The Stata Journal

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press, and Stata is a registered trademark of StataCorp LP.

The Stata Journal (2006)
6, Number 2, pp. 229–245

Estimation of multinomial logit models with unobserved heterogeneity using maximum simulated likelihood

Peter Haan
DIW Berlin
Königin-Luise-Straße 5
14195 Berlin, Germany
phaan@diw.de

Arne Uhlendorff
DIW Berlin
Königin-Luise-Straße 5
14195 Berlin, Germany
auhlendorff@diw.de

Abstract. In this paper, we suggest a Stata routine for multinomial logit models with unobserved heterogeneity using maximum simulated likelihood based on Halton sequences. The purpose of this paper is twofold. First, we describe the technical implementation of the estimation routine and discuss its properties. Further, we compare our estimation routine with the Stata program `gllamm`, which solves integration by using Gauss–Hermite quadrature or adaptive quadrature. For the analysis, we draw on multilevel data about schooling. Our empirical findings show that the estimation techniques lead to approximately the same estimation results. The advantage of simulation over Gauss–Hermite quadrature is a marked reduction in computational time for integrals with higher dimensions. Adaptive quadrature leads to more stable results relative to the other integration methods. However, simulation is more time efficient. We find that maximum simulated likelihood leads to estimation results with reasonable accuracy in roughly half the time required when using adaptive quadrature.

Keywords: st0104, multinomial logit model, multinomial logistic model, panel data, unobserved heterogeneity, maximum simulated likelihood, Halton sequences

1 Introduction

In many empirical applications, e.g., estimation of mixed logit models, the researcher is faced with the problem that standard maximum-likelihood estimation cannot be applied, as analytical integration is not possible. Instead, methods such as quadrature or simulation are required for approximation of the integral. We suggest a Stata routine for multinomial logit models with unobserved heterogeneity using maximum simulated likelihood (MSL).[1] The purpose of this paper is twofold. First, we provide a description of the technical implementation of the estimation routine and discuss its properties. Further, we compare our estimation routine with the Stata program `gllamm`. `gllamm` is a flexible program incorporating a variety of multilevel models including mixed logit; see Rabe-Hesketh, Skrondal, and Pickles (2004) or Rabe-Hesketh and Skrondal (2005). Our routine differs from `gllamm` for computational reasons: whereas in `gllamm`, inte-

[1] Our approach closely follows that of Train (2003), who implemented a program for mixed logit models in GAUSS.

grals are solved by using classical Gauss–Hermite or adaptive quadrature, we suggest simulation based on Halton sequences for integration. In our analysis, we compare the performance of the estimation techniques using multilevel data about schooling from the `gllamm` manual.

Our empirical findings show that when the integral is reasonably well approximated the estimation techniques lead to nearly the same results. The advantage of Halton-based simulation over classical Gauss–Hermite quadrature is computational time; this advantage increases with the dimensions of the integral. Adaptive quadrature leads to more stable results relative to the other integration methods. However, again simulation is more time efficient. We find that maximum simulated likelihood leads to estimation results with reasonable accuracy in roughly half the time required when using adaptive quadrature.

In the next section, we provide a brief discussion about the estimation of multinomial logit models with unobserved heterogeneity using MSL. Hereafter, we present a description of the technical implementation of the estimation routine and discuss its properties. In section 4, we compare the performance of MSL with estimation based on classical and adaptive quadrature using multilevel data about schooling. The final section concludes.

2 Multinomial logit models with unobserved heterogeneity

Mixed logit models are a highly flexible class of models approximating any random utility model (Train 2003). In this application, we focus on a specific model of this broad class, the multinomial logit panel-data model with random intercepts.[2] The results we present can be generalized and extended to other mixed logit models both with panel and cross-sectional data.

The theoretical framework of multinomial logit models can be described as follows. Each individual i is faced with J different choices at time t. The individual receives a certain level of utility at each choice alternative and chooses the alternative that maximizes the utility. As well documented in the literature—e.g., Train (2003)—the probability of making choice j conditional on observed characteristics X_{it} that vary between individuals and over time and unobserved individual effects α_i that are time constant has the following form:

$$\Pr(j|X_{it},\alpha_i) = \frac{\exp(X_{it}\beta_j + \alpha_{ij})}{\Sigma_{k=1}^{J}\exp(X_{it}\beta_k + \alpha_{ik})}$$

As the choice probabilities are conditioned on α_i, one must integrate over the distribution of the unobserved heterogeneity. Thus the sample likelihood for the multinomial logit with random intercepts has the following form:

[2] We use panel data and multilevel data interchangeably.

$$L = \prod_{i=1}^{N} \int_{-\infty}^{\infty} \prod_{t=1}^{T} \prod_{j=1}^{J} \left\{ \frac{\exp(X_{it}\beta_j + \alpha_j)}{\Sigma_{k=1}^{J} \exp(X_{it}\beta_k + \alpha_k)} \right\}^{d_{ijt}} f(\alpha) d\alpha \tag{1}$$

where $d_{ijt} = 1$ if individual i chooses alternative j at time t and zero otherwise. The coefficient vector and the unobserved heterogeneity term of one category are set to 0 for identification of the model. For convenience, we assume throughout our analysis that the unobserved heterogeneity α is identically and independently distributed over the individuals and follows a multivariate normal distribution with mean a and variance–covariance matrix $\mathbf{W}$, $\alpha \sim f(a, W)$. In most applications, α is specified to be normally distributed. However, as Train (2003) points out, the distributional assumption depends on the research question; if more appropriate, distributions such as log-normal or uniform can be assumed. As standard in random-effects models, the unobserved heterogeneity α is required to be independent of the explanatory variables X_{it}.

To maximize the sample likelihood, one must integrate over the distribution of unobserved heterogeneity. Yet, there exists no analytical solution for the integral in (1). In the literature, many methods for integral approximation have been suggested and discussed. We focus on classical Gauss–Hermite quadrature, adaptive quadrature, and simulation based on Halton sequences.

Gauss–Hermite and adaptive quadrature

Gauss–Hermite and adaptive quadrature are discussed in detail in the work of Rabe-Hesketh, Skrondal, and Pickles (2002). Gauss–Hermite quadrature approximates an integral by a specified number of discrete points. Adaptive quadrature uses Bayes' rule to find quadrature weights that lead to better approximations of the integral than those of normal Gauss–Hermite quadrature, significantly increasing the accuracy of integration. The Stata program `gllamm` incorporates both integration methods, yet adaptive quadrature is strongly recommended for its higher accuracy (Rabe-Hesketh, Skrondal, and Pickles 2002).

Estimation with maximum simulated likelihood

We suggest integrating over the unobserved heterogeneity by using simulation and maximizing a simulated likelihood. MSL draws R values from the distribution of the unobserved heterogeneity with variance–covariance matrix $\mathbf{W}$. For each of these draws, the likelihood is calculated and then averaged over the R draws, which implies that instead of the exact likelihood, a simulated sample likelihood (SL) is maximized:[3]

[3] When using random draws, MSL is equivalent to the ML estimator if $N^{0.5}/R \rightarrow 0$ and both N and R$\rightarrow \infty$. For more detailed information, see Cameron and Trivedi (2005).

$$\text{SL} = \prod_{n=1}^{N} \frac{1}{R} \sum_{r=1}^{R} \prod_{t=1}^{T} \prod_{j=1}^{J} \left\{ \frac{\exp(X_{it}\beta_j + \alpha_j^r)}{\Sigma_{k=1}^{j} \exp(X_{it}\beta_k + \alpha_k^r)} \right\}^{d_{ijt}} \tag{2}$$

Consider an example with three different choices ($j = 3$). For identification, β_1 and α_{i1} are normalized to zero. We assume that the unobserved heterogeneity differs between the two other choices ($\alpha_{i2} \neq \alpha_{i3}$) and allow for correlation of these terms. Hence, the distribution of the unobserved heterogeneity can be described by a bivariate normal distribution with the following:

$$\alpha \sim f \left\{ \begin{pmatrix} a_2 \\ a_3 \end{pmatrix}, \begin{pmatrix} \text{var}_2 & \text{cov}_{23} \\ \text{cov}_{23} & \text{var}_3 \end{pmatrix} \right\}$$

This equation implies that when applying MSL, an approximate two-dimensional integral is needed. Each draw r consists of two values, $(\epsilon_2, \epsilon_3)'$, which follow a standard normal distribution. We apply a Cholesky decomposition of the variance–covariance matrix $\mathbf{W}$. A Cholesky factor $\mathbf{L}$ of matrix $\mathbf{W}$ is defined such that $\mathbf{LL}' = \mathbf{W}$. Then the unobserved effects α^r are calculated by $\alpha^r = \mathbf{L}\epsilon^r$, which for our example implies the following:

$$\begin{pmatrix} \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{pmatrix} \begin{pmatrix} \epsilon_2 \\ \epsilon_3 \end{pmatrix} \tag{3}$$

The example can be easily extended to more complex choice situations. However, with more choices, integration becomes more and more time intensive as the dimension of the integral increases.

Instead of using random draws to obtain $(\epsilon_2, \epsilon_3)'$, we follow Train (2003) and recommend basing simulation on Halton sequences. Halton sequences generate quasirandom draws that provide a more systematic coverage of the domain of integration than independent-random draws and induce a negative correlation over observations. Several studies such as Train (2000) and Bhat (2001) have shown that for mixed logit models, the accuracy can be markedly increased by using Halton sequences; the authors find in their studies that the results are more precise with 100 Halton draws than with 1,000 random draws. These results confirm that quasirandom sequences go along with a lower integration error and faster convergence rates and therefore clearly require fewer draws than pseudorandom sequences.[4] However, as Train (2003) points out, using Halton draws in simulation-based estimation is not completely understood and caution is required. He provides an example of Halton sequences and discusses advantages and anomalies of this method for mixed logit models. Computational time and estimation

[4] The expected integration error using pseudorandom sequences is of order $R^{-.5}$, whereas the theoretical upper bound for the integration error using quasirandom sequences is of order R^{-1}; see Bhat (2001) or Cameron and Trivedi (2005). This comparison implies that a 10-fold increase in the number of quasirandom draws leads to the same improvement of accuracy as a 100-fold increase in the number of pseudorandom draws.

results slightly vary with the chosen primes for the Halton draws. This fact is documented by Train (2003), who found that the choice of the primes might noticeably affect the estimated coefficients.

The advantages of Halton draws might not hold for other models in the same way; see Cappellari and Jenkins (2006a), who discuss Halton sequences for multivariate probit models.

3 Stata routine for MSL estimation

Here we provide an `ml model` statement that refers to a multinomial logit panel-data model with two potentially correlated random intercepts that follow a bivariate normal distribution. This example can easily be extended to models with more alternatives.

For illustration, we apply our program to a real dataset about teachers' evaluations of pupil behavior.[5] The variables `id` and `scy3` identify pupils and schools, respectively. Teachers group pupils in three different quality levels (`tby`), which is the dependent variable in our estimation. The data contain several additional variables explaining the quality level of the pupils, such as sex, and provide information about 1,313 pupils in 48 schools. The number of pupils differs between schools; i.e., we have an unbalanced panel.

The panel dimension of the data is not over time but over the pupils of a certain school (`scy3`). Hence, in the estimation, we can control for unobserved school-specific effects, but we do not control for individual-specific unobserved heterogeneity.[6] For simplicity, we condition the rating of teachers next to unobservable effects on only one observable variable, namely, `sex`.

Before executing our program for MSL estimation, we apply the program `mdraws` by Cappellari and Jenkins (2006a) to generate Halton sequences and calculate the corresponding values following a standard normal distribution. `mdraws` can also be used to create pseudonormal draws.

For each draw, the values (`random_1‘r’` and `random_2‘r’`) must be the same for 1 observation within each unit, here within each school. Therefore, we create draws for every school and merge these draws to every pupil within each school. Here we approximate the integral by using 50 draws from the Halton sequence. We specify the primes used to create the Halton sequences as 7 and 11, because we later fit models with 150 draws and the number of draws should not be an integer multiple of any of the primes used. See Cappellari and Jenkins (2006a) for details. We use the `burn()` option to drop the first 15 draws of each sequence because the initial elements of any two sequences can be highly correlated.

[5] The dataset is available as an ASCII file, `jspmix.dat` (http://www.gllamm.org/jspmix.dat).

[6] The presented routine can easily be transferred to a model with time-constant individual-specific effects. Here the school (`scy3`) corresponds to the individual and one pupil to one individual observation at time t.

```
. matrix p = (7, 11)
. global draws "50"
. infile scy3 id sex stag ravi fry3 tby using jspmix.dat, clear
(1313 observations read)
. save jspmix.dta, replace
(note: file jspmix.dta not found)
file jspmix.dta saved
. keep scy3
. sort scy3
. by scy3: keep if _n==1
(1265 observations deleted)
. mdraws, neq(2) dr($draws) prefix(c) burn(15) prime(p)
Created 50 Halton draws per equation for 2 dimensions. Number of initial
draws dropped per dimension = 15 . Primes used:
   7  11
. forvalues r=1/$draws{
  2.   gen random_1`r'=invnormal(c1_`r')
  3.   gen random_2`r'=invnormal(c2_`r')
  4. }
. sort scy3
. save mdraws_$draws, replace
(note: file mdraws_50.dta not found)
file mdraws_50.dta saved
. use jspmix, clear
. sort scy3
. merge scy3 using mdraws_$draws.dta
variable scy3 does not uniquely identify observations in the master data
. drop _merge
. sort scy3
```

To get appropriate starting values for the coefficient vector, we use `mlogit` to fit a multinomial logit model without random intercepts. The variables `a1`, `a2`, and `a3` take on the value of 1 if the choice 1, 2, or 3 is made, respectively, and zero otherwise; the variables are defined using the `tabulate` command.

```
. mlogit tby sex, base(1)
  (output omitted)
. matrix Init= e(b)
. tabulate tby, gen(a)
  (output omitted)
. sort scy3
```

The following `ml model` statement can be applied independently of the chosen type of draws (e.g., pseudorandom or Halton). We apply the method `d0` because we fit panel-data models with joint unobserved heterogeneity for groups of observations. The method `d0` requires the researcher to supply the log-likelihood function. The first and second derivatives are obtained numerically; i.e., one need not supply analytically calculations of the gradient and the Hessian of the log-likelihood function.[7]

[7] The principles of computing maximum likelihood estimators with Stata are described in Gould, Pitblado, and Sribney (2006).

```
program define mlogit_sim_d0
  args todo b lnf
  tempvar etha2 etha3 random1 random2 lj pi1 pi2 pi3 sum lnpi L1 L2 last
  tempname lnsig1 lnsig2 atrho12 sigma1 sigma2 cov12

  mleval ‘etha2’   = ‘b’, eq(1)
  mleval ‘etha3’   = ‘b’, eq(2)
  mleval ‘lnsig1’  = ‘b’, eq(3) scalar
  mleval ‘lnsig2’  = ‘b’, eq(4) scalar
  mleval ‘atrho12’ = ‘b’, eq(5) scalar

  qui {
    scalar ‘sigma1’=(exp(‘lnsig1’))^2
    scalar ‘sigma2’=(exp(‘lnsig2’))^2
    scalar ‘cov12’=[exp(2*‘atrho12’)-1]/[exp(2*‘atrho12’)+1]* ///
       (exp(‘lnsig2’))*(exp(‘lnsig1’))
    gen double ‘random1’ = 0
    gen double ‘random2’ = 0
    gen double ‘lnpi’=0
    gen double ‘sum’=0
    gen double ‘L1’=0
    gen double ‘L2’=0
    by scy3: gen byte ‘last’=(_n==_N)
    gen double ‘pi1’=0
    gen double ‘pi2’=0
    gen double ‘pi3’=0
  }
  matrix W = ( ‘sigma1’ , ‘cov12’ \ ‘cov12’ , ‘sigma2’)

  capture matrix L=cholesky(W)

  if _rc != 0 {
        di "Warning: cannot do Cholesky factorization of rho matrix"
    }

  local l11=L[1,1]
  local l21=L[2,1]
  local l22=L[2,2]

  forvalues r=1/$draws{
   qui {
    replace ‘random1’ = random_1‘r’*‘l11’
    replace ‘random2’ = random_2‘r’*‘l22’ + random_1‘r’*‘l21’

    replace ‘pi1’= 1/(1 + exp(‘etha2’+‘random1’)+exp(‘etha3’+‘random2’))
    replace ‘pi2’= exp(‘etha2’+‘random1’)*‘pi1’
    replace ‘pi3’= exp(‘etha3’+‘random2’)*‘pi1’

    replace ‘lnpi’=ln(‘pi1’*a1+‘pi2’*a2+‘pi3’*a3)

    by scy3: replace ‘sum’=sum(‘lnpi’)
    by scy3: replace ‘L1’ =exp(‘sum’[_N]) if _n==_N

    by scy3: replace ‘L2’=‘L2’+‘L1’ if _n==_N
   }
  }

  qui gen ‘lj’=cond(!‘last’,0, ln(‘L2’/‘$draws’))
  qui mlsum ‘lnf’=‘lj’
  if (‘todo’==0|‘lnf’>=.) exit

end
```

Instead of estimating the variances and the correlation coefficient directly, we estimate transformed variables of these parameters, i.e., the logarithm of the standard deviations (`lnsig1` and `lnsig2`) and the inverse hyperbolic tangent of ρ (`atrho12`), to constrain them within their valid limits. Therefore, the first step in our program is to calculate the variances (`sigma1` and `sigma2`) and the covariance (`cov12`) of the bivariate normal distribution. Then we apply a Cholesky decomposition of the covariance matrix $\mathbf{W}$. To do this, the matrix $\mathbf{W}$ must be positive definite at each iteration. If not, our program traps the error, shows a warning, and uses the most recent estimate of $\mathbf{W}$, which is guaranteed to be positive definite. This assurance is based on the `capture` command.[8]

We calculate the likelihood for each draw from the individual-specific quasirandom terms `random1` and `random2` within the following loop. The two terms `random1_‘r’` and `random2_‘r’` are multiplied with the elements of the Cholesky matrix $\mathbf{L}$, following (3). The probabilities of making choice 1, 2, or 3 are expressed by `pi1`, `pi2`, and `pi3`. With the information about the realized choices, captured in variables `a1`, `a2`, and `a3`, the likelihood is evaluated for each observation. The corresponding log-likelihood values are added up within each unit for each draw (sum) and this sum is exponentiated for the last observation per unit (`L1`). These likelihood values are added up over all draws (`L2`). Following (2), the approximated likelihood is the average over the r draws. The simulated likelihood can be maximized by using the `ml maximize` and `ml model` commands. To set the starting values, we use the command `ml init`. For the β, we use the estimated coefficients from the `mlogit` saved as matrix `Init`. The starting values of `lnsig1`, `lnsig2`, and `atrho12` are set to 0.5.

```
. ml model d0 mlogit_sim_d0 (tby = sex) (tby = sex)  /lnsig1 /lnsig2 /atsig12
. matrix start = (Init)
. ml init start 0.5 0.5 0.5, copy
```

(*Continued on next page*)

[8] The procedure is the same as in the program `mvprobit` by Cappellari and Jenkins (2003, 2005, 2006b).

```
. ml maximize
initial:       log likelihood = -1338.0475
rescale:       log likelihood = -1338.0475
rescale eq:    log likelihood = -1301.4639
Iteration 0:   log likelihood = -1301.4639
Iteration 1:   log likelihood = -1300.4893
Iteration 2:   log likelihood = -1299.4587
Iteration 3:   log likelihood = -1299.4509
Iteration 4:   log likelihood = -1299.4509
                                                  Number of obs   =       1313
                                                  Wald chi2(1)    =      14.22
Log likelihood = -1299.4509                       Prob > chi2     =     0.0002

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
eq1          |
         sex |   .5488225     .14552     3.77   0.000     .2636085    .8340364
       _cons |     .59589   .1394991     4.27   0.000     .3224768    .8693032
-------------+----------------------------------------------------------------
eq2          |
         sex |   1.104577   .1748037     6.32   0.000     .7619681    1.447186
       _cons |  -.5663381   .1816152    -3.12   0.002    -.9222974   -.2103788
-------------+----------------------------------------------------------------
lnsig1       |
       _cons |  -.3369519   .1695314    -1.99   0.047    -.6692274   -.0046763
-------------+----------------------------------------------------------------
lnsig2       |
       _cons |  -.1021489   .1602249    -0.64   0.524    -.4161839    .2118861
-------------+----------------------------------------------------------------
atsig12      |
       _cons |   1.614593   .3185383     5.07   0.000     .9902697    2.238917
------------------------------------------------------------------------------
. _diparm lnsig1, function((exp(@))^2) deriv(2*(exp(@))*(exp(@)))
> label("sigma1")
      sigma1 |   .5097149   .1728254                      .2622506    .9906909
. _diparm lnsig2, function((exp(@))^2) deriv(2*(exp(@))*(exp(@)))
> label("sigma2")
      sigma2 |   .8152196   .2612369                       .435018    1.527713
. _diparm atsig12, tanh label("roh12")
       roh12 |   .9238359   .0466745                      .7574773    .9775391
. _diparm atsig12 lnsig1 lnsig2,
> function([exp(2*@1)-1]/[exp(2*@1)+1]*(exp(@2))*(exp(@3)))
> deriv(-(2*exp(2*@1+@2+@3)*(-1+exp(2*@1))/(1+exp(2*@1))^2)+
>        2*exp(2*@1+@2+@3)/(1+exp(2*@1))
>        [exp(2*@1)-1]/[exp(2*@1)+1]*(exp(@2))*(exp(@3))
>        [exp(2*@1)-1]/[exp(2*@1)+1]*(exp(@2))*(exp(@3))) label("cov12")
       cov12 |   .5955193    .188545                      .2259779    .9650606
```

As mentioned above, we estimate the variances and the covariance in a transformed metric. We use the program _diparm to calculate and display the parameters and their standard errors after the estimation. For this task, we must calculate the first derivative of the function. Also we can use _diparm to calculate the correlation and its standard error.

4 Illustrations

In the following section, we discuss the empirical performance of the MSL routine with a multilevel dataset about schooling (Junior School Project) that is taken from the `gllamm` manual (Rabe-Hesketh, Skrondal, and Pickles 2004). We described the data in the previous section. This example provides a comparison of the above-described integration methods, Gauss–Hermite and adaptive quadrature using `gllamm` and simulation based on Halton draws using our MSL routine. We are interested in two findings: 1) the accuracy of the procedures, evaluated for the stability of estimation results and 2) the computational time they require. Further, we want to show how the two estimators perform when the dimension of the integrals increases. Therefore, we fit models with only one random term (one-dimensional integral) and with two random terms (two-dimensional integral). One random term implies that unobserved effects are constant between the alternatives. In the second example (two random terms), the heterogeneity varies between the alternatives and is potentially correlated. The structure of unobserved heterogeneity is the same as that in section 2's example.

Computational time and accuracy of integral approximation depend on the chosen number of quadrature points or number of draws when estimating. Therefore, we present several estimations by increasing the number of quadrature points and draws. As there is a tradeoff between accuracy of integration and computational time, the number of points or draws can become a crucial variable. Providing a rigid test indicating the optimal number of draws is difficult. In practice, researchers often vary the number of draws or points to see whether the coefficients and the log likelihood remain constant as an indication whether an adequate number of draws is chosen (Cameron and Trivedi 2005). We present results of six estimations using MSL with 25, 50, 100, 150, 200, and 500 draws from the Halton sequences and six estimations with Gauss–Hermite and adaptive quadrature, both with 4, 8, and 16 points.[9] Because we do not directly test for accuracy, the comparison needs to be interpreted carefully. All estimates were computed with Intercooled Stata version 8.2 on a 3-GHz Pentium 4 PC running Windows 2000 Professional. To make computational time between both methods comparable, we used the same starting values for all estimations.

In the following code, we present the `gllamm` command for estimation of the model with the two-dimensional integral using four quadrature points (Gauss–Hermite). For more description of the syntax, see Rabe-Hesketh, Skrondal, and Pickles (2004).

```
use jspmix, clear
mlogit tby sex, base(1)
matrix Init = e(b)
scalar var = exp(0.5)
matrix start = Init, var, var, 0.5
matrix colnames start = sex _cons sex _cons a2 a3 _cons
matrix coleq start = c2 c2  c3 c3  scy1_1  scy1_2 scy1_2_1
gen school = scy3
sort school sex tby
```

[9] In addition to these results, we fitted the model using MSL based on pseudorandom draws. Our results are in line with previous studies, e.g., Train (2000) and Bhat (2001), and indicate that many more pseudorandom draws than Halton draws are required to get relatively stable results.

```
gen patt = _n
expand 3
sort patt
qui by patt: gen alt = _n
gen chosen = alt == tby
sort pat alt
tabulate alt, gen(a)
gen dum=1
replace dum = 0 if a1 == 1
eq dum: dum
eq a2: a2
eq a3: a3

gllamm alt sex, expand(patt chosen m) i(scy3) link(mlogit) /*
*/family(binom) nrf(2) eq(a2 a3) nip(4) trace from(start)
```

Table 1 shows the MSL results for the model with a common term of unobserved heterogeneity. Comparing the coefficients and the log likelihood between the estimations, we find that the results are fairly stable when using at least 50 draws. When using only 25 Halton draws, the deviations of the coefficients from those obtained with better approximated integrals can be seen. However, even with more than 100 draws, we find that results slightly differ between the number of draws; the log likelihood varies between the estimations in the first decimal place. Estimation time varies between the estimations with an acceptable approximation of the integral from 42 seconds (50 draws) to 8 minutes 18 seconds (500 draws); estimation results suggest that computational time increases approximately linearly with the number of draws.

(*Continued on next page*)

Table 1: One random intercept: Maximum simulated likelihood

Parameter	Coefficient (SE)					
`tby` = 2						
Sex	0.543 (0.146)	0.551 (0.146)	0.549 (0.146)	0.550 (0.146)	0.550 (0.146)	0.550 (0.146)
Constant	0.685 (0.141)	0.598 (0.145)	0.592 (0.146)	0.592 (0.145)	0.592 (0.146)	0.591 (0.145)
`tby` = 3						
Sex	1.064 (0.141)	1.072 (0.145)	1.070 (0.146)	1.071 (0.145)	1.071 (0.146)	1.070 (0.145)
Constant	−0.399 (0.160)	−0.486 (0.163)	−0.492 (0.164)	−0.492 (0.164)	−0.492 (0.164)	−0.493 (0.164)
`lnsig1`	−0.391 (0.146)	−0.289 (0.154)	−0.301 (0.155)	−0.301 (0.159)	−0.321 (0.163)	−0.312 (0.162)
`sig1`	0.457 (0.133)	0.561 (0.172)	0.547 (0.170)	0.548 (0.174)	0.526 (0.172)	0.536 (0.173)
Log likelihood	−1,303.791	−1,303.605	−1,303.751	−1,303.937	−1,303.658	−1,303.740
Time (hh:mm:ss)	00:00:20	00:00:42	00:01:26	00:02:10	00:03:05	00:08:18
No. of draws	25	50	100	150	200	500

Source: http://www.gllamm.org/jspmix.dat.
Note: Numbers of observations: 1,313.

Table 2 compares one random intercept calculated with both Gauss–Hermite and adaptive quadrature. Comparing the results derived with simulation with those estimated with quadrature, we find that the estimation results are similar when the integral is reasonably well approximated. When using Gauss–Hermite quadrature, at least eight quadrature points are required for integration. The log likelihood and the coefficients clearly differ between the estimation with four and eight points.

Turning to the adaptive quadrature, the picture changes. With only four quadrature points, the integral seems to be reasonably well approximated, as a further increase in the number of quadrature points leads to similar estimated parameters. This finding underscores the result of Rabe-Hesketh, Skrondal, and Pickles (2002), who show the computational advantage of adaptive quadrature versus Gauss–Hermite quadrature.

Table 2: One random intercept: Gauss–Hermite and adaptive quadrature

	Gauss–Hermite quadrature			Adaptive quadrature		
	Coefficient			Coefficient		
Parameter	(SE)			(SE)		
`tby` = 2						
	0.553	0.554	0.549	0.550	0.550	0.550
Sex	(0.146)	(0.146)	(0.146)	(0.146)	(0.146)	(0.146)
	0.693	0.619	0.593	0.594	0.594	0.594
Constant	(0.146)	(0.155)	(0.147)	(0.145)	(0.146)	(0.146)
`tby` = 3						
	1.074	1.075	1.070	1.071	1.071	1.071
Sex	(0.171)	(0.171)	(0.171)	(0.171)	(0.171)	(0.171)
	−0.391	−0.465	−0.492	−0.490	−0.491	−0.491
Constant	(0.165)	(0.172)	(0.166)	(0.163)	(0.164)	(0.164)
	0.398	0.564	0.530	0.551	0.543	0.544
`sig1`	(0.101)	(0.181)	(0.166)	(0.178)	(0.175)	(0.175)
Log likelihood	−1,305.189	−1,303.681	−1,303.843	−1,303.802	−1,303.804	−1,303.804
Time (hh:mm:ss)	00:00:21	00:00:46	00:01:10	00:01:24	00:01:42	00:03:12
No. of quadrature points	4	8	16	4 (Adaptive)	8 (Adaptive)	16 (Adaptive)

Source: http://www.gllamm.org/jspmix.dat.
Note: Numbers of observations: 1,313.

For the one-dimensional integral, Halton-based simulation performs similarly to quadrature. Relative to Gauss–Hermite quadrature, we find hardly any difference in computational time for a comparable degree of accuracy. Adaptive quadrature leads to more stable results with four quadrature points; computation time, however, is higher than in a simulation with 50 draws and about the same as in a simulation with 100 draws. This finding indicates that with one term there is no advantage of using MSL relative to adaptive quadrature.

In the following discussion, the complexity of the estimation increases by allowing the unobserved heterogeneity to differ between the alternatives. Here the advantage of computational time of Halton-based simulation over Gauss–Hermite quadrature becomes evident. As shown in table 3, with at least 100 draws, coefficients and the log likelihood become relatively stable. For 100 draws, the estimation takes more than 3 minutes. Table 4 shows that, for a comparable level of integral approximation, Gauss–Hermite quadrature requires more than 11.5 minutes. Results from MSL become more stable with 200 and 500 draws. The estimation with 200 draws takes less than 7 minutes, and the one with 500 draws about 20 minutes. When doubling the number of quadrature points for the Gauss–Hermite approach, computational time approximately quadruples (50 minutes) and the results are similar to those from the adaptive quadrature.

Table 3: Two random intercepts: Maximum simulated likelihood

Parameter	Coefficient (SE)					
`tby` = 2						
Sex	0.542 (0.145)	0.549 (0.146)	0.546 (0.146)	0.545 (0.146)	0.546 (0.146)	0.546 (0.146)
Constant	0.616 (0.142)	0.596 (0.139)	0.577 (0.144)	0.601 (0.140)	0.576 (0.142)	0.593 (0.141)
`tby` = 3						
Sex	1.095 (0.175)	1.105 (0.175)	1.099 (0.175)	1.102 (0.175)	1.101 (0.175)	1.101 (0.175)
Constant	−0.534 (0.184)	−0.566 (0.182)	−0.585 (0.178)	−0.563 (0.180)	−0.585 (0.181)	−0.569 (0.180)
`lnsig1`	−0.367 (0.201)	−0.337 (0.170)	−0.327 (0.174)	−0.366 (0.167)	−0.362 (0.175)	−0.361 (0.171)
`lnsig2`	−0.153 (0.167)	−0.102 (0.160)	−0.145 (0.158)	−0.142 (0.154)	−0.162 (0.163)	−0.158 (0.161)
`atrho`	1.535 (0.422)	1.615 (0.319)	1.471 (0.320)	1.550 (0.339)	1.487 (0.353)	1.496 (0.346)
`sig1`	0.479 (0.192)	0.510 (0.173)	0.520 (0.181)	0.481 (0.160)	0.484 (0.170)	0.485 (0.166)
`sig2`	0.735 (0.246)	0.815 (0.261)	0.749 (0.236)	0.753 (0.231)	0.724 (0.236)	0.729 (0.234)
`cov12`	0.54 (0.185)	0.596 (0.189)	0.561 (0.184)	0.550 (0.172)	0.535 (0.181)	0.538 (0.177)
`cor`	0.911 (0.071)	0.924 (0.047)	0.900 (0.061)	0.914 (0.056)	0.903 (0.065)	0.904 (0.063)
Log likelihood	−1,299.9	−1,299.451	−1,299.700	−1,299.635	−1,299.726	−1,299.599
Time (hh:mm:ss)	00:00:45	00:01:34	00:03:23	00:05:00	00:06:52	00:19:54
No. of draws	25	50	100	150	200	500

Source: http://www.gllamm.org/jspmix.dat.
Note: Numbers of observations: 1,313.

Table 4: Two random intercepts: Gauss–Hermite and adaptive quadrature

	Gauss–Hermite quadrature			Adaptive quadrature		
Parameter	Coefficient (SE)			Coefficient (SE)		
`tby` = 2						
Sex	0.548 (0.145)	0.551 (0.146)	0.546 (0.146)	0.547 (0.146)	0.546 (0.146)	0.546 (0.146)
Constant	0.668 (0.142)	0.621 (0.142)	0.595 (0.141)	0.598 (0.140)	0.597 (0.141)	0.597 (0.141)
`tby` = 3						
Sex	1.104 (0.175)	1.105 (0.175)	1.101 (0.175)	1.102 (0.175)	1.101 (0.175)	1.101 (0.175)
Constant	−0.480 (0.181)	−0.539 (0.181)	−0.567 (0.181)	0.564 (0.180)	−0.565 (0.180)	−0.565 (0.180)
`sig1`	0.352 (0.098)	0.504 (0.169)	0.480 (0.168)	0.489 (0.171)	0.488 (0.170)	0.488 (0.170)
`sig2`	0.596 (0.169)	0.752 (0.238)	0.730 (0.234)	0.743 (0.240)	0.739 (0.238)	0.738 (0.238)
`cov`	0.406 (0.108)	0.560 (0.180)	0.537 (0.177)	0.547 (0.182)	0.545 (0.181)	0.545 (0.181)
`cor`	0.887 —	0.910 —	0.907 —	0.908 —	0.908 —	0.908 —
Log likelihood	−1,300.950	−1,299.482	−1,299.681	−1,299.663	−1,299.664	−1,299.665
Time (hh:mm:ss)	00:02:47	00:11:38	00:47:41	00:08:16	00:30:38	02:03:12
No. of quadrature points	4	8	16	4 (Adaptive)	8 (Adaptive)	16 (Adaptive)

Note: Numbers of observations: 1,313. — = not calculated.
Source: http://www.gllamm.org/jspmix.dat.

With adaptive quadrature, again four points are sufficient to approximate the integral. Results hardly change with more quadrature points. Computational time with four points is about 8 minutes. Relative to simulation, adaptive quadrature leads to more robust results. However, using simulation with 100 draws, one can approximate the integral such that coefficients and the log likelihood are approximately stable in less than 3.5 minutes. Here the tradeoff between computational time and accuracy becomes evident. Halton-based simulation leads to results in less computational time, whereas adaptive quadrature provides results that are more stable.

From a practical point of view, the implementation of MSL based on Halton sequences is relatively simple and has significant advantages in computational time if it is compared with Gauss–Hermite quadrature and simulation based on pseudorandom sequences, not reported here. This implementation is particularly true for higher-dimensional integrals. Compared with adaptive quadrature, our routine seems to be less stable. However, given the advantage of computational time, Halton-based MSL could be an adequate model choice. The time advantage becomes even more important when sample size or the dimension of the integral increases.[10]

Therefore, we recommend the presented routine as an alternative to the quadrature approach implemented in `gllamm`. Moreover, the principles of our routine can be a useful starting point for evaluating likelihood functions that are not preprogrammed in Stata and involve a multivariate normal distribution of the unobserved heterogeneity.

5 Conclusion

In this article, we have suggested a Stata routine for multinomial logit models with unobserved heterogeneity using MSL based on Halton sequences. The routine refers to a model with two random intercepts but can easily be extended to models with a higher dimension. Further extensions of the presented code are possible; examples are Haan (2005), fitting a dynamic conditional logit model, or Uhlendorff (2006), fitting a dynamic multinomial logit model with endogenous panel attrition.

Using multilevel data about schooling, we compare the performance of our code to that of the Stata program `gllamm`, which numerically approximates integrals using classical Gauss–Hermite quadrature and adaptive quadrature. Estimation by MSL provides approximately the same estimation results as estimation with Gauss–Hermite quadrature or adaptive quadrature. Compared with classical quadrature, simulation markedly reduces computational time when a higher-dimensional integral needs to be approximated. However, relative to adaptive quadrature, the advantage of simulation vanishes in our example. Adaptive quadrature leads to stable results with only a few quadrature points (four). Estimations with 100 draws are less stable but lead to qualitatively the same results and take roughly half the estimation time. This finding underscores the tradeoff between computational time and accuracy of the results, which becomes important if estimation takes not a few minutes but instead hours or days.

6 Acknowledgments

This project is funded by the German Science Foundation (DFG) in the priority program "Potentials for More Flexibility on Heterogeneous Labor Markets" (projects STE 681/5-1 and ZI 302/7-1).

We thank David Drukker, Stephen Jenkins, Oswald Haan, Katharina Wrohlich, Dirk Hofmann, and an anonymous referee for valuable comments and suggestions.

[10] Using Bayes' rule for simulation might be one way to reduce the tradeoff between estimation time and accuracy. Train (2003) suggests using Bayesian simulation instead of classical MSL, as the Bayesian method leads to consistent estimates even with a fixed number of draws.

7 References

Bhat, C. R. 2001. Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model. *Transportation Research, Part B* 35: 677–693.

Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.

Cappellari, L., and S. P. Jenkins. 2003. Multivariate probit regression using simulated maximum likelihood. *Stata Journal* 3: 278–294.

———. 2005. Software update: st0045_1: Multivariate probit regression using simulated maximum likelihood. *Stata Journal* 5: 285.

———. 2006a. Calculation of multivariate normal probabilities by simulation, with applications to maximum simulated likelihood estimation. *Stata Journal* 6: 156–189.

———. 2006b. Software update: st0045_2: Multivariate probit regression using simulated maximum likelihood. *Stata Journal* 6: 284.

Gould, W., J. Pitblado, and W. Sribney. 2006. *Maximum Likelihood Estimation with Stata*. 3rd ed. College Station, TX: Stata Press.

Haan, P. 2005. State dependence and female labor supply in Germany: The extensive and the intensive margin. DIW Discussion Paper no. 538. Downloadable from http://www.diw.de/deutsch/produkte/publikationen/diskussionspapiere/docs/papers/dp538.pdf.

Rabe-Hesketh, S., and A. Skrondal. 2005. *Multilevel and Longitudinal Modeling Using Stata*. College Station, TX: Stata Press.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2002. Reliable estimation of generalized linear mixed models using adaptive quadrature. *Stata Journal* 2: 1–21.

———. 2004. *GLLAMM Manual*. University of California–Berkeley, Division of Biostatistics, Working Paper Series. Paper No. 160.
http://www.bepress.com/ucbbiostat/paper160/.

Train, K. 2000. Halton sequences for mixed logit. Economics Department, University of California, Berkeley, Working Paper E00-278.
http://repositories.cdlib.org/iber/econ/E00-278/.

———. 2003. *Discrete Choice Methods with Simulation*. Cambridge: Cambridge University Press.

Uhlendorff, A. 2006. From no pay to low pay and back again? Low pay dynamic in West Germany. Mimeo.

About the authors

Peter Haan and Arne Uhlendorff are research associates at DIW Berlin, Berlin, Germany, and PhD students at Free University, Berlin, Germany. Arne Uhlendorff is also a research affiliate at IZA Bonn, Bonn, Germany.