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 $\alpha_i$ necessary.
- Random effects and pooled models basically assume no correlation of $\alpha_i$ 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 \( \alpha_{ij} \)
- Cond. probability model: Prob. of sequence \( y_{i1}, \ldots, y_{iT_i} \) cond. of "overall tendency" to each outcome \( j \neq B \).
- \( \alpha_i \) disappears!

\[
\Pr(y_i \mid \bigwedge_{j \neq B} \sum_{t=1}^{T_i} y_{itj}) = \frac{\Pi_{t=1}^{T_i} \Pi_{j=1, j \neq B}^J \exp(X_{it}\beta_j')^{y_{itj}}}{\sum_{d_{itj} \in \Delta_i} \left( \Pi_{t=1}^{T_i} \Pi_{j=1, j \neq B}^J \exp(X_{it}\beta_j')^{d_{itj}} \right)}
\]

with

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

\[
\ln L = \sum_i \left( \sum_{j \neq B} \sum_t y_{itj} X_{it} \beta'_j - \ln \sum_{\Delta_i} \exp \sum_{j \neq B} \sum_t d_{itj} X_{it} \beta'_j \right)
\]
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 ⇒ no likelihood defined at observation level ⇒ d-family method
- Computation speed and accuracy ⇒ d2 method, i.e. \( \ln L, g, H \) have to be analytically derived
- J-1 equations, i.e.
  \[
  (y_1, \ldots, y_{J-1}) = (y_1, \ldots, y_{B-1}, y_{B+1}, \ldots, y_J)
  \]
- J-1 parameters \( \theta_j = X_{it} \beta'_j \); not used, direct use of \((J-1) \times M\) coefficients \( \beta_{jm} \)
Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function \texttt{cmlogit\_eval()}

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

Process step-by-step:
Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function \texttt{cmlogit_eval()}

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

Process step-by-step:

3. Derive \(N, T, J\)
"Core": Mata evaluator function `cmlogit_eval()`

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

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

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

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

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

Process step-by-step:

6. At gradient-step (if (todo>0)), compute $C_{(j,m)} = \sum_t y_{itj} x_{itm}$
"Core": Mata evaluator function \texttt{cmlogit\_eval()}

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

\[
\ln L = \sum_{i} \left( A - \ln B \right)
\]

\[
\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_{i} \left( C_{(j,m)} - \frac{D_{(j,m)}}{B} \right) \quad \text{for } j \neq B
\]

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

Process step-by-step:

7. Loop over $\Delta_{i}$ (permutations of $y_{i}$) using \texttt{cvpermute}
Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function cmlogit_eval()

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

Process step-by-step:

9. At gradient-step (if (todo>0)), add up

\[
D_{(j,m)} = \sum_{\Delta_i} \sum_t d_{ij} x_{itm} \exp(\sum_{j \neq B} \sum_t d_{itj} X_{it} \beta'_{j})
\]
Implementation: Maximum likelihood (cont.)

"Core": Mata evaluator function cmlogit_eval()

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

Process step-by-step:

10. At Hessian-step (if \( \text{todo} > 1 \)), add up

\[
E_{(j,m)(k,l)} = \sum_{\Delta_i} \sum_t d_{itj} x_{itm} \sum_t d_{itk} x_{itl} \exp(\sum_{j \neq B} \sum_t d_{itj} X_{it} \beta_j')
\]
"Core": Mata evaluator function cmlogit_eval()

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

Process step-by-step:

11. After loop over $\Delta_i$, build panel-wise $\ln L_i, g_i, H_i$
"Core": Mata evaluator function `cmlogit_eval()`

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

\[
\ln L = \sum_i (A - \ln B)
\]

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

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

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

\[
\ln L = \sum_i (A - \ln B)
\]

\[
\frac{\partial \ln L}{\partial \beta_{jm}} = \sum_i \left( C(j,m) - \frac{D(j,m)}{B} \right) \quad \text{for } j \neq B
\]

\[
\frac{\partial^2 \ln L}{\partial \beta_{jm} \partial \beta_{kl}} = \sum_i \left( \frac{D'(j,m)D(k,l)}{B^2} - \frac{E(j,m)(k,l)}{B} \right) \quad \text{for } j, k \neq B
\]

Process step-by-step:

And that’s it! (with one ml-step)
First applications: How to use it

Syntax

\texttt{femlogit} \texttt{depvar indepvars, group(varlist)} \texttt{[baseoutcome(\#)]}

Data structure

- Long panel-wise, condensed alternative-wise:

\begin{tabular}{|c|c|c|c|}
  \hline
  \textit{i} & \textit{t} & \textit{y}_{it} & \textit{x}_{it} \\
  \hline
  1 & 1 & 1 & .5 \\
  1 & 2 & 2 & .2 \\
  1 & 3 & 3 & .9 \\
  2 & 1 & 1 & .1 \\
  2 & 2 & 2 & .3 \\
  2 & 3 & 1 & .2 \\
  \hline
\end{tabular}

- \textit{t} not necessary.
Examples: Benchmark clogit

How precise and how fast is it?
Comparison with clogit for $J = 2$.

- Data used:
- Relative difference of coefficients: $9.078e-16$.
Examples: Simulated data

Performance with more alternatives
Simulated data

- N=1000, T=5, J=5
- Unobs. het. $\alpha_{ij}$: over all i random draw $(\alpha_{i1}, \ldots, \alpha_{i5})$ from uniform distribution over 4-simplex $\Delta^4$.
- Error $\varepsilon_{ij}$: over all i and t, for each j indep. draws from Gumbel-distribution ($E(\varepsilon_{ij}) = \gamma, \text{Var}(\varepsilon_{ij}) = \pi/\sqrt{6}$).
- Indep. variable: $x$ correlated with $\alpha$
  - $x_{it} = u_{it} + \alpha_{i2}$,
  - $u_{it}$ 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 $t$

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

- 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
Pseudo R2       =     0.0075

Log likelihood =  -4516.1385

                      Coef.  Std. Err.      z    P>|z|     [95% Conf. Interval]
union
  age             .0170301   .0041446     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_simsa 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)       =         .
Prob > chi2        =         .

Log likelihood = -4516.1385

```

| union          | Coef.  | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|----------------|--------|-----------|-------|------|----------------------|
| age            | .0170301| .0041446  | 4.11  | 0.000| .0089042 to .0251561 |
| grade          | .0853572| .0418781  | 2.04  | 0.042| .0032777 to .1674368 |
| not_simsa      | .0083678| .1127963  | 0.07  | 0.941| -.2127088 to .2294445 |
| south          | -.748023| .1251752  | -5.98 | 0.000| -.9933619 to -.5026842 |
| black          | (omitted) |          |       |      |                      |
