdium,sp,sp*imum,sp*fdium,logsales,logsales^2 $
Probit          ; Lhs = ip ; rhs = x $
Partials        ; effects: sp & sp = .05(.05)1 ; plot (ci) $
```

7. A Group of Dummy Variables for a Set of Categories

The data set also includes a set of sector dummy variables for four sectors. It might be interesting to examine the different results for the four sectors. The Namelist instruction defines the data matrix Sector which contains all four dummy variables. One of them must be dropped in estimation. The probit command contains 'sector.' – note the ending dot. This instructs NLOGIT to drop the last category. We will want all four categories for the next instruction. In the results of the probit estimation the coefficients on the first three dummy variables relate to the change in the predicted probability related to the omitted category, in this case, food. We might be interested in different transitions. For example, an interestng margin might be a comparison of the raw materials sector to the consumer goods sector. The Partials command requests a transition matrix that computes these transition probabilities.

```
? Group of dummy variables
Namelist        ; Sector = rawmtl,invgood,consgood,food $
Probit          ; Lhs = ip ; Rhs = x,sector. $
Partials        ; Effects : sector ; transition $
```

8. Testing for Structural Change.

A common test is for homogeneity of the parameter vector across different groups. For example, in our application here, it might be interesting to test whether underlying structural of the model has changed over the five year period of the data. Consider the structure

$$P_{it} = F(\boldsymbol{\beta}_t'\mathbf{x}_{it}),\ i = 1,\ldots,1270,\ t = 1,\ldots,5 \text{ (1993 to 1997)}$$

which allows for different coefficient vectors in each year. We are interested in testing the hypothesis

$$H_0 : \boldsymbol{\beta}_1 = \boldsymbol{\beta}_2 = \boldsymbol{\beta}_3 = \boldsymbol{\beta}_4 = \boldsymbol{\beta}_5$$
$$H_1: \text{ not } H_0.$$

In a linear regression context, this would be a 'Chow' test and would be tested with an F test. Since this is not a linear regression model, we can't use the F test here. The easiest way to do this test is with a likelihood ratio test. The strategy is to fit the restricted model (pool the 5 years of data) and the unrestricted model (estimate the model separately for each year), and compare the log likelihoods. The log likelihood for the unrestricted model is the sum of the five years. Here is how you can automate this computation. The last part of the last CALC displays the 95% critical value from the chi-squared table. There are two ways to proceed. The first set of commands builds the procedure from first principles. The second uses a built in procedure. Carry out the test. What do you conclude? Should the null hypothesis be rejected? Repeat the test using a logit model instead of a probit model. Does the conclusion change? Try the exercise again while adding the sector dummy variables to the model. To do these, it is only necessary to change the model name from PROBIT to LOGIT, or the NAMELIST command by adding variables to it.

```
Namelist      ; x=one,imum,fdium,sp,sp*imum,sp*fdium,logsales,logsales^2 $
Probit        ; Lhs = IP ; Rhs = X ; quietly $  (Suppress the model results)
Calc          ; Logl0 = Logl ; Logl1 = 0 ; i = 0 $
Procedure
Probit        ; If[t = i] ; Quietly ; Lhs = IP ; Rhs = X $ (Suppress model results)
Calc          ; Logl1 = Logl1 + Logl $
EndProc $
Execute       ; i = 1,5 $ (This suppresses the individual year results.)
Calc          ; List ; Chisq = 2*(Logl1 - Logl0) ; Df = 4*Col(X) ; Ctb(.95,df) $

? Use internal automated procedure
Probit        ; for[ (test) t ] ; lhs=ip;rhs=x;quietly$
```

9. Hypothesis Tests:

This exercise will illustrate the three methods of carrying out hypothesis tests. Two tests are carried out. All of the procedures save for the last carry out the test of whether the sector dummy variables should be included in the index function in the probit model. In the last test, The model is

$$y_i^* = \boldsymbol{\beta}'\mathbf{x}_i + \varepsilon_i$$
$$\varepsilon \sim N[0,\sigma_i^2],\ \ \sigma_i = \exp(\boldsymbol{\gamma}'\mathbf{z}_i).$$
$$y_i = 1(y_i^* > 0]$$

and the test of whether $\gamma = 0$ is carried out using an LM test. The (small) advantage of the LM test is that it is not actually necessary to estimate the model to carry out the test as the statistic is based on the restricted, homoscedastic model.

```
Namelist        ; X       = One,IMUM,FDIUM,LogSales $
Namelist        ; Sectors = RawMtl,InvGood$
? We include Sectors in the model then test the hypothesis that the
? two coefficients are zero.
Probit          ; if[t=5] ; Lhs = IP ; Rhs = X $
Calc            ; Logl0 = LogL $
? Built in command tests using chi squared.
Probit          ; if[t=5] ; Lhs = IP ; Rhs = X,Sectors
                ; Parameters ; test: sectors $
? Likelihood ratio test.
Calc            ; Logl1 = LogL ; List ; LRstat = 2*(logL1 - LogL0) $
? Wald test using matrix algebra
Calc            ; List ; Ctb(.95,2) $
Calc            ; KX = Col(X) ; K1 = KX + 1 ; Kc = Col(Sectors); K = KX + KC$
Matrix          ; c = B(K1:K) ; vc = Varb(K1:K , K1:K) $
Matrix          ; List ; Waldstat = c'<vc>c  $
? Wald test using the Wald command that automates the matrix commands.
Wald            ; start = b ; Var = Varb
                ; labels=Kx_d,Kc_c ; fn1 = c1 - 0 ; fn2 = c2 - 0 $
? Lagrange multiplier test for omitted variables
Probit          ; Lhs = IP ; Rhs = X ; Quietly $
Probit          ; Lhs = IP ; Rhs = X,Sectors
                ; Start = b,0,0 ; LMTest $
? Lagrange multiplier test for heteroscedasticity
Probit          ; if[t=5] ; Lhs = IP ; Rhs = x ; Quietly $
Probit          ; if[t=5] ; Lhs = IP ; Rhs = x ; Het
                ; Hfn = Sectors ; Start = b,0,0 ; LMTest $
```

10. Simulation:

Using the binary choice model simulator, examine how a 1.1 fold increase in LOGSALES which corresponds to a roughly 10% increase in sales would affect the probability of innovation. The BinaryChoice command carries out a simulated change in every observation, and shows what would happen to the predicted sample responses. The Simulate command displays the average predicted probabilities over a range of values of logSales.

```
Probit          ; Lhs = IP ; Rhs=one,logsales,imum,fdium $
BinaryChoice    ; Lhs = IP ; Rhs = one,logsales,imum,fdium
                ; model=probit ; start=b ; scenario: logsales * = 1.1 $
Simulate        ; scenario: & logsales = 5(1)15 ; plot (ci) $$
```