bmp_description

February 23, 2021

1 Using `bayesmixedlogit` and `bayesmixedlogitwtp` in Stata

This document presents an overview of the `bayesmixedlogit` and `bayesmixedlogitwtp` Stata packages. It mirrors closely the helpfile obtainable in `stata` (i.e., through `help bayesmixedlogit` or `help bayesmixedlogitwtp`). Further background for the packages can be found in Baker(2014).

1.0.1 Description

`bayesmixedlogit` can be used to fit mixed logit models using Bayesian methods – more precisely, `bayesmixedlogit` produces draws from the posterior parameter distribution and then presents summary and other statistics describing the results of the drawing. Detailed analysis of the draws is left to the discretion of the user.

Implementation of bayesmixedlogit follows Train (2009, chap. 12), and details of how the algorithm works are described in Baker(2014). A diffuse prior for the mean values of the random coefficients is assumed, and the prior distribution on the covariance matrix of random coefficients is taken to be an identity inverse Wishart. `bayesmixedlogit` uses the `Mata` routines `amcmc()` (if not installed, search online; from the `Stata` command-line prompt, type `ssc install amcmc`) for adaptive Markov chain Monte Carlo sampling from the posterior distribution of individual level coefficients and fixed coefficients. The data setup for `bayesmixedlogit` is the same as for `clogit` (`ssc install clogit`). Much of the syntax follows that used by Hole (2007) in development of the command `mixlogit`.

1.0.2 Options

`group(varname)` specifies a numeric identifier variable for choice occasions. `group()` is required.

`identifier(varname)` identifies coefficient sets (those observations for which a set of coefficients apply). Thus, when a person is observed making choices over multiple occasions, one would use `group(`*`varname`*`)` to specify the choice occasions, while `identifier(varname)` would identify the person. `identifier()` is required.

`rand(`*`varlist`*`)` specifies independent variables with random coefficients. The variables immediately following the dependent variable in the syntax are considered to have fixed coefficients (see the examples below). While a model can be run without any independent variables with fixed coefficients, at least one random-coefficient independent variable is required for bayesmixedlogit to work. `rand()` is required.

`draws(#)` specifies the number of draws that are to be taken from the posterior distribution of the parameters. The default is `draws(1000)`.

drawsrandom(#) is an advanced option. The drawing algorithm treats each set of random coefficients as a Gibbs step in sampling from the joint posterior distribution of parameters. In difficult, large-dimensional problems, it might be desirable to let individual Gibbs steps run for more than one draw to achieve better mixing and convergence of the algorithm.

drawsfixed(#) is a more advanced option. The drawing algorithm treats fixed coefficients as a Gibbs step in sampling from the joint posterior distribution of parameters. In difficult, large-dimensional problems, it might be desirable to let this step in Gibbs sampling run for more than a single draw. The default is drawsfixed(1).

```
burn(#) specifies the length of the burn-in period; the first # draws are discarded upon complet

thin(#) specifies that only every #th draw is to be retained, so if thin(3) is specified, only e
    which randomly mixes draws.  Both options may be applied.

araterandom(#) specifies the desired acceptance rate for random coefficients and should be a num
    desired acceptance rate, the user has some control over adaptation of the algorithm to the p

aratefixed(#) specifies the desired acceptance rate for fixed coefficients and works in the same

samplerrandom(string) specifies the type of sampler that is to be used when random parameters ar
    parameters are drawn all at once.  If mwg -- an acronym for "Metropolis within Gibbs" -- is
    Gibbs step.  The default is samplerrandom(global), but mwg might be useful in situations in

samplerfixed(string) specifies the type of sampler that is used when fixed parameters are drawn.

dampparmfixed(#) works exactly as option dampparmrandom(#) but is applied to drawing fixed param

dampparmrandom(#) is a parameter that controls how aggressively the proposal distributions for r
    of trying to achieve the acceptance rate specified in araterandom(#).  If the parameter is s

from(rowvector) specifies a row vector of starting values for all parameters in order.  If these

fromvariance(matrix) specifies a matrix of starting values for the random parameters.

jumble specifies to randomly mix draws.

noisy specifies that a dot be produced every time a complete pass through the algorithm is finis
    probabilities evaluated at the latest parameters.  While ln_fc(p) is not an objective functi
    other problems.

saving(filename) specifies a location to store the draws from the distribution.  The file will c

replace specifies that an existing file is to be overwritten.

append specifies that an existing file is to be appended, which might be useful if multiple runs

indsave(filename) specifies a file to which individual-level random parameters are to be saved.
    numbers of individuals, specifying this option can cause memory problems. Users should be ca
```

```
indkeep(#) is for use with indsave and specifies that only the last # draws of the individual-le

indwide is for use with indsave and affords the user a degree of control over how individual-lev
    is saved in a row, where draws are marked by the group identifier.  If instead the user woul
    with the first entry of the row being the group identifier.  By analogy with reshape, by def


replaceind functions in the same way as replace, but in reference to the file specified in indsa

appendind functions in the same way as append, but in reference to the file specified in indsave
```

[ ]: