Implementation of a multinomial logit model with fixed effects

Klaus Pforr

Mannheim Centre for European Social Research (MZES)
University of Mannheim
klaus.pforr@mzes.uni-mannheim.de

July 1, 2011,
Ninth German Stata Users Group Meeting, Bamberg

Outline

Motivation

Statistical model

Implementation

First applications

Outlook

Motivation

Why mlogit?

- Fixed effect models available for continuous, binary and count data dependent variables.
- Polytomous categorical dependent variables commonly used in all fields of social sciences.

Why fixed effects?

Counter omitted variable bias!

- With fixed effects models no assumptions about α_i necessary.
- Random effects and pooled models basically assume no correlation of α_i and X_{it}.

Statistical model

mlogit across time with unobserved heterogeneity

$$\Pr(y_{it}=j)=\frac{\exp(\alpha_{ij}+X_{it}\beta_j')}{1+\sum_{k=1,k\neq B}^{J}\exp(\alpha_{ij}+X_{it}\beta_k')} \quad \text{for } j\neq \text{ base outcome } B$$

$$\Pr(y_{it}=B)=\frac{1}{1+\sum_{k=1,k\neq B}^{J}\exp(\alpha_{ij}+X_{it}\beta_k')}$$

Solution by Chamberlain(1980)

- $\sum_{t=1}^{T_i} y_{itj}$ is sufficient statistic for α_{ij}
- Cond. probability model: Prob. of sequence $y_{i1},\ldots,y_{iT_i}$ cond. of "overall tendency" to each outcome $j\neq B$.
- α_i disappeares!

$$\Pr(y_i|\bigwedge_{j\neq B}\sum_{t=1}^{T_i} y_{itj})=\frac{\prod_{t=1}^{T_i}\prod_{j=1,j\neq B}^{J}\exp(X_{it}\beta_j')^{y_{itj}}}{\sum_{d_i\in\Delta_i}((\prod_{t=1}^{T_i}\prod_{j=1,j\neq B}^{J}\exp(X_{it}\beta_j')^{d_{itj}}))}$$

with

$$\Delta_i=\{(d_{i1},\ldots,d_{iT_i})'|\forall j=1,\ldots,J, j\neq B:\sum_{t=1}^{T_i} d_{itj}=k_{ij}\}.$$

Statistical model (cont.)

Δ_i is the set of all permutations of y_i.

Example: Let y_i=(1,2,3).

$\Delta_i = \{(1,2,3),(1,3,2),(2,1,3),(2,3,1),(3,1,2),(3,2,1)\}$.

Estimation with maximum-likelihood

The log. likelihood function:

$$\ln L = \sum_i \Big(\sum_{j \neq B} \sum_t y_{itj} X_{it} \beta_j' - \ln \sum_{\Delta_i} \exp \sum_{j \neq B} \sum_t d_{itj} X_{it} \beta_j' \Big)$$

Implementation: General layout

Top-level ado

- Syntax
- Further preparation

Actual estimation with maximum likelihood

- Iteration management & display of results via Stata `ml`
- Log likelihood, gradient, Hessian with Mata evaluator function

Implementation: Top-level ado

"Outer shell"

- Standard parsing with `syntax`: varlist, group id, optional base outcome
- Missings: Standard listwise deletion via `markout`
- Collinear Variables: Copied & adjusted `_rmcoll` from `mlogit`
- Matsize check: Copied & adjusted from `clogit`
- Editing of equations for ml: Copied & adjusted from `mlogit`
- Offending observations/groups, i.e. checks variance in dep. & indep. var's; copied & adjusted from `clogit`
- Init. values: inspired by `clogit`
- Remaining preparation for mata function:
  - Globals for group id var., indep. var's for ml evaluator function
  - Matrix out2eq: Mapping from outcome indices to outcomes values and equation indices.

Implementation: Maximum likelihood

"Interface": Stata `ml`

Putting equations in Stata's ml terminology

- Panel structure $\Rightarrow$ no likelihood defined at observation level $\Rightarrow$ d-family method
- Computation speed and accurary $\Rightarrow$ d2 method, i.e. $\ln L, g, H$ have to be analytically derived
- J-1 equations, i.e. $(\mathbf{y}_1, \ldots, \mathbf{y}_{J-1}) = (y_1, \ldots, y_{B-1}, y_{B+1}, \ldots, y_J)$
- J-1 parameters $\theta_j = X_{it}\beta_j'$; not used, direct use of $(J-1) \times M$ coefficients β_{jm}

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j, k \neq B$$

Process step-by-step:

1. Declare variables.

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

2. Get data, etc. from Stata.

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j, k \neq B$$

Process step-by-step:

3. Derive N, T, J.

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

4. Loop over i using `panelsetup`

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j, k \neq B$$

Process step-by-step:

5. Compute $A = \sum_{j \neq B} \sum_t y_{itj} X_{it} \beta'_j$

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

6. At gradient-step (`if (todo>0)`), compute $C_{(j,m)} = \sum_t y_{itj} x_{itm}$

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

7. Loop over Δ_i (permutations of y_i) using `cvpermute`

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

8. Add up $B = \sum_{\Delta_i} \exp(\sum_{j \neq B} \sum_t d_{itj} X_{it} \beta'_j)$

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

9. At gradient-step (`if (todo>0)`), add up $D_{(j,m)} = \sum_{\Delta_i} \sum_t d_{itj} x_{itm} \exp(\sum_{j \neq B} \sum_t d_{itj} X_{it} \beta_j')$

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j, k \neq B$$

Process step-by-step:

10. At Hessian-step (`if (todo>1)`), add up $E_{(j,m)(k,l)} = \sum_{\Delta_i} \sum_t d_{itj} x_{itm} \sum_t d_{itk} x_{itl} \exp(\sum_{j \neq B} \sum_t d_{itj} X_{it} \beta_j')$

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

11. After loop over Δ_i, build panel-wise $\ln L_i, g_i, H_i$

MZES

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j,k \neq B$$

Process step-by-step:

12. After loop over i, build sample $\ln L, g, H$

Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function `cmlogit_eval()`

- Compute $\ln L, g, H$ with current coef. vector

$$\ln L = \sum_i (A - \ln B)$$

$$\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left(C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \qquad \text{for } j \neq B$$

$$\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left(\frac{D'_{(j,m)} D_{(k,l)}}{B^2} - \frac{E_{(j,m)(k,l)}}{B} \right) \qquad \text{for } j, k \neq B$$

Process step-by-step:

And that's it! (with one ml-step)

First applications: How to use it

Syntax

femlogit *depvar indepvars*, **group**(*varlist*) [**baseoutcome**(*#*)]

Data structure

- Long panel-wise, condensed alternative-wise:

i	t	y_{it}	x_{it}
1	1	1	.5
1	2	2	.2
1	3	3	.9
2	1	1	.1
2	2	2	.3
2	3	1	.2

- t not necessary.

Examples: Benchmark clogit

How precise and how fast is it?

Comparison with `clogit` for $J = 2$.

- Data used:
  `http://www.stata-press.com/data/r11/union.dta`
- Relative difference of coefficients: 9.078e-16.
- Speed: `clogit`: 2.42 sec., `femlogit`: 101.58 sec..

Examples: Simulated data

Performance with more alternatives

Simulated data

- N=1000, T=5, J=5
- Unobs. het. α_{ij}: over all i random draw $(\alpha_{i1}, \ldots, \alpha_{i5})$ from uniform distribution over 4-simplex Δ^4.
- Error ε_{itj}: over all i and t, for each j indep. draws from Gumbel-distribution (E$(\varepsilon_{itj}) = \gamma$,Var$(\varepsilon_{itj}) = \pi/\sqrt{6}$).
- Indep. variable: x correlated with α
  - $x_{it} = u_{it} + \alpha_{i2}$,
  - u_{it} drawn from uniform distribution.
- Coefficients $\beta_2 = 2, \beta_3 = 3, \beta_4 = 4, \beta_5 = 5$.

Examples: Simulated data (cont.)

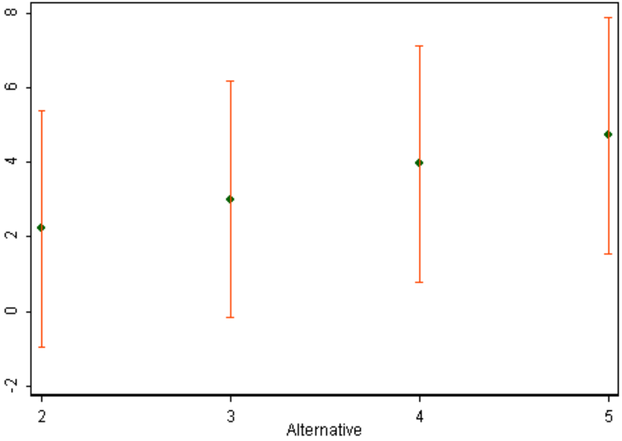
- Utility U_{itj}: for each i and t

$$
\begin{aligned}
U_{it1} &= \varepsilon_{it1} \\
U_{it2} &= 10\alpha_{i2} + \beta_2 x_{it} + \varepsilon_{it2} \\
&\vdots \\
U_{it5} &= 10\alpha_{i5} + \beta_5 x_{it} + \varepsilon_{it5}
\end{aligned}
$$

- Dep. var.: $y_{it} = j$ with $U_{itj} = \max_k(U_{itk})$

Examples: Simulated data (cont.)

Results



informative observations: N=3405; speed: 20.83 sec.

Outlook

Things to do

- "tomorrow"
  - Document and publish
- in near future
  - Add standard options (if/in-able, ml-options, etc.)
  - Think about special postestimation
  - Robust estimates
- in far future
  - Intuitive Interpretation
  - Nested logit with fixed effects
  - Parametric serial correlation
  - Implementation of RE-Models & Hausman-Test

Thank you!

Example 1: clogit

```
. clogit union age grade not_smsa south black, group(idcode)
note: multiple positive outcomes within groups encountered.
note: 2744 groups (14165 obs) dropped because of all positive or
      all negative outcomes.
note: black omitted because of no within-group variance.

Iteration 0:   log likelihood = -4521.3385
Iteration 1:   log likelihood = -4516.1404
Iteration 2:   log likelihood = -4516.1385
Iteration 3:   log likelihood = -4516.1385

Conditional (fixed-effects) logistic regression   Number of obs   =      12035
                                                  LR chi2(4)      =      68.09
                                                  Prob > chi2     =     0.0000
Log likelihood = -4516.1385                       Pseudo R2       =     0.0075

------------------------------------------------------------------------------
       union |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .0170301    .004146     4.11   0.000     .0089042    .0251561
       grade |   .0853572   .0418781     2.04   0.042     .0032777    .1674368
    not_smsa |   .0083678   .1127963     0.07   0.941    -.2127088    .2294445
       south |   -.748023   .1251752    -5.98   0.000    -.9933619   -.5026842
       black |  (omitted)
------------------------------------------------------------------------------
```

Example 2: femlogit

```
. femlogit union age grade not_smsa south black, group(idcode) b(0)

note: 2744 groups (14165 obs) dropped because of all positive or
      all negative outcomes.
note: black omitted because of no within-group variance.

Iteration 0:   log likelihood = -4521.3385
Iteration 1:   log likelihood = -4516.1404
Iteration 2:   log likelihood = -4516.1385
Iteration 3:   log likelihood = -4516.1385

                                                Number of obs     =      12035
                                                Wald chi2(4)      =          .
Log likelihood = -4516.1385                     Prob > chi2       =          .
```

union	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	.0170301	.004146	4.11	0.000	.0089042	.0251561
grade	.0853572	.0418781	2.04	0.042	.0032777	.1674368
not_smsa	.0083678	.1127963	0.07	0.941	-.2127088	.2294445
south	-.748023	.1251752	-5.98	0.000	-.9933619	-.5026842
black	(omitted)					