DPB: Dynamic Panel Binary data models in `gretl`

Riccardo Lucchetti, Claudia Pigini

Università Politecnica delle Marche

Fourth Gretl Conference

Berlin, June 13^{th}, 2015

Dynamic binary panel data models

- Non-linear dynamic models for binary dependent variables are becoming essential in microeconometrics, especially given the increasing availability of panel datasets; examples:

  female labor supply, fertility choices, self-assessed health condition, poverty traps, remittance decisions by migrants, access to credit

- Static models are relatively mainstream and are supported by most statistical and econometric software

- Dynamic models are more complex to implement and estimation routines are not always readily available to the practitioner

Main issues

- Consider the model

$$\begin{aligned} y^*_{it} &= \gamma y_{it-1} + \mathbf{x}'_{it}\boldsymbol{\beta} + \alpha_i + \varepsilon_{it} \\ y_{it} &= 1\{y^*_{it} \geq 0\} \quad \text{for} \quad i = 1, \ldots, n \;\; t = 2, \ldots, T \end{aligned}$$

- Heckman (1981a): given the observable characteristics $\mathbf{x}_{it}$, it's important to separate:
  - *true state dependence*: how experiencing an event in the present affects the probability of that same event occurring in the future
  - *permanent unobserved heterogeneity*: propensity to experience that same event at all times
- Dealing with unobserved heterogeneity raises the *initial conditions problem*: α_i is correlated with the initial observation.

Estimation approaches

- *Random-effects* approaches: modelling the joint distribution of the outcomes conditional on y_1
  - Heckman (1981b): models $y_{i1}|\alpha_i$ via a separate approximate reduced-form model
  - Wooldridge (2005): models $\alpha_i|y_{i1}$ via the history of strictly exogenous covariates

- *Fixed-effects* approaches for fixed T: condition the joint distribution of $\mathbf{y}_i$ on a suitable sufficient statistic for α_i, which exists
  - in absence of covariates with $\text{T} = 3$ (Chamberlain, 1985);
  - with covariates on the basis of a weighted conditional log-likelihood (Honorè and Kyriazidou, 2000). Convergence slower than $\sqrt{n}$
  - in a Quadratic Exponential model (Bartolucci and Nigro, 2010).

Estimators implemented in DPB

- *Dynamic Probit* (DP) proposed by Heckman (1981b)
- *AR1 Dynamic Probit* (ADP) proposed by Hyslop (1999) building on the DP model
- *Generalised AR1 Dynamic Probit* (GADP) proposed by Keane and Sauer (2009) building on the DP and ADP model
- *Quadratic Exponential* (QE) proposed by Bartolucci and Nigro (2010)

Wooldridge's estimator can be implemented with the built-in command

```
probit depvar wool_indepvars --random-effects
```

Dynamic probit model

In the DP model (Heckman, 1981b), for $i = 1, \ldots, n$

$$\begin{aligned} y_{it} &= 1\{\gamma y_{it-1} + \mathbf{x}_{it}'\boldsymbol{\beta} + \alpha_i + \varepsilon_{it} \geq 0\} \quad \text{for} \quad t = 2, \ldots, T \\ y_{i1} &= 1\{\mathbf{z}_{i1}'\boldsymbol{\pi} + \theta\alpha_i + \varepsilon_{i1} \geq 0\} \end{aligned}$$

- y_{it}: binary response variable; $\mathbf{x}_{it}$: individual characteristics;
- $\mathrm{E}\left[\varepsilon_{it}|\mathbf{X}_i, \alpha_i\right] = 0$; $\mathrm{E}\left[\alpha_i|\mathbf{X}_i\right] = 0$
- $[\theta\alpha_i + \varepsilon_{i1}, \alpha_i + \varepsilon_{i2}, \ldots, \alpha_i + \varepsilon_{iT}]' \sim N(\mathbf{0}; \boldsymbol{\Sigma})$; $V(\alpha_i) = \sigma_\alpha^2$; $V(\varepsilon_{it}) = 1$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1+\theta^2\sigma_\alpha^2 & \theta\sigma_\alpha^2 & \theta\sigma_\alpha^2 & \ldots \\ \theta\sigma_\alpha^2 & 1+\sigma_\alpha^2 & \sigma_\alpha^2 & \ldots \\ \theta\sigma_\alpha^2 & \sigma_\alpha^2 & 1+\sigma_\alpha^2 & \ldots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Dynamic Probit model — ML estimation

Under the above premises, the parameter vector $\psi = \left[\beta', \gamma, \pi', \theta, \sigma_\alpha\right]$ can be estimated by ML.

The contribution to the likelihood by unit i is:

$$\begin{aligned} \mathscr{L}_i(\psi) &= \int_{\mathbb{R}} \Big\{ \Phi\left[(\mathbf{z}'_{i1}\boldsymbol{\pi} + \theta\alpha_i)(2y_{i1} - 1)\right] \times \\ &\quad \prod_{t=2}^{T} \Phi\left[(\gamma y_{it-1} + \mathbf{x}'_{it}\boldsymbol{\beta} + \alpha_i)(2y_{it} - 1)\right] \Big\} \mathrm{d}\Phi\left(\frac{\alpha_i}{\sigma_\alpha}\right) \end{aligned}$$

where $\Phi(\cdot)$ is the standard normal c.d.f. The integral over α_i can be evaluated numerically by Gauss-Hermite quadrature (Butler and Moffitt, 1982).

In practice. . . a simple hansl script

1. Include the function package and load your dataset (setting up the panel structure if necessary)

```
include DPB.gfn
open DP_artdata.gdtb
setobs id time --panel-vars
```

2. Create two lists: one for the main equation and one for the initial condition

```
list X = const x
list Z = const x z
```

3. Call the package public functions to set up the model, estimate the parameters and print the output:

```
bundle b = DPB_setup("DP",y,X,Z)
DPB_estimate(&b)
DPB_printout(&b)
```

Handling options

You can use the public function DPB_setoption to change the default:

- *number of quadrature points*:

  ```
  err = DPB_setoption(&b, "nrep", 32)
  ```

  default is 24

- *estimator of the covariance matrix*:

  ```
  err = DPB_setoption(&b, "vcv", 1)
  ```

  0 Sandwich (default), 1 OPG, 2 Hessian

- *verbosity level*:

  ```
  err = DPB_setoption(&b, "verbose", 2)
  ```

  1 log-lik at each iteration (default), 0 no output, 2 verbose mle

Gauss-Hermite quadrature in gretl

To compute the probability for the DP log-likelihood, we use the built in function quadtable()

```
matrix h = quadtable(quadpoints, 1, 0, 1)
matrix alphas = h[,1] .* sig_a
scalar LL_i = p * h[,2]
```

- quadtable() is more general than Gauss-Hermite (GaussLegendre and GaussLaguerre also available)
- In principle, not impossible to write analytical derivatives in hansl (advantages not certain, however; see Rabe-Hesketh et al. (2002))
- Possible margins for improvement: parallelization, adaptive GHQ

AR1 Dynamic Probit model

Hyslop (1999) generalised the DP model to accommodate autocorrelated errors. For $t = 2, \dots, T$,

$$\varepsilon_{it} = \rho \varepsilon_{it-1} + \eta_{it}$$

$$|\rho| \leq 1, \quad \eta_{it} \sim N(0, 1 - \rho^2)$$

The variance-covariance matrix of the errors becomes:

$$\mathbf{\Sigma} = \begin{bmatrix} 1 + \theta^2\sigma^2_\alpha & \rho + \theta\sigma^2_\alpha & \rho^2 + \theta\sigma^2_\alpha & \dots \\ \rho + \theta\sigma^2_\alpha & 1 + \sigma^2_\alpha & \rho + \sigma^2_\alpha & \dots \\ \rho^2 + \theta\sigma^2_\alpha & \rho + \sigma^2_\alpha & 1 + \sigma^2_\alpha & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Note that the ADP reduces to the DP model for $\rho = 0$.

Generalised AR1 Dynamic Probit model

More general version of $\boldsymbol{\Sigma}$ by Keane and Sauer (2009). The initial condition becomes

$$y_{i1} = 1\{\mathbf{z}_{i1}'\boldsymbol{\pi} + \theta\alpha_i + u_i \geq 0\},$$

with $-1 < \tau = E(u_i \cdot \varepsilon_{i2}) \leq 1$, since $V(u_i) = V(\varepsilon_{it}) = 1$ for identification, so

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1+\theta^2\sigma_\alpha^2 & \tau\rho+\theta\sigma_\alpha^2 & \tau\rho^2+\theta\sigma_\alpha^2 & \dots \\ \tau\rho+\theta\sigma_\alpha^2 & 1+\sigma_\alpha^2 & \rho+\sigma_\alpha^2 & \dots \\ \tau\rho^2+\theta\sigma_\alpha^2 & \rho+\sigma_\alpha^2 & 1+\sigma_\alpha^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

The GADP reduces to the ADP model for $\tau = 1$ and to the DP for $\rho = 0$. Nice, but possibly very weakly identified.

Likelihood for ADP and GADP models

- The specification of ε_{it} as an AR(1) process makes it impossible to integrate out the random effect α_i via quadrature;
- in order to compute the likelihood, T-variate normal probabilities must be evaluated by simulation via the GHK algorithm:

$$\mathscr{L}_i^*(\psi) = \frac{1}{R}\sum_{r=1}^{R} \Phi^*_{Tr}(\mathbf{a}_i, \mathbf{b}_i, \mathbf{C})$$

where

- $\mathbf{a}_i$ and $\mathbf{b}_i$, are the integration limits (possibly nonfinite)
- $\mathbf{C} = \mathrm{cholesky}(\boldsymbol{\Sigma})$ in the ADP or GADP models
- r is the number of random draws in the simulation.

The GHK algorithm in gretl

The Geweke (1989), Hajivassiliou and McFadden (1998), and Keane (1994) algorithm, is computed by the *internally parallelized* built-in function ghk()

```
scalar inf = $huge
U = halton(TT, nrep)
Y = Y0 | Y1
tndx_i = ndx_0[i] | ndx_i
top_i = (inf .* Y - tndx_i .* (!Y))
bot_i = (-tndx_i .* Y - inf .* (!Y))
matrix S_i = {}
P i = ghk(C[1:T,1:T], bot_i, top_i, U[1:T,], &S_i)
```

U is a $T \times R$ matrix containing a sequence of draws (Halton or uniform) to use in the simulation; S_i will contain the analytical score on exit.

Derivatives

Analytical derivatives have two advantages:

1. computation speedup
2. improved precision when computing Hessian (numerical)

We use the chain rule

$$\frac{\partial \mathscr{L}_i^*(\psi)}{\partial \psi} = \frac{\partial \frac{1}{R}\sum_{r=1}^{R} \Phi_{Tr}^*(\mathbf{a}_i, \mathbf{b}_i, \mathbf{C})}{\partial [\mathbf{a}_i, \mathbf{b}_i, \mathrm{vech}(\mathrm{C})]} \times \frac{\partial [\mathbf{a}_i, \mathbf{b}_i, \mathrm{vech}(\mathrm{C})]}{\partial \psi}$$

where

- $\frac{\partial \frac{1}{R}\sum_{r=1}^{R} \Phi_{Tr}^*(\mathbf{a}_i, \mathbf{b}_i, \mathbf{C})}{\partial [\mathbf{a}_i, \mathbf{b}_i, \mathrm{vech}(\mathrm{C})]}$ is *already implemented in C*
- $\frac{\partial [\mathbf{a}_i, \mathbf{b}_i, \mathrm{vech}(\mathrm{C})]}{\partial \psi}$ is *implemented in hansl for DPB*

More on options

- Changing the *sequence of draws*:

```
b = DPB_setup("ADP", y, X, Z)
foo = DPB_setoption(&b, "draws", 1)
```

  0 Halton (default), 1 Uniform

- Changing *the number of GHK replications*:

```
b = DPB_setup("GADP", y, X, Z)
bar = DPB_setoption(&b, "nrep", 200)
```

  128 default

- Only for the DP model, change the *method*:

```
b = DPB_setup("DP", y, X, Z)
baz = DPB_setoption(&b, "method", 1)
```

  0 GHQ (default), 1 GHK. For ADP and GADP only GHK is allowed.

Quadratic Exponential model

Dynamic logit model: no general way to derive a sufficient statistic for the incidental parameters

Quadratic Exponential model (Bartolucci and Nigro, 2010): directly defines the joint probability of $\mathbf{y}_i$

$$p(\mathbf{y}_i|\mathbf{X}_i, y_{i1}, \alpha_i; \psi) = \frac{\exp\left(\sum_t y_{it} y_{it-1}\gamma + \sum_t y_{it}\mathbf{x}'_{it}\boldsymbol{\beta}_1 + y_{iT}\left(\mu + \mathbf{x}'_{iT}\boldsymbol{\beta}_2\right) + y_{i+}\alpha_i\right)}{\sum_{\mathbf{b}\in\mathbb{B}} \exp\left(\sum_t b_t b_{t-1}\gamma + \sum_t b_t\mathbf{x}'_{it}\boldsymbol{\beta}_1 + b_T\left(\mu + \mathbf{x}'_{iT}\boldsymbol{\beta}_2\right) + b_+\alpha_i\right)}$$

where $\mathbb{B} \equiv \left\{\mathbf{b} : \mathbf{b} \in \{0,1\}^T\right\}$, that is the set of all possible T-vectors $\mathbf{b}$ containing zeros and ones.

Quadratic Exponential model

The conditional distribution based on the *total score* y_{i+} is

$$p(\mathbf{y}_i|\mathbf{X}_i, y_{i1}, y_{i+}; \boldsymbol{\psi}) = \frac{p(\mathbf{y}_i|\mathbf{X}_i, y_{i1}, \alpha_i; \boldsymbol{\psi})}{p(y_{i+}|\mathbf{X}_i, y_{i1}, \alpha_i; \boldsymbol{\psi})} =$$

$$\frac{\exp\left[\sum_t y_{it-1} y_{it} \gamma + \sum_t y_{it} \mathbf{x}'_{it} \boldsymbol{\beta}_1 + y_{iT}\left(\mu + \mathbf{x}'_{iT}\boldsymbol{\beta}_2\right)\right]}{\sum_{\mathbf{b}: b_+ = y_{i+}} \exp\left[\sum_t b_t b_{t-1} \gamma + \sum_t b_t \mathbf{x}'_{it} \boldsymbol{\beta}_1 + b_T\left(\mu + \mathbf{x}'_{iT}\boldsymbol{\beta}_2\right)\right]}.$$

The conditional log-likelihood can be written as

$$\ell(\boldsymbol{\psi}) = \sum_{i=1}^{n} 1\{0 < y_{i+} < T\} \log p(\mathbf{y}_i|\mathbf{X}_i, y_{i1}, y_{i+}; \boldsymbol{\psi})$$

and maximised with respect to $\boldsymbol{\psi} = \left[\gamma, \boldsymbol{\beta}'_1, \mu, \boldsymbol{\beta}'_2\right]'$.

Computation of the denominator in the QE model

Consider the expression

$$\sum_{\mathbf{b}:b_+=y_{i+}} \exp\left[\sum_t b_t b_{t-1}\gamma + \sum_t b_t \mathbf{x}'_{it}\boldsymbol{\beta}_1 + b_T\left(\mu + \mathbf{x}'_{iT}\boldsymbol{\beta}_2\right)\right]$$

The “exp” term has to be computed

- for each individual (possibly, tens of thousands)
- at every iteration (possibly, hundreds)

and the sum may go over several hundred terms $\left[\frac{T!}{y_+!(T-y_+)!}\right]$.

For performance reasons, we want to *avoid loops* as much as possible, and to *precompute* quantities as much as possible.

Computation of the denominator in the QE model

The denominator can be written as a function of the matrix Q_j, where $j = j(T_i, y_{i+}, y_{i*})$, $y_{i*} = \sum_t y_{it-1} y_{it}$, with structure

$$Q_j = \left[Q^1(T, y_+) \middle| q^2(y_*) \right]$$

- $Q^1(T, y_+)$: a matrix with $\frac{T!}{y_+!(T-y_+)!}$ rows and T columns whose rows are $\mathbf{b} : b_+ = y_{i+}$
- $q^2(y_*)$ holds the number of consecutive ones in the corresponding row of Q^1.

Computation of the denominator in the QE model

For example: $\mathbf{y}_i = [0, 1, 1, 0]$, so $T_i = 4$, $y_+ = 2$, $y_* = 1$

$$Q = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

The $Q^1(T, y_+)$ matrices can be computed recursively with the special cases $Q^1(n, 0)$, $Q^1(n, 1)$, and $Q^1(n, n)$.

Computation of the denominator in the QE model

Since none of the Q_j depend on parameters, they are *precomputed* with the following algorithm:

1. initialise an empty array of matrices Q;
2. for each individual i;
   1. compute the $j(i)$ index as a function of T_i, y_{i+} and y_{i*};
   2. if Q_j has already been computed, stop and go to the next individual i; else
      1. compute $Q^1(T_i, y_{i+})$ via the recursive method described above;
      2. compute Q_j;
      3. store Q_j into the array at position j.

The likelihood becomes a simple function of Q_j, which doesn't need to be recomputed during Newton-Raphson iterations.

Performance comparisons

We compare DPB with:

- redprob: Stata module for the DP model
- redpace: Stata module for the ADP model
- cquadext: Stata command for the QE model (in the cquad module)
- cquad_ext: R function for the QE model (in the cquad package)

We use the union dataset ($N = 799$, $T = 6$) and replicate the example in Stewart (Stata Journal, 2006).

24 quadrature points, 500 GHK replications.

Results obtained on a system with
32 Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00GHz

Performance comparisons

	Probit	RE-Probit	DP-ghq	DP-ghk*	ADP*	QE
Gretl						
log-lik	-1573.64	-1563.18	-1860.21	-1860.27	-1854.618	-467.49
n. of it			41	36	44	5
time	0.02s	02.26s	15.33s	04m.06.52s	05m.20.67s	0.14s
Stata-mp						
log-lik	-1573.74	-1563.18	-1860.21	-1861.04	-1855.49	-467.65
n. of it	4	7	5	4	3	5
time	0.11s	0.63s	41.69s	34m.02.20s	31m.09.89s	0.39s
R						
log-lik						-467.65
n. of it						5
time						2.24s

* start form a specific vector of initial values provided in Stewart (2006)

Conclusions

Other issues addressed in the paper

1. Treatment of missing values in unbalanced panels
2. Identification of autocorrelation coefficients in short panels for RE probit models with AR(1) disturbances
3. A detailed example on how to compute Wooldridge's estimator
4. A detailed example on how to compute partial effects

For further details:

Lucchetti, R. and Pigini, C. (2015) "DPB: Dynamic Panel Binary data models in Gretl", gretl working paper $\#$ 1

Further research: *Dynamic sample selection model*