Distribution-Free Estimation of Heteroskedastic Binary Response Models in Stata

Jason R. Blevins
Shakeeb Khan

The Ohio State University, Department of Economics
Duke University, Department of Economics

2015 Stata Conference
Columbus, Ohio

Introduction

Based on work from three papers:

1. Khan, S. (2013). Distribution Free Estimation of Heteroskedastic Binary Response Models Using Probit Criterion Functions. *Journal of Econometrics* 172, 168–182.
2. Blevins, J. R. and S. Khan (2013). Local NLLS Estimation of Semiparametric Binary Choice Models. *Econometrics Journal* 16, 135–160.
3. Blevins, J. R. and S. Khan (2013). Distribution-Free Estimation of Heteroskedastic Binary Response Models in Stata. *Stata Journal* 13, 588–602.

Binary Response Models

$$y_i = 1\left\{x_i'\beta + \varepsilon_i > 0\right\}$$

Notation:

- $y_i \in \{0, 1\}$ is an observed response variable
- x_i is a k-vector of observed covariates
- β is a vector of parameters of interest
- ε_i is an unobserved disturbance

Binary Response Models

$$y_i = 1\left\{x_i'\beta + \varepsilon_i > 0\right\}$$

Question: Given a random sample $\{y_i, x_i\}_{i=1}^{n}$, what can we learn about the unknown vector β?

Answer: Not much without saying more about the distribution $F_{\varepsilon|x}$.

Parametric Binary Response Models

If $F_{\varepsilon|x}$ is known, then we can estimate β via ML.

Logit (`logit`):

$$\varepsilon_i \mid x_i \sim \text{Logistic}(0, \sigma^2) \text{ with } \sigma^2 = 1$$

Probit (`probit`):

$$\varepsilon_i \mid x_i \sim \text{N}(0, \sigma^2) \text{ with } \sigma^2 = 1$$

Heteroskedastic probit (`hetprobit`):

$$\varepsilon_i \mid x_i \sim \text{N}(0, \sigma_i^2) \text{ with } \sigma_i^2 = \exp(z_i{}'\gamma)$$

Parametric Binary Response Models

In reality we can't ever know $F_{\varepsilon|x}$. But isn't the normal distribution good enough?

The Logit and Probit models also assume *homoskedasticity*:

$$F_{\varepsilon|x} = F_{\varepsilon}.$$

In general, our estimate of β is inconsistent if $F_{\varepsilon|x}$ is misspecified (either the parametric family or the form of heteroskedasticity).

Two New Semiparametric Estimators

Previous semiparametric approaches require global optimization of difficult functions, nonparametric estimation, etc.

Khan (2013) and Blevins and Khan (2013) are based on Probit criterion functions, which Stata (and almost all other statistical software) handles well already.

Main assumption: $\text{Med}(\varepsilon_i \mid x_i) = 0$ almost surely (conditional median independence).

Nonlinear Least Squares Estimation in Stata

Probit regression model:

$$E[y_i \mid x_i] = \Phi(x_i'\beta)$$

The nonlinear least squares estimator $\widehat{\beta}$ minimizes

$$Q_n(\beta) = \frac{1}{n} \sum_{i=1}^{n} \left[y_i - \Phi\left(x_i'\beta\right)\right]^2$$

Stata's `nl` command fits a nonlinear, parametric regression function $f(x, \theta) = E[y \mid x]$ via least squares. Example:

```
.  nl (y = normal({b0} + {b1}*x1 + {b2}*x2))
```

Local Nonlinear Least Squares Estimator

The local nonlinear least squares (LNLLS) estimator (Blevins and Khan, 2013) is a vector $\hat{\beta}$ that minimizes

$$Q_n(\beta) = \frac{1}{n} \sum_{i=1}^{n} \left[y_i - F\left(\frac{x_i' \beta}{h_n} \right) \right]^2 .$$

F is a nonlinear regression function, such as a cdf.

h_n is a bandwidth sequence such that $h_n \to 0$ as $n \to \infty$.

Scale normalization: $\hat{\beta} = (\hat{\theta}', 1)'$.

Intuition: When $h_n \to 0$, $F\left(\frac{x_i' \beta}{h_n}\right) \to 1\{x_i' \beta > 0\}$.

Local Nonlinear Least Squares Estimator

Choices for the regression function:

1. $F(u) = \Phi(u)$ (the normal CDF)
   - Computationally very similar to NLLS probit.
   - Consistent, limiting distribution is non-Normal.
   - Rate of convergence is $n^{-1/3}$.
   - Jackknifing: optimal rate $n^{-2/5}$ and asymptotic Normality.
2. $F(u) = (1/2 - \alpha_F - \beta_F) + 2\alpha_F \Phi(u) + 2\beta_F \Phi(\sqrt{2}u)$
   - Specifically chosen to reduce bias (α_F, β_F in paper).
   - Consistent and asymptotically Normal.
   - Rate of convergence is $n^{-2/5}$.
   - No need to jackknife.

Example with bandwidth $h_n = 0.1$:

```
. nl (y = normal(({b0} + {b1}*x1 + x2) / 0.1))
```

Local Nonlinear Least Squares Estimator

As with the NLLS probit objective function, the bias-reducing F function can be expressed entirely using Stata's built in `normal` function, for example:

```
.  local h = _N^(-1/5)
.  local index "({b0} + {b1}*x1 + x2) / ‘h’"
.  local beta = 1.0
.  local alpha = -0.5 * (1 - sqrt(2) + sqrt(3))*‘beta’
.  local const = 0.5 - ‘alpha’ - ‘beta’
.  nl (y = ‘const’ + 2*‘alpha’*normal(‘index’)
          + 2*‘beta’*normal(sqrt(2)*‘index’))
```

Local Nonlinear Least Squares Estimator

The jackknife estimator just involves estimating with the normal CDF using two bandwidths $h_{1n} = \kappa_1 n^{-1/5}$ and $h_{2n} = \kappa_2 n^{-1/5}$ forming the weighted sum:

$$\hat{\theta}_{\mathrm{jk}} = w_1 \hat{\theta}_1 + w_2 \hat{\theta}_2,$$

This is also easily done in Stata.

Sieve Nonlinear Least Squares Estimator

The objective function for the sieve nonlinear least squares (SNLLS) estimator of Khan (2013) is also variation on the NLLS probit objective function:

$$Q_n(\theta, g) = \frac{1}{n} \sum_{i=1}^{n} \left[y_i - \Phi\left(x_i{}' \beta \cdot g(x_i) \right) \right]^2$$

where g is an unknown scaling function and $\beta = (\theta', 1)'$ is a vector of parameters.

Based on a new result showing observational equivalence between parametric Probit models with multiplicative heteroskedasticity and semiparametric models under conditional median independence.

Sieve Nonlinear Least Squares Estimator

In practice, approximate g by a linear-in-parameters sieve:

$$g_n(x_i) \equiv \exp(b^{\kappa_n}(x_i)'\gamma_n)$$

where

$$b^{\kappa_n}(x_i) = (b_{01}(x_i), \cdots, b_{0\kappa_n}(x_i))'$$

and γ_n is a κ_n-vector of parameters.

Estimate $\alpha = (\theta, \gamma)$ by minimizing

$$Q_n(\alpha) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - \Phi(x_i'\beta \cdot g_n(x_i))\right]^2.$$

SNLLS Properties

Consistent and asymptotically normal if $\kappa_n \to \infty$ while $\kappa_n / n \to 0$.

Rate of convergence is $n^{-2/5}$.

Choice probabilities can also be estimated:

$$\hat{P}_i = \Phi(x_i' \hat{\beta} \cdot \hat{g}_n(x_i)).$$

SNLLS in Stata via nl

Example with two regressors x_1 and x_2:

$$g_n(x_i) = \exp(\gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_1 x_2 + \gamma_4 x_1^2 + \gamma_5 x_2^2).$$

Again, we could use `nl`:

```
.  nl (y = normal(({b0} + {b1}*x1 + x2)
      * exp({g0} + {g1}*x1 + {g2}*x2
         + {g3}*x1*x2 + {g4}*x1*x1 + {g5}*x2*x2)))
```

Variance-Covariance Matrix Estimation

Although the point estimates reported by `nl` for these estimators will be correct, the reported standard errors are not.

- The point estimates are correct because our estimators are indeed defined by nonlinear least squares criteria.
- The limiting distribution of the Probit NLLS estimator is based on different assumptions, such as $E[\varepsilon_i \mid x_i] = 0$, not $\text{Med}(\varepsilon_i \mid x_i) = 0$.
- Our estimators also perform smoothing and scaling, so the asymptotic properties are different.
- Among other things, a custom Stata package allows us to report appropriate standard errors.

The DFBR Package

The `dfbr` command handles several messy, error-prone steps:

- Automates specifying objective function and parameters.
- Feasible optimal bandwidth estimation for LNLLS.
- Jackknife weight and bandwidth selection for LNLLS.
- Automatic sieve basis construction for SNLLS.
- Calculates bootstrap standard errors for both estimators.

Implemented in Mata

Mata is a fast, C-like language used internally by many Stata routines.

The critical parts of `dfbr` are implemented in Mata:

- Optimization (multiple starting values, NM and BFGS).
- Analytical gradients and Hessians.
- Bootstrapping (via `moremata`, Jann, 2005)

Installation and Usage

Installation:

```
.  ssc install moremata
.  net install dfbr, from(http://jblevins.org/)
.  help dfbr
```

Sieve nonlinear least squares estimation (default):

```
  dfbr depvar indepvars [if] [in]
      [, sieve basis(basis_vars) options]
```

Local nonlinear least squares estimation:

```
  dfbr depvar indepvars [if] [in], local
      [normal bandwidth(#) options]
```

Data Generation

```
. set obs 1000
. gen x1 = invnormal(runiform())
. gen x2 = 1 + invnormal(runiform())
```

```
. generate eps = sqrt(12)*uniform() - sqrt(12)/2
. replace eps = exp(x1 * abs(x2) / x2) * eps
```

```
. generate y = -0.3 + 2.1 * x1 + x2 + eps > 0
```

Local NLLS Example

```
. dfbr y x1 x2, local brep(100) bandwidth(0.25)

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

Local Nonlinear Least Squares (LNLLS)           Number of obs     =       1000
                                                Bandwidth         = 2.50000e-01
```

y	Observed Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	-.2685403	.2724087	-0.99	0.324	-.8024516	.265371
x1	1.95139	.3475335	5.61	0.000	1.270237	2.632543

Coefficient on x2 normalized to 1.

Jackknife NLLS Example

```
. dfbr y x1 x2, local normal

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

Local Nonlinear Least Squares (LNLLS)         Number of obs   =       1000

------------------------------------------------------------------------------
             |   Observed
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |  -.1861059   .3290632    -0.57   0.572     -.831058    .4588462
          x1 |   2.015668   .4906921     4.11   0.000     1.053929    2.977406
------------------------------------------------------------------------------
Coefficient on x2 normalized to 1.
```

Sieve NLLS Example

```
. dfbr y x1 x2, sieve basis(x1 x2 x1x2 x1_2 x2_2)

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

Sieve Nonlinear Least Squares (SNLLS)         Number of obs   =       1000
```

y	Observed Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cons	-.180205	.1120847	-1.61	0.108	-.399887	.039477
x1	2.136893	.1772399	12.06	0.000	1.789509	2.484277

```
Coefficient on x2 normalized to 1.
Sieve basis: _cons x1 x2 x1x2 x1_2 x2_2
```

Monte Carlo Experiments

$$y = 1\{-0.3 + 2.1x_{1i} + x_{2i} + \varepsilon_i > 0\}$$
$$x_{1i} \sim \mathrm{N}(0,1)$$
$$x_{2i} \sim \mathrm{N}(1,1)$$

Three distributions of ε_i:

1. Homoskedastic Normal: $\mathrm{N}(0,1)$.
2. Heteroskedastic Normal: $\mathrm{N}(0,\sigma_i^2)$ with $\sigma_i = \exp(x_{1i}\,|x_{2i}|\,/x_{2i})$.
3. Heteroskedastic Uniform: $\mathrm{U}(0,1)$, standardized and multiplied by σ_i.

101 replications each using 1,000 observations

Monte Carlo Experiments

Table: Homoskedastic Normal

	β_0		β_1	
Estimator	Bias	MSE	Bias	MSE
Logit	0.004	0.000	-0.021	0.000
Probit	0.004	0.000	-0.022	0.001
Het. Probit	0.003	0.000	-0.015	0.001
Local NLLS	-0.002	0.000	-0.028	0.002
Jackknife NLLS	0.006	0.000	-0.010	0.002
Sieve NLLS	0.002	0.000	-0.025	0.001

Monte Carlo Experiments

Table: Heteroskedastic Normal

	β_0		β_1	
Estimator	Bias	MSE	Bias	MSE
Logit	0.341	0.116	0.526	0.277
Probit	0.377	0.143	0.586	0.343
Het. Probit	0.015	0.000	-0.183	0.035
Local NLLS	0.009	0.000	-0.002	0.002
Jackknife NLLS	0.013	0.001	0.003	0.004
Sieve NLLS	0.045	0.002	0.093	0.010

Monte Carlo Experiments

Table: Heteroskedastic Uniform

	β_0		β_1	
Estimator	Bias	MSE	Bias	MSE
Logit	0.419	0.176	0.578	0.334
Probit	0.452	0.205	0.625	0.391
Het. Probit	-0.054	0.003	-0.453	0.207
Local NLLS	-0.001	0.001	-0.113	0.020
Jackknife NLLS	-0.007	0.001	-0.113	0.021
Sieve NLLS	0.087	0.007	0.143	0.021

Conclusion

Installation:

```
.  ssc install moremata
.  net install dfbr, from(http://jblevins.org/)
.  help dfbr
```

More information:

http://jblevins.org/research/dfbr/