# 1 Files in this folder

This folder contains a program to estimate the Generalized multinomial logit model (G-MNL), its helper functions and a demo with one simulated dataset. The G-MNL model is described in Fiebig, Keane, Louviere, Wasi (Marketing Science, 2010). The program was developed by Michael Keane and Nada Wasi. It is written in Matlab by Nada. Any suggestion/comment can be sent to nada.wasi@gmail.com. It can be distributed freely for non-commercial use only. Current version dated Dec 2009.

To run the demo:

1) drop this folder in c:\

2) in Matlab, addpath c:\gmnl2009b

3) run gmnl_setup.m

Users only need to model gmnl_setup.m to run different models (no need to modify other functions). The outputs of the demo are saved in the subfolder c:\gmnl2009b\output. Note: some subprocedures are modified from Kenneth Train's code in Gauss and other publicly available codes in Matlab. While most variables with similar names have similar meaning to other codes, not all of them are the same.

gmnl_data1.mat is the simulated dataset. Other M-files are related functions.

# 2 GMNL_2009b.m

The main function that the script gmnl_setup.m calls is the function:

**Results = GMNL_2009b (YVEC, XMAT, TIMES, EstimOpt, ModelOpt)**

Users do not need to modify this function but need to give the function the correct inputs. The script gmnl_setup.m shows how to transform Y, X from a stacked format commonly used in other software packages (e.g., STATA, Limdep) into the required YVEC and XMAT of this program. Let NALT denotes the number of alternatives. T denotes the number of scenarios, assumed to be the same across number of people.[1] NP denotes the number of people. K denotes the number of attributes.

- Input

**YVEC:** a 3-D array of dependent variable (0,1) with dimension NALT x T x NP

[1] The code works for the case when T differs, but need some modification in the set-up. This will be explained a later version of the manual.

**XMAT:** a 3-D array attribute variables with dimension K x (NALTxT) x NP

**TIMES:** an NP vector for number of scenarios face by each individual

**EstimOpt:** a structure containing details of estimation procedures and options (see more explanation below)

**ModelOpt:** 3 models {S_MNL, MXL, or G_MNL}

- Output

Results.bhat: parameter estimates

Results.std: standard errors

Results.LL: loglikelihood at convergence

Results.Tol: tolerance at convergence

Results.Iter: number of iterations

Results.b0: initial values

The parameter estimates for these 3 models are different. It is important to know what get estimated, the order of parameters, and suboptions of each model. The example below describe parameters of these 3 models for a 4-attribute example. Denote utility of person $n$ received from choice $j$ in period $t$ by

$$U_{njt} = \beta_n X_{njt} + \varepsilon_{njt}$$

where $X$ is the vector of attributes; $\beta_n$ is the vector of utility weights on product attributes; $\varepsilon_{njt} \sim i.i.d.EV.$

## 2.1 Scale heterogeneity model (S-MNL)

$$\begin{aligned} U_{njt} &= \beta_n X_{njt} + \varepsilon_{njt} \\ &= \sigma_n \beta X_{njt} + \varepsilon_{njt} \end{aligned}$$

To constrain the distribution of $\sigma_n$ to be on the positive support, we specify its distribution to be lognormal: $\ln \sigma_n \sim N(\overline{\sigma}, \tau)$.The mean, $\overline{\sigma}$, is not being estimated but is normalized to one. $\{\varepsilon_{0n}\}$ by default is drawn from $TN(-2, 2)$.

$$\begin{bmatrix} \beta_{1n} \\ \beta_{2n} \\ \beta_{3n} \\ \beta_{4n} \end{bmatrix} = \sigma_n \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix}$$

The parameter estimates are $(\beta_1, \beta_2, \beta_3, \beta_4, \tau)$.

## 2.2 Mixed logit model (MXL)

$$
\begin{aligned}
U_{njt} &= \beta_n X_{njt} + \varepsilon_{njt} \\
&= (\beta + \eta_n) X_{njt} + \varepsilon_{njt} \\
&= (\beta + L\varepsilon_n) X_{njt} + \varepsilon_{njt}
\end{aligned}
$$

where $var(\eta_n) = LL' = \sum$. $\{\varepsilon_n\}$ by default are drawn from standard normal. This program allows 3 options for specifying $\sum$.

### 2.2.1 EstimOpt.COVTYPE = uncorr

$$
\beta_n = \begin{bmatrix} \beta_{1n} \\ \beta_{2n} \\ \beta_{3n} \\ \beta_{4n} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} + \begin{bmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \\ 0 & 0 & l_3 & 0 \\ 0 & 0 & 0 & l_4 \end{bmatrix} \begin{bmatrix} \varepsilon_{1n} \\ \varepsilon_{2n} \\ \varepsilon_{3n} \\ \varepsilon_{4n} \end{bmatrix}
$$

The parameter estimates are $(\beta_1, \beta_2, \beta_3, \beta_4, l_1, l_2, l_3, l_4, )$.

### 2.2.2 EstimOpt.COVTYPE = fullcov

$$
\beta_n = \begin{bmatrix} \beta_{1n} \\ \beta_{2n} \\ \beta_{3n} \\ \beta_{4n} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} + \begin{bmatrix} l_1 & 0 & 0 & 0 \\ l_5 & l_2 & 0 & 0 \\ l_8 & l_6 & l_3 & 0 \\ l_{10} & l_9 & l_7 & l_4 \end{bmatrix} \begin{bmatrix} \varepsilon_{1n} \\ \varepsilon_{2n} \\ \varepsilon_{3n} \\ \varepsilon_{4n} \end{bmatrix}
$$

The estimates are $(\beta_1, \beta_2, \beta_3, \beta_4, l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10})$.

### 2.2.3 EstimOpt.COVTYPE = factor

When the number of attributes is large, estimating full covariance matrix can cause the number of parameters to be proliferated. This option is a compromise between uncorrelated coefficients and full variance-covariance matrix. Coefficients of all attributes are allowed to be correlated, but in a restricted way. In the paper, we use this option when the number of attributes is 15 or more. The current version of the program allows for the following 1-factor analytic structure:

$$
\eta_{n,k} = \alpha_{1k}\phi_{1n} + \xi_{n,k}
$$

where $\phi_{1n} \sim N(0,1)$ and $\xi_{n,k} \sim N(0, \sigma^2_{\xi,k})$ for $n = 1, ..., N$ and $k = 1, ..., K$.

$$var\,(\eta_n) = \alpha'\alpha + \Omega_\xi = \begin{bmatrix} \alpha_{11}^2 + \sigma_{\xi,1}^2 & & & \\ \alpha_{11}\alpha_{12} & \alpha_{12}^2 + \sigma_{\xi,2}^2 & & \\ \vdots & & \ddots & \\ \alpha_{11}\alpha_{1K} & . & . & \alpha_{1K}^2 + \sigma_{\xi,K}^2 \end{bmatrix}$$

The parameter estimates for 4-attribute-case are $(\beta_1, \beta_2, \beta_3, \beta_4, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \sigma_{\xi,1}, \sigma_{\xi,2}, \sigma_{\xi,3}, \sigma_{\xi,4})$.

## 2.3 Generalized multinomial logit model (G-MNL)

$$\begin{aligned} U_{njt} &= \beta_n X_{njt} + \varepsilon_{njt} \\ &= (\sigma_n \beta + \gamma\eta_n + (1-\gamma)\sigma_n\eta_n) X_{njt} + \varepsilon_{njt} \end{aligned}$$

G-MNL nest S-MNL with MXL by adding two additional parameters to MXL. The specification of $\sigma_n$ are similar to S-MNL and the 3 options for specifying $var(\eta_n)$ are similar to MIXL. The additional parameter here is $\gamma$. We do not estimate $\gamma$ directly but estimate $\gamma^*$ where $\gamma = \exp(\gamma^*)/(1 + \exp(\gamma^*))$. Specifically,

### 2.3.1 EstimOpt.COVTYPE = uncorr

$$\begin{bmatrix} \beta_{1n} \\ \beta_{2n} \\ \beta_{3n} \\ \beta_{4n} \end{bmatrix} = \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} + \frac{\exp(\gamma^*)}{1 + \exp(\gamma^*)} \begin{bmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \\ 0 & 0 & l_3 & 0 \\ 0 & 0 & 0 & l_4 \end{bmatrix} \begin{bmatrix} \varepsilon_{1n} \\ \varepsilon_{2n} \\ \varepsilon_{3n} \\ \varepsilon_{4n} \end{bmatrix}$$
$$+ \left(1 - \frac{\exp(\gamma^*)}{1 + \exp(\gamma^*)}\right) \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \begin{bmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \\ 0 & 0 & l_3 & 0 \\ 0 & 0 & 0 & l_4 \end{bmatrix} \begin{bmatrix} \varepsilon_{1n} \\ \varepsilon_{2n} \\ \varepsilon_{3n} \\ \varepsilon_{4n} \end{bmatrix}$$

The parameter estimates are $(\beta_1, \beta_2, \beta_3, \beta_4, l_1, l_2, l_3, l_4, \tau, \gamma^*)$.

### 2.3.2 EstimOpt.COVTYPE = fullcov

$$\begin{bmatrix} \beta_{1n} \\ \beta_{2n} \\ \beta_{3n} \\ \beta_{4n} \end{bmatrix} = \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} + \frac{\exp(\gamma^*)}{1+\exp(\gamma^*)} \begin{bmatrix} l_1 & 0 & 0 & 0 \\ l_5 & l_2 & 0 & 0 \\ l_8 & l_6 & l_3 & 0 \\ l_{10} & l_9 & l_7 & l_4 \end{bmatrix} \begin{bmatrix} \varepsilon_{1n} \\ \varepsilon_{2n} \\ \varepsilon_{3n} \\ \varepsilon_{4n} \end{bmatrix}$$

$$+ \left(1 - \frac{\exp(\gamma^*)}{1+\exp(\gamma^*)}\right) \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \begin{bmatrix} l_1 & 0 & 0 & 0 \\ l_5 & l_2 & 0 & 0 \\ l_8 & l_6 & l_3 & 0 \\ l_{10} & l_9 & l_7 & l_4 \end{bmatrix} \begin{bmatrix} \varepsilon_{1n} \\ \varepsilon_{2n} \\ \varepsilon_{3n} \\ \varepsilon_{4n} \end{bmatrix}$$

The parameter estimates are $(\beta_1, \beta_2, \beta_3, \beta_4, l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}, \tau, \gamma^*)$.

### 2.3.3 EstimOpt.COVTYPE = factor

See explanation of the restriction on $var\,(\eta_n)$ in MIXL section. The parameter estimates for 4-attribute-case are $(\beta_1, \beta_2, \beta_3, \beta_4, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \sigma_{\xi,1}, \sigma_{\xi,2}, \sigma_{\xi,3}, \sigma_{\xi,4}, \tau, \gamma^*)$.

# 3 Other options

## 3.1 EstimOpt.NOTSCALE option

**EstimOpt.NOTSCALE:** a vector 1 x K dimension taking values 1 or 0. Attributes with value 1 will not be scaled. For example, in the paper, we consider one modified S-MNL where we scale all attributes except the intercept. Let's assume the first attribute is an intercept and we want to assume homogeneous intercept across the sample. We will specify ModelOpt = S_MNL and EstimOpt.NOTSCALE = [1 0 0 0], which implies:

$$\begin{bmatrix} \beta_{1n} \\ \beta_{2n} \\ \beta_{3n} \\ \beta_{4n} \end{bmatrix} = \begin{bmatrix} 1 \\ \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \\ \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \\ \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \end{bmatrix} .* \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix}$$

## 3.2 EstimOpt.CZVAR and EstimOpt.BACTIVE option

**EstimOpt.CZVAR:** {0 or 1} indicates if there is any parameter to be constrained to its initial value.

**EstimOpt.BACTIVE:** a 1 x K vector. Each element is {0 or 1} 0 if the parameters are not being estimated (not active); 1 otherwise.

This option is relevant when one wants to allow some random & some fixed coefficients, or wants to constraint $\gamma$ to a certain value. For example, in the paper, we have a model where we allows for scale heterogeneity except the intercept but allows for random intercept. Consider the same 4-attribtue example again with one random intercept. Here we will specify ModelOpt = G_MNL; EstimOpt.COV = uncorr; EstimOpt.NOTSCALE = [1 0 0 0].The parameter estimates without EstimOpt.CZVAR = 1 are $(\beta_1, \beta_2, \beta_3, \beta_4, l_1, l_2, l_3, l_4, \tau, \gamma^*)$. In our specification, we want $l_2 = l_3 = l_4 = 0$ and $\gamma^*$ = large negative number so that $\gamma = 0$. To do so, we set EstimOpt.CZVAR = 1; EstimOpt.BACTIVE = [1 1 1 1 1 0 0 0 1 0]. The initial values of $l_2, l_3, l_4$ are set to 0, and the initial value of $\gamma^*$ is set to -5. This implies:

$$\begin{bmatrix} \beta_{1n} \\ \beta_{2n} \\ \beta_{3n} \\ \beta_{4n} \end{bmatrix} = \begin{bmatrix} 1 \\ \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \\ \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \\ \exp(\overline{\sigma} + \tau\varepsilon_{0n}) \end{bmatrix} .* \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} + \begin{bmatrix} l_1\varepsilon_{1n} \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

### 3.3 Optimization options

**EstimOpt.OPTIM** = {1,2} 1 for BHHH algorithm; 2 for Matlab algorithm (to be added)

**EstimOpt.NREP** = positive integer for number of draws. default is 500.

Options specifically for the written BHHH algortihm

**EstimOpt.DISPLAY** ={0,1} 1 for display all steps and iteration; default = 0.

**EstimOpt.NITER** = positive interge for maximum number of iteration. default = 500.

**EstimOpt.NUMGRAD** = {0,1} 1 to use numerical gradients; 0 to use analytical gradients. Using analytical gradients can help speed up the program but can be less accurate if users do not scale X matrix properly. Analytical gradients for factor model is not available. Before using analytical gradients, it is safer to check if analytical gradients are close to numerical gradients by using options below.

**EstimOpt.GRADCHECK** = {0,1} by setting EstimOpt.GRADCHECK = 1. The program will only show values of analytical gradients and numerical gradients, and terminate.

**EstimOpt.LBVAR** and **EstimOpt.UBVAR** = a K x 1 vector to set the lower bound and upper bound of parameters. By default, if ModelOpt is G_MNL, the program will set the lower and upper bounds of $\gamma^*$ to -5 and 5, respectively.

**EstimOpt.DRAWS** = {1,2,3} 1 draw 1+K x NP from independent standard normal; 2 draw 1+K x NP using halton draws; 3 draw 1 x NP vector from a truncated normal and K x NP from standard normal. EstimOpt.LB and EstimOpt.UB are bounded for this truncated normal. Default is EstimOpt.DRAWS = 3 with EstimOpt.LB = -2 and EstimOpt.UB = 2.

**EstimOpt.SETSEED** = {0,1} to set a fixed seed for random draws. Default is 1.