Using STATA's - ml method d2 - to estimate a multistate Markov transition model with unobserved heterogeneity

Thomas Buettner
London School of Economics

May 22, 2003

1 CAVEAT....

Inference on the basis of ONE observation...

2 STATA's - ml methods -

If "linear form restrictions" are met, i.e. $\ln L = \sum_i \ln L_i$;

- - lf - supply likelihood function only

If "linear form restrictions" are violated;

- - d0 - code likelihood function only
- - d1 - code analytical gradient
- - d2 - code analytical Hessian

3 The main message ....

When - ml method d0 -

- has trouble converging (”numerical derivatives cannot be computed”)
- or is just too slow,

STATA’s - ml method d2 - can offer HUGE improvements

- convergence and
- speed of convergence

Coding a - ml method d2 - estimator

- is in principle straightforward,

- **BUT** can be complicated by STATA’s limited matrix capabilities.
  [NOTE: Bobby Gutierrez pointed out during the meeting that there is an undocumented routine - mlmatbysum - that solves my problem]

4 The state transition model

- time is discrete t
- there are S discrete states
- in each period t, we observe each individual i, its
  - state s_{it}
  - individual specific characteristics x_{it} (potentially time varying)

t	0	1	2	3	4	5	...
s_t	2	2	1	2	3	2	...
x_t	x_0	x_1	x_2	x_3	x_4	x_5	...
$Transprob$	$P(22\|x_0)$	$P(21\|x_1)$	$P(12\|x_2)$	$P(23\|x_3)$	$P(32\|x_4)$	...	...

- without unobserved heterogeneity: MNL (- mlogit -) for each state separately;
- consider unobserved heterogeneity in the form of discrete types a la Heckman-Singer (1984)
  location and mass of types to be estimated jointly with other parameters

So, conditional on x_{it}, we get transition matrices:

Type 1 (mass q_1)

		State in $t+1$		
		1	2	3
State in t	1	.	.	.
	2	.	.	.
	3	.	.	.

Type 2 (mass q_2)

		State in $t+1$		
		1	2	3
State in t	1	.	.	.
	2	.	.	.
	3	.	.	.

5 The likelihood function

Conditional on individual i being of type j, the likelihood of observing the sequence of states $\{s_{it}\}_{t=t0}^{t_1}$ is

$$L_{ij}=\prod_{t=t_0}^{t_1-1} P(s_{it}\rightarrow s_{it+1}|x_{it},\gamma=j,\beta_{s_{it},s_{it+1}})$$

The unconditional likelihood for the individual becomes:

$$L_i=\sum_{j\in\Gamma} q_j L_{ij}$$

And the sample log likelihood takes the form:

$$\ln L=\sum_{i=1}^{N}\ln L_i=\sum_{i=1}^{N}\ln\left(\sum_{j\in\Gamma} q_j L_{ij}\right)$$

Note: "Linear form restrictions" are not met.
The total coefficient vector to be estimated is:

$$(\beta_{1,1},\beta_{1,2},...,\beta_{n,n-1},\beta_{n,n},q_1,...,q_m)$$

5.1 The gradient:

"Typical element":

$$\frac{\partial \ln L}{\partial \beta_{kl}} = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{t=t0}^{t1} q_j \frac{L_{ij}}{L_i} \frac{1}{P_{ijt}} \frac{\partial P_{ijt}}{\partial \beta_{kl}}$$

5.2 The Hessian:

"Typical element":

$$\begin{aligned} \frac{\partial \ln L}{\partial \beta_{kl} \partial \beta_{mn}} &= \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{t=t0}^{t1} q_j \frac{L_{ij}}{L_i} \frac{1}{P_{ijt}} \\ & \left\{ \frac{1}{L_{ij}} \frac{\partial L_{ij}}{\partial \beta_{kl}} \frac{\partial P_{ijt}}{\partial \beta_{mn}} - \frac{1}{L_i} \frac{\partial L_i}{\partial \beta_{kl}} \frac{\partial P_{ijt}}{\partial \beta_{mn}} - \frac{1}{P_{ijt}} \frac{\partial P_{ijt}}{\partial \beta_{kl}} \frac{\partial P_{ijt}}{\partial \beta_{mn}} + \frac{\partial P_{ijt}}{\partial \beta_{kl} \partial \beta_{mn}} \right\} \end{aligned}$$

6 Computational issues

6.1 Constraints:

- System memory
- Computing time

→ want to generate as few temporary variables as possible

6.2 Issues:

- Expand data (which has $N * T$ obs. to start with) by a factor M to facilitate computation
- Log likelihood and gradient straightforward to compute

6.3 Problem:

- Hessian involves computation of terms of the form:

$$A'WB$$

$$\begin{aligned} \text{where } A &= (a * [1, 1, ..., 1]). * X \quad \text{(element by element)} \\ B &= (b * [1, 1, ..., 1]). * X \quad \text{(element by element)} \\ & W\text{- block diagonal with blocks of ones} \end{aligned}$$

- STATA cannot do this!!! Or at least I couldn't...
  [NOTE: As Bobby Gutierrez pointed out during the meeting,
  STATA CAN DO THIS with the, so far, undocumented -ML MATBYSUM-
  so the problem is solved] .

- W is square with dimensions $N * T * M$ (i.e. potentially very big (> max matsize SE: 11000))

6.4 "Solution"

Circumvent problem by manually multiplying in weights:

- create A : each variable in X multiplied by a.
- create B : each variable in X multiplied by a.
- use - matrix glsaccum - to get $(A,B)'W(A,B) = \begin{bmatrix} A'WA & A'WB \\ B'WA & B'WB \end{bmatrix}$ and only use the $A'WB$ part.
- This involves generating LOADS of temporary variables

  ... takes a long time to compute

... and a lot of memory – I once ran out of memory on a 256M machine

(recall that data is already expanded)

...BUT: IT WORKS !!!