The Stata Journal (2009)
9, Number 4, pp. 584–592

Fitting and interpreting Cragg's tobit alternative using Stata

William J. Burke
Department of Agricultural, Food, and Resource Economics
Michigan State University
East Lansing, MI
burkewi2@msu.edu

Abstract. In this article, I introduce the user-written command `craggit`, which simultaneously fits both tiers of Cragg's (1971, *Econometrica* 39: 829–844) "two-tier" (sometimes called "two-stage" or "double-hurdle") alternative to tobit for corner-solution models. A key limitation to the tobit model is that the probability of a positive value and the actual value, given that it is positive, are determined by the same underlying process (i.e., the same parameters). Cragg proposed a more flexible alternative that allows these outcomes to be determined by separate processes through the incorporation of a probit model in the first tier and a truncated normal model in the second. Also, `tobit` is nested in `craggit`, making the latter a popular choice among "two-tier" models. In the article, I also present postestimation syntax to facilitate the understanding and interpretation of results.

Keywords: st0179, craggit, two tier, two stage, double hurdle, corner-solution models, Craggit's tobit

1 Introduction

Introductory econometric texts commonly reference tobit for fitting models with limited dependent variables that are sometimes called "corner-solution" models.[1] Fitting and interpreting the tobit model is fairly straightforward through the use of `tobit` and associated postestimation commands in Stata.[2] A key limitation to the tobit model is that the probability of a positive value and the actual value, given that it is positive, are determined by the same underlying process (i.e., the same parameters). Cragg (1971) proposed a more flexible alternative, sometimes called a "two-tier" or "double-hurdle" model, which allows these outcomes to be determined by separate processes. Thus far, fitting Cragg's tobit alternative using Stata has required a disjointed process, and interpreting results has been complicated and tedious.

In this article, I present a command, `craggit`, that enables a more coherent fitting of Cragg's model, as well as presenting postestimation syntax to facilitate the interpretation of results. The model in this article is applied to cross-sectional data. The `craggit` command can be used with either balanced or unbalanced panel data as well,

1. For examples, see Pindyck and Rubinfeld (1998), Kennedy (2003), Wooldridge (2009), or Baum (2006).
2. For example, see Baum (2006, 262–266).

 st0179

but only as a pooled estimator (e.g., the `vce(cluster)` option can be used to compute standard errors robust to autocorrelation, but the command is not designed to control for unobserved heterogeneity).

Conceptually, a corner-solution model is where

$$y_i = y_i^* \quad \text{if} \quad y_i^* > 0$$
$$y_i = 0 \quad \text{if} \quad y_i^* \leq 0$$

and

$$y_i^* = \alpha + \boldsymbol{X}_i \boldsymbol{\beta} + \varepsilon_i$$

In practice, as the name suggests, a corner-solution model applies to dependent variables where data are truncated and "piles up" at some given value, but is continuous otherwise. Some examples would be agricultural production or input demands where some observations' optimizing behavior results in no production or demand, while the outcome is continuous for others.

2 The tobit model

Introduced by Tobin (1958), the tobit model proposes the likelihood function,

$$f(y \,|\, x_1) = \{1 - \Phi(\boldsymbol{x}_1 \boldsymbol{\beta}/\sigma)\}^{1(y=0)} \left[(2\pi)^{-\frac{1}{2}} \sigma^{-1} \exp\left\{-(y - \boldsymbol{x}_1 \boldsymbol{\beta})^2 / 2\sigma^2\right\}\right]^{1(y>0)}$$

where Φ is the standard normal cumulative distribution function and exponential indicator functions are $1(y = 0)$ and $1(y > 0)$. There are four values of interest after fitting a tobit model: 1) the probability that y is zero, $P(y_i = 0 \,|\, \boldsymbol{x}_i)$; 2) the probability that y is positive, $P(y_i > 0 \,|\, \boldsymbol{x}_i)$; 3) the expected value of y, conditional on y being positive, $E(y_i \,|\, y_i > 0, \boldsymbol{x}_i)$; and 4) the so-called "unconditional" expected value of y, $E(y_i \,|\, \boldsymbol{x}_i)$.[3]

Then, of course, one could calculate the marginal effects of an explanatory variable on each probability and expectation. Fitting this model is fairly simple using the `tobit` command in Stata, and calculation of these effects around data mean values can be obtained using various `margins` postestimation commands. For a thorough discussion on the tobit model and its interpretation, refer to Wooldridge (2009, 587–595).

3 Cragg's alternative to the tobit model

Again, while useful, the major drawback of the tobit model is that the choice of $y > 0$ and the value of y, given that $y > 0$, is determined by the same vector of parameters ($\boldsymbol{\beta}$ from above). For example, this imposes that the direction (sign) of a given determinant's marginal effect will be the same on both the probability that $y > 0$ and the expectation of y, conditional or otherwise. As an alternative, Cragg proposed the following, which

3. The commonly used phrase "unconditional expectation" is a bit of a misnomer because all expectations are conditional on the explanatory variables (x).

integrates the probit model to determine the probability of $y > 0$ and the truncated normal model for given positive values of y,

$$f(w, y \,|\, \boldsymbol{x}_1, \boldsymbol{x}_2) = \{1 - \Phi(\boldsymbol{x}_1\boldsymbol{\gamma})\}^{1(w=0)} \Big[\Phi(\boldsymbol{x}_1\boldsymbol{\gamma})(2\pi)^{-\frac{1}{2}}\sigma^{-1} \exp\left\{-(y - \boldsymbol{x}_2\boldsymbol{\beta})^2/2\sigma^2\right\}/\Phi(\boldsymbol{x}_2\boldsymbol{\beta}/\sigma)\Big]^{1(w=1)}$$

where w is a binary indicator equal to 1 if y is positive and 0 otherwise. Notice in Cragg's model the probability of $y > 0$ and the value of y, given $y > 0$, are now determined by different mechanisms (the vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$, respectively). Furthermore, there are no restrictions on the elements of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, implying that each decision may even be determined by a different vector of explanatory variables altogether. Also notice that the tobit model is nested within Cragg's alternative because if $\boldsymbol{x}_1 = \boldsymbol{x}_2$ and $\boldsymbol{\gamma} = \boldsymbol{\beta}/\sigma$, the models become identical. For a more thorough discussion of this and other double-hurdle alternatives to tobit, refer to Wooldridge (2002, 536–538).

Fitting Cragg's alternative requires the additional assumption of conditional independence for the latent variable's distribution, or $D(y^* \,|\, w, \boldsymbol{x}) = D(y^* \,|\, \boldsymbol{x})$.[4] Please refer to Wooldridge (forthcoming) for a thorough discussion. In short, this implies that the unbiasedness of results is sensitive to model misspecification.

From Cragg's model, we can obtain the same probabilities and expected values as with tobit by using an updated functional form. The probabilities regarding whether y is positive are

$$P(y_i = 0 \,|\, \boldsymbol{x}_{1i}) = 1 - \Phi(\boldsymbol{x}_{1i}\boldsymbol{\gamma}) \tag{1}$$

$$P(y_i > 0 \,|\, \boldsymbol{x}_{1i}) = \Phi(\boldsymbol{x}_{1i}\boldsymbol{\gamma}) \tag{2}$$

The expected value of y, conditional on $y > 0$ is

$$E(y_i \,|\, y_i > 0, \boldsymbol{x}_{2i}) = \boldsymbol{x}_{2i}\boldsymbol{\beta} + \sigma \times \lambda(\boldsymbol{x}_{2i}\boldsymbol{\beta}/\sigma) \tag{3}$$

where $\lambda(c)$ is the inverse Mills ratio (IMR)

$$\lambda(c) = \phi(c)/\Phi(c)$$

where ϕ is the standard normal probability distribution function. Finally, the "unconditional" expected value of y is

$$E(y_i \,|\, \boldsymbol{x}_{1i}, \boldsymbol{x}_{2i}) = \Phi(\boldsymbol{x}_{1i}\boldsymbol{\gamma})\left\{\boldsymbol{x}_{2i}\boldsymbol{\beta} + \sigma \times \lambda(\boldsymbol{x}_{2i}\boldsymbol{\beta}/\sigma)\right\} \tag{4}$$

For a given observation, the partial effect of an independent variable, x_j, around the probability that $y > 0$ is[5]

$$\frac{\partial P(y > 0 \,|\, \boldsymbol{x}_1)}{\partial x_j} = \gamma_j \phi(\boldsymbol{x}_1\boldsymbol{\gamma}) \tag{5}$$

4. As described in Wooldridge (forthcoming), the exponential type-two tobit relaxes this assumption; however, convergence can be difficult, especially when $x_1 = x_2$.
5. The marginal effect around the probability $y = 0$ is the negative of the value in (5).

where γ_j is the element of $\boldsymbol{\gamma}$ representing the coefficient on x_j. Equations (1), (2), and (5) are the same as the probabilities and partial effect from a probit regression of w on $\boldsymbol{x}_1$. The partial effect of an independent x_j on the expected value of y, given $y > 0$, is

$$\frac{\partial E(y_i \,|\, y_i > 0, \boldsymbol{x}_{2i})}{\partial x_j} = \beta_j \left[1 - \lambda(\boldsymbol{x}_2\boldsymbol{\beta}/\sigma) \left\{\boldsymbol{x}_2\boldsymbol{\beta}/\sigma + \lambda(\boldsymbol{x}_2\boldsymbol{\beta}/\sigma)\right\}\right] \tag{6}$$

where β_j is the element of β representing the coefficient on x_j. Equations (3) and (6) are the same as the expected values and partial effect from a truncated normal regression of y on x_2, with emphasis that the effect is conditional on y being positive.

The partial effect of an independent x_j on the "unconditional" expected value of y is somewhat trickier, because it depends on whether x_j is an element of $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, or both. First, if x_j is an element of both vectors, the partial effect is

$$\begin{aligned}\frac{\partial E(y \,|\, \boldsymbol{x}_1, \boldsymbol{x}_2)}{\partial x_j} &= \gamma_j \phi(\boldsymbol{x}_1\boldsymbol{\gamma}) \times \left\{\boldsymbol{x}_2\boldsymbol{\beta} + \sigma \times \lambda(\boldsymbol{x}_2\boldsymbol{\beta}/\sigma)\right\} \\ &\quad + \Phi(\boldsymbol{x}_1\boldsymbol{\gamma}) \times \beta_j \left[1 - \lambda(\boldsymbol{x}_2\boldsymbol{\beta}/\sigma)\left\{\boldsymbol{x}_2\boldsymbol{\beta}/\sigma + \lambda(\boldsymbol{x}_2\boldsymbol{\beta}/\sigma)\right\}\right] \ \text{ if } x_j \,\epsilon\, \boldsymbol{x}_1, \boldsymbol{x}_2\end{aligned} \tag{7}$$

Now, if x_j is only determining the probability of $y > 0$, then $\beta_j = 0$, and the second term on the right-hand side of (7) is canceled. On the other hand, if x_j is only determining the value of y, given that $y > 0$, then $\gamma_j = 0$, and the first right-hand side term in (7) is canceled. In either of the latter cases, the marginal effect will still be a function of parameters and explanatory variables in both tiers of the regression.

4 Fitting the Cragg model using Stata

The maximum likelihood estimation (MLE) of $\boldsymbol{\gamma}$ can be obtained by regressing w on $\boldsymbol{x}_1$ by using `probit` in Stata. Similarly, the MLE of $\boldsymbol{\beta}$ and $\boldsymbol{\sigma}$ can be obtained by regressing y on $\boldsymbol{x}_2$ by using `truncreg`. Or, all parameters can be fit simultaneously using `craggit`.

The syntax is

`craggit` *depvar1* [*indepvars1*] [*if*] [*in*] [*weight*] `,` `second(`*depvar2*
[*indepvars2*]`)` [`noconstant` `vce(`*vcetype*`)` `hetero(`*varlist*`)` `level(`*#*`)`
maximize_options]

depvar1 is the indicator variable for whether $y > 0$ (w in the above notation), and *indepvars1* are the independent variables explaining that decision ($\boldsymbol{x}_1$ in the above notation). *depvar2* is the continuous response for positive values (y from above), and *indepvars2* are the independent variables explaining it ($\boldsymbol{x}_2$ from above). `second()` is so named because the determinants of y, given $y > 0$, are traditionally thought of as the second "tier" or "hurdle" of the model.

Using an example of household-level demand for subsidized fertilizer in Zambia (these data are described in the appendix available online[6]), output from `craggit` estimation is presented as follows:

```
. craggit basal_g disttown cland educ age,
> second(qbasal_g disttown cland educ age)
Estimating Cragg's tobit alternative
Assumes conditional independence
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -1.316e+08
rescale:       log likelihood = -692711.38
rescale eq:    log likelihood = -7868.8964
Iteration 0:   log likelihood = -7868.8964
  (output omitted)
Iteration 12:  log likelihood = -7498.6146
                                                  Number of obs   =       6378
                                                  Wald chi2(4)    =     310.27
Log likelihood = -7498.6146                       Prob > chi2     =     0.0000

             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Tier1        |
    disttown |  -.0069824   .0010228    -6.83   0.000     -.008987   -.0049779
       cland |   .1013351    .008374    12.10   0.000     .0849223    .1177478
        educ |   .0510291   .0054314     9.40   0.000     .0403837    .0616744
         age |   .0065328   .0014688     4.45   0.000     .0036541    .0094116
       _cons |  -1.717704   .0929843   -18.47   0.000     -1.89995   -1.535458
-------------+----------------------------------------------------------------
Tier2        |
    disttown |  -4.082447   12.82117    -0.32   0.750    -29.21148    21.04659
       cland |   346.8557   195.8594     1.77   0.077    -37.02165     730.733
        educ |   248.2076   178.2567     1.39   0.164    -101.1691    597.5843
         age |    12.1808   20.47023     0.60   0.552    -27.94012    52.30172
       _cons |  -10131.47   7309.131    -1.39   0.166     -24457.1    4194.162
-------------+----------------------------------------------------------------
sigma        |
       _cons |   1110.764   391.2037     2.84   0.005     344.0187    1877.509
```

Results from the section labeled `Tier1` are the MLE of $\boldsymbol{\gamma}$, and those labeled `Tier2` are the MLE of $\boldsymbol{\beta}$ from Cragg's likelihood function. The constant term in the section labeled `sigma` is the MLE of σ.

Whether estimations are obtained simultaneously or one regression at a time, the results will be identical because of the separability of Cragg's likelihood function. That is, while using `craggit` makes estimation more coherent, it will not change results. The primary benefit of using `craggit` is its ability to facilitate postestimation analysis and interpretation.

I emphasize that, as with all double-hurdle models, separability in estimation does not imply separability in interpretation. To illustrate this point, consider a variable that explains the continuous value of y but not the probability that $y > 0$. If fit separately,

6. Visit https://www.msu.edu/~burkewi2/estimating_cragg_appendix.pdf.

the partial effect of such a variable on the unconditional expected value of y [the second right-hand-side term in (7)] would be a function of probit results, even though it would not be included in the probit regression. For another example, x_j may be in both explanatory vectors and have countervailing impacts in each tier (i.e., β_j and γ_j may have different signs). Here the direction of the overall effect cannot be known unless all results are considered together, as in (7).

5 Interpreting results after fitting with craggit

Postestimation analysis commands will be presented for a general case, but it may be useful to follow along with the example by using Zambian fertilizer data presented in the online appendix. After estimation using `craggit`, new variables representing the three scalar values $(\boldsymbol{x}_{1i}\widehat{\boldsymbol{\gamma}}, \boldsymbol{x}_{2i}\widehat{\boldsymbol{\beta}}, \widehat{\sigma})$ can be generated for each observation using `predict`:

```
predict x1g, eq(Tier1)
predict x2b, eq(Tier2)
predict sigma, eq(sigma)
```

The bold terms in the above syntax are the names of the new variables generated by the `predict` commands. The choice of variable name is not important to the final results, except that the names should be used consistently throughout. For this article, generated variable names (and later, program names) will continue to be in bold and will endeavor to be consistent with the notation in (1)–(7).

`craggit` is equipped to handle a model for heteroskedastic standard errors that is a function of observables with the `hetero(`*varlist*`)` option. If that is used, the above command will generate unique values of sigma for each observation ($\widehat{\sigma}_i$).

Now all the information we need to calculate the predicted values in (1)–(4) and partial effects in (5)–(7) is either predicted as a new variable or stored in Stata's active memory. The following commands will generate new variables with unique values for each observation that can then be aggregated to any desired level.

To calculate the probability that $y = 0$ from (1):

```
generate Pw0 = 1 - normal(x1g)
```

To calculate the probability that $y > 0$ from (2):

```
generate Pw1 = normal(x1g)
```

To calculate expected values, it is useful to generate a new variable for the IMR:

```
generate IMR = normalden(x2b/sigma)/normal(x2b/sigma)
```

To calculate the expected value of y, given that $y > 0$, from (3):

```
generate Eyyx2 = x2b + sigma*IMR
```

To calculate the unconditional expected value of y from (4):

```
generate Eyx1x2 = normal(x1g)*(x2b + sigma*IMR)
```

If `indepj` is an independent variable of interest (x_j from the above notation), then to calculate the partial effect on the probability that $y > 0$ from (5):

```
generate dPw1_dxj = [Tier1]_b[indepj]*normalden(x1g)
```

To calculate the partial effect on the expected value of y, given $y > 0$ from (6):

```
generate dEyyx2_dxj=[Tier2]_b[indepj]*(1-IMR*(x2b/sigma+IMR))
```

Finally, to calculate the partial effect on the unconditional expected value of y from (7):

```
generate dEy_dxj=[Tier1]_b[indepj]*normalden(x1g)*(x2b+sigma*IMR)   ///
  +[Tier2]_b[indepj]*normal(x1g)*(1-IMR*(x2b/sigma+IMR))
```

Once these predicted values have been generated for each observation, the average partial effect (APE) of the independent variable can be found using the `summarize` command. For example, the APE on the unconditional expected value of y is the mean of **dEy_dxj**:

```
summarize dEy_dxj
```

We can also see how this APE varies across groups by using the `tabulate` command. Suppose we have a categorical variable, `catvar`:

```
tabulate catvar, summarize(dEy_dxj)
```

The standard deviations reported by these summaries describe only the data and should not be considered a parameter estimate. That is, the standard deviation of the predicted partial effects should not be used as a standard error (SE) for inference on the APE. One viable option for inference on an APE is bootstrapping. Bootstrapping starts by reestimating the model and generating a new APE on a random subsample within the data. This process is repeated many times until many APEs have been computed from numerous random subsamples. The standard deviation from those APEs can be used as an SE for the full sample APE.

Fortunately, Stata can be programmed to run this process for you, albeit after several lines of programming:[7]

7. Thank you to David Tschirley and Ana Fernandez for providing syntax to bootstrap APE standard errors.

```
program define APEboot, rclass
     preserve
     craggit depvar1 indepvars1, second(depvar2 indepvars2)
     predict bsx1g, eq(Tier1)
     predict bsx2b, eq(Tier2)
     predict bssigma, eq(sigma)
     generate bsIMR = normalden(bsx2b/bssigma)/normal(bsx2b/bssigma)
     generate bsdEy_dxj=                                           ///
         [Tier1]_b[indepj]*normalden(bsx1g)*(bsx2b+bssigma*bsIMR)   ///
         +[Tier2]_b[indepj]*normal(bsx1g)*(1-bsIMR*(bsx2b/bssigma+bsIMR))
     summarize bsdEy_dxj
     return scalar ape_xj=r(mean)
     matrix ape_xj=r(ape_xj)
     restore
end
bootstrap ape_xj = r(ape_xj), reps(#): APEboot
```

Here, again, the variable and program names in bold type can be chosen subjectively, so long as they are used consistently and none of the variables generated by the program has the same name as the existing variables. Again the explanatory variable whose APE is being computed is `indepj`. The # sign must be replaced by the number of times Stata should iterate the bootstrap process (the more iterations there are, the longer the process will take, at about 5–10 seconds per iteration; 100 is a reasonable starting point). While the bootstrap is running, after the last command from above is entered, Stata provides progress updates:

```
Bootstrap replications (100)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
.x.......x..x..x..............................x..    50
x...x.......x.....................                  100
```

where each "`.`" represents a completed iteration and the occasional "`x`" represents a "failed" iteration. Here the failed iterations indicate that Stata was not able to fit coefficients for that subsample. This is likely because the random draw did not provide enough variation in the binary indicator variable (w) for `craggit` to converge. Stata ignores these failures, and they will not have much impact on the final results, except to reduce the number of iterations used to compute the bootstrap SE.

To calculate the APE for a different explanatory variable, simply rewrite all the commands from `program` to `bootstrap`, changing the name of `indepj`. First, however, the original program must be cleared from Stata's memory:

```
capture program drop APEboot
```

Unfortunately, bootstrapping can be a time-consuming process, especially when conducting inference on the APE of multiple variables. An alternative would be to approximate a standard error using the delta method. This is when an SE is approximated using a Taylor expansion around the data mean, which can be accomplished after estimation with `craggit` by using the `nlcom` postestimation command and a few local macros.[8] After fitting and predicting **`x1g`** and **`x2b`**:

8. Thank you to Joleen Hadrich and Joshua Ariga for assistance on the syntax for the delta method inference.

```
summarize x1g
local x1gbar = r(mean)
summarize x2b
local x2bbar = r(mean)
nlcom [Tier1]_b[indepj]*normalden(`x1gbar')*(`x2bbar'+[sigma]_b[_cons]*  ///
  (normalden(`x2bbar'/[sigma]_b[_cons])/normal(`x2bbar'/                  ///
  [sigma]_b[_cons])))+[Tier2]_b[indepj]*normal(`x1gbar')                 ///
  *(1- (normalden(`x2bbar'/[sigma]_b[_cons])/normal(`x2bbar'/            ///
  [sigma]_b[_cons]))*(`x2bbar'/[sigma]_b[_cons]+(normalden(`x2bbar'/     ///
  [sigma]_b[_cons])/normal(`x2bbar'/[sigma]_b[_cons]))))
```

The SE provided after `nlcom` can be used with the APE from the `summarize` command to compute the p-value manually[9] (the p-value provided after `nlcom` is only valid for the partial effect at the mean of x, not the APE).

6 References

Baum, C. F. 2006. *An Introduction to Modern Econometrics Using Stata*. College Station, TX: Stata Press.

Cragg, J. G. 1971. Some statistical models for limited dependent variables with application to the demand for durable goods. *Econometrica* 39: 829–844.

Kennedy, P. 2003. *A Guide to Econometrics*. 5th ed. Cambridge, MA: MIT Press.

Pindyck, R. S., and D. L. Rubinfeld. 1998. *Econometric Models and Economic Forecasts*. 4th ed. New York: McGraw–Hill.

Tobin, J. 1958. Estimation of relationships for limited dependent variables. *Econometrica* 26: 24–36.

Wooldridge, J. M. 2002. *Econometric Analysis of Cross Section and Panel Data*. Cambridge, MA: MIT Press.

———. 2009. *Introductory Econometrics: A Modern Approach*. 4th ed. Mason, OH: South-Western.

———. Forthcoming. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

About the author

William J. Burke is a PhD student and agricultural economics research specialist at Michigan State University. He has coauthored several policy analysis studies focused on Africa and has a strong interest in facilitating the use of nonstandard model structures.

9. See page 5 of the online appendix for an example of computing the p-value manually.