Flexible Multinomial Logit Models with Preference Space and Willingness-to-Pay Space Utility Specifications in R: The logitr Package

John Paul Helveston
George Washington University

Abstract

In many applications of discrete choice models, modelers are interested in estimating consumer's marginal "willingness-to-pay" (WTP) for different attributes. WTP can computed by dividing the estimated parameters of a utility model in the preference space by the price parameter or by estimating a utility model in the WTP space. For homogeneous models, these two procedures generally produce the same estimates of WTP, but the same is not true for heterogeneous models where model parameters are assumed to follow a specific distribution. The **logitr** package was written to allow for flexible estimation of multinomial logit models with preference space and WTP space utility specifications. The package supports homogeneous multinomial logit (MNL) and heterogeneous mixed logit (MXL) models, including support for normal and log-normal parameter distributions. Since MXL models and models with WTP space utility specifications are non-convex, an option is included to run a multi-start optimization loop with random starting points in each iteration. The package also includes a market simulation function to estimate the expected market shares of a set of alternatives using an estimated model.

Keywords: multinomial logit, preference space, willingness-to-pay space, discrete choice, R.

```
Warning: package 'kableExtra' was built under R version 3.4.4
```

1. Introduction

In many applications of discrete choice models, modelers are interested in estimating consumer's marginal "willingness-to-pay" (WTP) for different attributes. WTP can be estimated in two ways:

1. Estimate a discrete choice model in the "preference space" where parameters have units of utility and then compute the WTP by dividing the parameters by the price parameter.
2. Estimate a discrete choice model in the "WTP space" where parameters have units of WTP.

While the two procedures generally produce the same estimates of WTP for homogenous models, the same is not true for heterogeneous models where model parameters are assumed to follow a specific distribution, such as normal or log-normal (Train and Weeks 2005). For example, in a preference space specification, a normally distributed attribute parameter divided by a log-normally distributed price parameter produces a strange WTP distribution with large tails. In contrast, a WTP space specification allows the modeler to directly assume WTP is normally distributed. The **logitr** package was developed to enable modelers to choose between these two utility spaces when estimating multinomial logit models.

2. The random utility model

The random utility model is a well-established framework in many fields for estimating consumer preferences from observed consumer choices (Louviere, Hensher, and Swait 2000, Train (2009)). Random utility models assume that consumers choose the alternative j a set of alternatives that has the greatest utility u_j. Utility is a random variable that is modeled as $u_j = v_j + \varepsilon_j$, where v_j is the "observed utility" (a function of the observed attributes such that $v_j = f(\mathbf{x}_j)$) and ε_j is a random variable representing the portion of utility unobservable to the modeler.

Adopting the same notation as in (Helveston, Feit, and Michalek 2018), consider the following utility model:

$$u_j^* = \boldsymbol{\beta}^{*\prime}\mathbf{x}_j - \alpha^* p_j + \varepsilon_j^*, \qquad \varepsilon_j^* \sim \text{Gumbel}\left(0, \sigma^2\frac{\pi^2}{6}\right) \tag{1}$$

where $\boldsymbol{\beta}^*$ is the vector of coefficients for non-price attributes $\mathbf{x}_j$, α^* is the coefficient for price p_j, and the error term, ε_j^*, is an IID random variable with a Gumbel extreme value distribution of mean zero and variance $\sigma^2(\pi^2/6)$. This model is not identified since there exists an infinite set of combinations of values for $\boldsymbol{\beta}^*$, α^*, and σ that produce the same choice probabilities. In order to specify an identifiable model, the modeler must normalize equation (1). One approach is to normalize the scale of the error term by dividing equation (1) by σ, producing the "preference space" utility specification:

$$\left(\frac{u_j^*}{\sigma}\right) = \left(\frac{\boldsymbol{\beta}^*}{\sigma}\right)'\mathbf{x}_j - \left(\frac{\alpha^*}{\sigma}\right)p_j + \left(\frac{\varepsilon_j^*}{\sigma}\right), \qquad \left(\frac{\varepsilon_j^*}{\sigma}\right) \sim \text{Gumbel}\left(0, \frac{\pi^2}{6}\right) \tag{2}$$

The typical preference space parameterization of the multinomial logit (MNL) model can then be written by rewriting equation (2) with $u_j = (u_j^*/\sigma)$, $\boldsymbol{\beta} = (\boldsymbol{\beta}^*/\sigma)$, $\alpha = (\alpha^*/\sigma)$, and $\varepsilon_j = (\varepsilon_j^*/\sigma)$:

$$u_j = \boldsymbol{\beta}'\mathbf{x}_j - \alpha p_j + \varepsilon_j \qquad \varepsilon_j \sim \text{Gumbel}\left(0, \frac{\pi^2}{6}\right) \tag{3}$$

The vector $\boldsymbol{\beta}$ represents the marginal utility for changes in each non-price attribute, and α represents the marginal utility obtained from price reductions. In addition, the coefficients $\boldsymbol{\beta}$ and α are measured in units of *utility*, which only has relative rather than absolute meaning.

The alternative normalization approach is to normalize equation (1) by α^* instead of σ, producing the "willingness-to-pay (WTP) space" utility specification:

$$\left(\frac{u_j^*}{\alpha^*}\right) = \left(\frac{\boldsymbol{\beta}^*}{\alpha^*}\right)'\mathbf{x}_j - p_j + \left(\frac{\varepsilon_j^*}{\alpha^*}\right), \qquad \left(\frac{\varepsilon_j^*}{\alpha^*}\right) \sim \text{Gumbel}\left(0, \frac{\sigma^2}{(\alpha^*)^2}\frac{\pi^2}{6}\right) \tag{4}$$

Since the error term in equation is scaled by $\lambda^2 = \sigma^2/(\alpha^*)^2$, we can rewrite equation (4) by multiplying both sides by $\lambda = (\alpha^*/\sigma)$ and renaming $u_j = (\lambda u_j^*/\alpha^*)$, $\boldsymbol{\omega} = (\boldsymbol{\beta}^*/\alpha^*)$, and $\varepsilon_j = (\lambda \varepsilon_j^*/\alpha^*)$:

$$u_j = \lambda \left(\boldsymbol{\omega}'\mathbf{x}_j - p_j\right) + \varepsilon_j \qquad \varepsilon_j \sim \text{Gumbel}\left(0, \frac{\pi^2}{6}\right) \tag{5}$$

Here $\boldsymbol{\omega}$ represents the marginal WTP for changes in each non-price attribute, and λ represents the scale of the deterministic portion of utility relative to the standardized scale of the random error term.

The utility models in equations 3 and 5 represent the preference space and WTP space utility specifications, respectively. In equation 3, WTP is estimated as $\hat{\boldsymbol{\beta}}/\hat{\alpha}$; in equation 5, WTP is simply $\hat{\boldsymbol{\omega}}$.

3. The logitr package

3.1. Installation

This package has not been uploaded to CRAN, but it can be directly installed from Github using the **devtools** library. The package also depends on the **nloptr** library.

First, make sure you have the **devtools** and **nloptr** libraries installed:

```
R> install.packages("devtools")
R> install.packages("nloptr")
```

Then load the **devtools** library and install the **logitr** package:

```
R> library("devtools")
R> install_github("jhelvy/logitr")
```

3.2. Data format

The data must be arranged the following way:

1. The data must be a `data.frame` object.
2. Each row is an alternative from a choice observation. Each choice observation does not have to have the same number of alternatives.
3. Each column is a variable.
4. One column must identify `obsID` (the "observation ID"): a sequence of numbers that identifies each unique choice occasion. For example, if the first three choice occasions had 2 alternatives each, then the first 9 rows of the `obsID` variable would be `1,1,2,2,3,3`.
5. One column must identify `choice`: a dummy variable that identifies which alternative was chosen (`1`=chosen, `0`=not chosen).
6. For WTP space models, once column must identify `price`: a continous variable of the price values.

An example of of the 'Yogurt' data set from the **mlogit** package illustrates this format:

```
R> library("logitr")

Loading required package: nloptr

R> data(yogurt)
R> head(yogurt, 12)

   id obsID choice price feat   brand dannon hiland weight yoplait
1   1     1      0   8.1    0  dannon      1      0      0       0
2   1     1      0   6.1    0  hiland      0      1      0       0
3   1     1      1   7.9    0  weight      0      0      1       0
4   1     1      0  10.8    0 yoplait      0      0      0       1
5   1     2      1   9.8    0  dannon      1      0      0       0
6   1     2      0   6.4    0  hiland      0      1      0       0
7   1     2      0   7.5    0  weight      0      0      1       0
8   1     2      0  10.8    0 yoplait      0      0      0       1
9   1     3      1   9.8    0  dannon      1      0      0       0
10  1     3      0   6.1    0  hiland      0      1      0       0
11  1     3      0   8.6    0  weight      0      0      1       0
12  1     3      0  10.8    0 yoplait      0      0      0       1
```

3.3. The logitr() function

The main model estimation function is the `logitr()` function:

```
R> model = logitr(data, choiceName, obsIDName, parNames, priceName=NULL,
R>              randPars=NULL, randPrice=NULL, modelSpace="pref",
R>              options=list(...))
```

The function returns a list of values, so assign the model output to a variable (e.g. `model`) to store the output values.

Arguments

Argument	Description	Default
`data`	The choice data, formatted as a data.frame object.	–
`choiceName`	The name of the column that identifies the `choice` variable.	–
`obsIDName`	The name of the column that identifies the `obsID` variable.	–
`parNames`	The names of the parameters to be estimated in the model. Must be the same as the column names in the `data` argument. For WTP space models, do not include `price` in `parNames`.	–
`priceName`	The name of the column that identifies the price variable. Only required for WTP space models.	`NULL`

Argument	Description	Default
`randPars`	A named vector whose names are the random parameters and values the destribution: `'n'` for normal or `'ln'` for log-normal.	`NULL`
`randPrice`	The random distribution for the price parameter: `'n'` for normal or `'ln'` for log-normal. Only used for WTP space MXL models.	`NULL`
`modelSpace`	Set to `'wtp'` for WTP space models.	`'pref'`
`options`	A list of options.	–

Options

Argument	Description	Default
`numMultiStarts`	Number of times to run the optimization loop, each time starting from a different random starting point for each parameter between `startParBounds`. Recommended for non-convex models, such as WTP space models and MXL models.	`1`
`keepAllRuns`	Set to `TRUE` to keep all the model information for each multistart run. If `TRUE`, the `logitr()` function will return a list with two values: `models` (a list of each model), and `bestModel` (the model with the largest log-likelihood value).	`FALSE`
`startParBounds`	Set the `lower` and `upper` bounds for the starting parameters for each optimization run, which are generated by `runif(n, lower, upper)`.	`c(-1,1)`
`startVals`	A vector of values to be used as starting values for the optimization. Only used for the first run if `numMultiStarts > 1`.	`NULL`
`useAnalyticGrad`	Set to `FALSE` to use numerically approximated gradients instead of analytic gradients during estimation (which is slower).	`TRUE`
`scaleInputs`	By default each variable in `data` is scaled to be between 0 and 1 before running the optimization routine because it usually helps with stability, especially if some of the variables have very large or very small values (e.g. `> 10^3` or `< 10^-3`). Set to `FALSE` to turn this feature off.	`TRUE`
`standardDraws`	By default, a new set of standard normal draws are generated during each call to `logitr` (the same draws are used during each multistart too). The user can override those draws by providing a matrix of standard normal draws if desired.	`NULL`
`numDraws`	The number of draws to use for MXL models for the maximum simulated likelihood.	`200`

Argument	Description	Default
`drawType`	The type of draw to use for MXL models for the maximum simulated likelihood. Set to `'normal'` to use random normal draws or `'halton'` for Halton draws.	`'halton'`
`printLevel`	The print level of the `nloptr` optimization loop. Type `nloptr.print.options()` for more details.	`0`
`xtol_rel`	The relative `x` tolerance for the `nloptr` optimization loop. Type `nloptr.print.options()` for more details.	`1.0e-8`
`xtol_abs`	The absolute `x` tolerance for the `nloptr` optimization loop. Type `nloptr.print.options()` for more details.	`1.0e-8`
`ftol_rel`	The relative `f` tolerance for the `nloptr` optimization loop. Type `nloptr.print.options()` for more details.	`1.0e-8`
`ftol_abs`	The absolute `f` tolerance for the `nloptr` optimization loop. Type `nloptr.print.options()` for more details.	`1.0e-8`
`maxeval`	The maximum number of function evaluations for the `nloptr` optimization loop. Type `nloptr.print.options()` for more details.	`1000`

Values

Value	Description
`coef`	The model coefficients at convergence.
`standErrs`	The standard errors of the model coefficients at convergence.
`logLik`	The log-likelihood value at convergence.
`nullLogLik`	The null log-likelihood value (if all coefficients are 0).
`gradient`	The gradient of the log-likelihood at convergence.
`hessian`	The hessian of the log-likelihood at convergence.
`numObs`	The number of observations.
`numParams`	The number of model parameters.
`startPars`	The starting values used.
`multistartNumber`	The multistart run number for this model.
`time`	The user, system, and elapsed time to run the optimization.
`iterations`	The number of iterations until convergence.
`message`	A more informative message with the status of the optimization result.
`status`	An integer value with the status of the optimization (positive values are successes). Type `logitr.statusCodes()` for a detailed description.
`modelSpace`	The model space (`'pref'` or `'wtp'`).
`standardDraws`	The draws used during maximum simulated likelihood (for MXL models).
`randParSummary`	A summary of any random parameters (for MXL models).
`parSetup`	A summary of the distributional assumptions on each model parameter (`"f"`="fixed", `"n"`="normal distribution", `"ln"`="log-normal distribution").

Value	Description
`options`	A list of all the model options.

References

Helveston JP, Feit EM, Michalek JJ (2018). "Pooling stated and revealed preference data in the presence of RP endogeneity." *Transportation Research Part B: Methodological*, **109**, 70–89.

Louviere JJ, Hensher DA, Swait J (2000). *Stated Choice Methods: Analysis and Applications*. Cambridge University Press, Cambridge, Massachusetts.

Train KE (2009). *Discrete Choice Methods with Simulation*. 2nd edition. Cambridge University Press.

Train KE, Weeks M (2005). "Discrete Choice Models in Preference and Willingness-to-Pay Space." In *Appl. Simul. Methods Environ. Resour. Econ.*, chapter 1, pp. 1–16.

Affiliation:

John Paul Helveston
George Washington University
Science & Engineering Hall
800 22nd St NW
Washington, DC 20052
E-mail: `jph@gwu.edu`
URL: `http://jhelvy.com`