Package: RSGHB (via r-universe)

November 13, 2026

Version 1.2.2.9001

Title Functions for Hierarchical Bayesian Estimation: A Flexible Approach

Description Functions for estimating models using a Hierarchical Bayesian (HB) framework. The flexibility comes in allowing the user to specify the likelihood function directly instead of assuming predetermined model structures. Types of models that can be estimated with this code include the family of discrete choice models (Multinomial Logit, Mixed Logit, Nested Logit, Error Components Logit and Latent Class) as well ordered response models like ordered probit and ordered logit. In addition, the package allows for flexibility in specifying parameters as either fixed (non-varying across individuals) or random with continuous distributions. Parameter distributions supported include normal, positive/negative log-normal, positive/negative censored normal, and the Johnson SB distribution. Kenneth Train's Matlab and Gauss code for doing Hierarchical Bayesian estimation has served as the basis for a few of the functions included in this package. These Matlab/Gauss functions have been rewritten to be optimized within R. Considerable code has been added to increase the flexibility and usability of the code base. Train's original Gauss and Matlab code can be found here: <http://elsa.berkeley.edu/Software/abstracts/train1006mxlhb.html> See Train's chapter on HB in Discrete Choice with Simulation here: <http://elsa.berkeley.edu/books/choice2.html>; and his paper on using HB with non-normal distributions here: <http://eml.berkeley.edu//~train/trainsonnier.pdf>. The authors would also like to thank the invaluable contributions of Stephane Hess and the Choice Modelling Centre: <https://cmc.leeds.ac.uk/>. Compatibility updates by Renova.

Depends methods, MCMCpack

Imports graphics, grDevices, stats, utils

URL https://github.com/RSGInc/RSGHB

License GPL-3

Copyright Resource Systems Group, Inc. 2019

NeedsCompilation yes

Author Jeff Dumont [aut, cre], Jeff Keller [aut], Chase Carpenter [ctb]

Maintainer Jeff Dumont <Jeff.Dumont@rsginc.com>

Repository https://renova-r.r-universe.dev

Date/Publication 2026-11-13 04:57:15 UTC

RemoteUrl https://github.com/renova-r/RSGHB

RemoteRef renova-1.2.2.9001

RemoteSha 742256a548336944ebda7589652a3b5742d0b418

Contents

choicedata 2
doHB 3
plot.RSGHB 7
writeModel 8

Index **11**

`choicedata` *A synthetic discrete choice dataset*

Description

A synthetic discrete choice dataset with each observation having two generic alternatives composed of two attributes (travel time and toll). The data are panel in nature (as opposed to cross-sectional) with 8 choice tasks per decision maker.

Usage

```
data(choicedata)
```

Format

A data frame with one row per choice task.

`ID` Decision maker identifier

`thecount` Experiment number

`tt1` Travel time for the first alternative

`tt2` Travel time for the second alternative

`toll2` Toll for the second alternative

`asc1` Dummy for alternative specific constant

`Choice` Choice made

Details

These choice data were synthetically derived to be used in generating examples of how RSGHB is used to estimate discrete choice models.

See Also

doHB

Examples

```
data(choicedata)

# Frequency of alternatives chosen
table(choicedata$Choice)
```

---

doHB *Estimate a Hierarchical Bayesian Model*

Description

The user can initiate the model estimation by calling the doHB function. The function will optionally perform initial diagnostic tests to look for common errors in specifying the model. Upon completion, the function will optionally write a number of output files with the model parameters and convergence statistics to the user's working directory.

The flexibility comes in allowing the user to specify the likelihood function directly instead of assuming predetermined model structures. Types of models that can be estimated with this code include the family of discrete choice models (Multinomial Logit, Mixed Logit, Nested Logit, Error Components Logit and Latent Class) as well as ordered response models like ordered probit and ordered logit. In addition, the package allows for flexibility in specifying parameters as either fixed (non-varying across individuals) or random with continuous distributions. Parameter distributions supported include normal, positive/negative log-normal, positive/negative censored normal and the Johnson SB distribution.

Kenneth Train's Matlab and Gauss code for doing Hierarchical Bayesian estimation has served as the basis for a few of the functions included in this package. (See references below).

Usage

```
doHB(likelihood_user, choicedata, control = list())
```

Arguments

likelihood_user

A function that returns likelihood values for each observation in your data set. This function must accept arguments fc and b, representing the fixed parameters and individual parameters, respectively, and computes the likelihood of observing the data given those values.

choicedata	A data.frame of choice data to be used in estimation. Minimally requires an 'ID' column associated with the vector of likelihoods returned by likelihood_user.
control	A list of estimation controls. See below for more details.

Details

The fc argument to the likelihood_user function is a numeric vector of length length(gVarNamesFixed). It is NULL if gVarNamesFixed is NULL.

The b argument to the likelihood_user function is a numeric matrix with length(gVarNamesNormal) columns and length(likelihood_user(...)) rows. In other words, one column per random parameter and one row per choice task. It is NULL if gVarNamesNormal is NULL.

There are a number of global variables that can be set to control the model estimation. Some need to be specified directly in the model control list while others have default values that can be adjusted by the analyst if something other than the default is desired.

User-specified controls

constraintsNorm - A list of monotonic constraints to be applied during estimation. The structure of the constraints is c(param1number, inequality, param2number). For constraints relative to 0, use 0 instead of param2number. For inequality, use 1 for < and 2 for >.

For example constraintsNorm = list(c(5,1,0), c(6,1,5), c(7,1,6)) would constrain the 5th parameter < 0, the 6th parameter < 5th parameter, and the 7th parameter < the 6th parameter. If NULL, no constraints are used. (Defaults to NULL)

degreesOfFreedom - Additional degrees of freedom for the prior variance-covariance matrix, not including the number of parameters. (Defaults to 5)

FC - A vector of starting values for the fixed parameters. There should be an element for each name in *gVarNamesFixed*. (Defaults to rep(0, length(gVarNamesFixed)))

fixedA - Fixes the means of the underlying normal distribution of random variables to certain values as opposed to estimating them. This would be important for example in an error components logit model or an integrated choice and latent variable model. The format for this input is a vector of length equal to the number of random parameters. Use NA for variables that should be estimated, e.g., fixedA = c(NA, NA, NA, NA, NA, NA, NA, 0). In this case, the mean of the underlying normal for the 8th random variable would be fixed to 0. If NULL, all means are estimated. (Defaults to NULL)

fixedD - Fixes the variance of the underlying normal distribution of the random variables to certain values as opposed to estimating them. This would be important for example in an integrated choice and latent variable model. The format for this input is a vector of length equal to the number of random parameters. Use NA for variables that should be estimated, e.g., fixedD = c(NA, NA, NA, NA, NA, NA, NA, 1). In this case, the variance of the underlying normal for the 8th random variable would be fixed to 1. If NULL, all variances are estimated. (Defaults to NULL)

gDIST - A vector of integers (1-6) that indicate which type of distribution should be applied to the random parameters. 1 = Normal, 2 = Postive Log-Normal, 3 = Negative Log-Normal, 4 = Positive Censored Normal, 5 = Negative Censored Normal, 6 = Johnson SB. There should be an element for each name in gVarNamesNormal. (Defaults to rep(1, length(gVarNamesNormal)))

gFULLCV - Indicates whether a full variance-covariance structure should be used for the random parameters. (Defaults to TRUE)

gINFOSKIP - Number of iterations between printing/plotting information about the iteration process. (Defaults to 250)

gMAXCOEF - A vector of maximums for the Johnson SB distributions. If Johnson SB is used, each random coefficent needs an element but only the elements that correspond to a JSB in `gDIST` are used. (Defaults to `0`)

gMINCOEF - A vector of minimums for the Johnson SB distributions. If Johnson SB is used, each random coefficent needs an element but only the elements that correspond to a JSB in `gDIST` are used. (Defaults to `0`)

gNCREP - Number of burn-in iterations to use prior to convergence. (Defaults to `100000`)

gNEREP - Number of iterations to keep for averaging after convergence has been reached. (Defaults to `100000`)

gNSKIP - Number of iterations in between retaining draws for averaging. (Defaults to `1`)

gVarNamesFixed - A character vector of names for the fixed parameters. (REQUIRED)

gVarNamesNormal - A character vector of names for the random parameters. (REQUIRED)

gStoreDraws - Whether to store the draws for the individual level parameters. Doing so can dramatically increase the memory usage of the model if there are a large number of individuals or draws. (Defaults to `FALSE`)

hIW - A boolean indicating if a hierarchical Inverted Wishart should be used when sampling in posterior distribution for the covariance matrix. New in version 1.2.0. (Defaults to `FALSE`)

modelname - The model name which is used for creating output files. (Defaults to `"HBModel"`)

nodiagnostics - If `TRUE`, an initial diagnostic report and prompt is not reported to the screen. This makes batch processing more seamless. (Defaults to `FALSE`)

priorVariance - The prior variance assumed. Ignored if `pvMatrix` is not `NULL`. (Defaults to `2`)

pvMatrix - A custom prior variance-covariance matrix to be used in estimation. The prior variance-covariance matrix needs to be a matrix object and of the correct size: `length(gVarNamesNormal)` by `length(gVarNamesNormal)`. If provided, overrides `priorVariance`. (Defaults to `NULL`)

rho - The initial proportionality fraction for the jumping distribution of the Metropolis-Hastings algorithm for the random parameters. This fraction is adjusted after each iteration to target an acceptance rate of `targetAcceptanceNormal`. (Defaults to `0.1`)

rhoF - The proportionality fraction for the jumping distribution for the Metropolis-Hastings algorithm for the fixed parameters. This fraction is adjusted after each iteration to target an acceptance rate of `targetAcceptanceFixed`. (Defaults to `0.0001`)

svN - A vector of starting values for the means of the underlying normals for the random parameters. There should be an element for each name in `gVarNamesNormal`. (Defaults to `rep(0, length(gVarNamesNormal))`)

targetAcceptanceFixed - The target acceptance rate in the Metropolis-Hastings algorithm for the fixed parameters. (Defaults to `0.3`)

targetAcceptanceNormal - The target acceptance rate in the Metropolis-Hastings algorithm for the random parameters. (Defaults to `0.3`)

verbose - Whether estimation information should be printed/plotted during the iteration process. (Defaults to `TRUE`)

writeModel - Indicates whether the model results should be written as a series of CSV files to the working directory upon estimation completion. (Defaults to `FALSE`, see `writeModel`)

Value

a model object of class RSGHB. Contains:

modelname	A character string identifying the model.
params.fixed	A character vector naming the estimated fixed parameters.
params.vary	A character vector naming the estimated random parameters.
distributions	A character vector of assumed distributions for each random parameter.
pv	The prior variance-covariance matrix assumed for estimation.
df	Additional degrees of freedom in the model.
gNP	The number of individuals in the model.
gOBS	The number of observations in the model.
gNCREP	The number of burn-in iterations used prior to convergence.
gNEREP	The number of iterations used for averaging.
constraints	A list of constraints. (see 'Details' above)
iter.detail	A data.frame of model statistics at every gINFOSKIP'th iteration.
A	A matrix containing the sample-level means of the underlying normals at each iteration. Is NULL if no random parameters were estimated.
B, Bsd	A matrix containing the mean individual-level draws across iterations for the underlying normals. The Bsd object provides the standard deviations of these individual draws. Is NULL if no random parameters were estimated.
C, Csd	A matrix containing the mean individual-level draws across iterations for the underlying normals while including the appropriate distribution transformations. The Csd object provides the standard deviations of these individual draws. Is NULL if no random parameters were estimated.
D	An array of the sample variance-covariance matrix for each iteration. Is NULL if no random parameters were estimated.
F	A matrix containing the set of fixed (non-random) parameters at each iteration. Is NULL if no fixed parameters were estimated.
choices	A vector of choices if provided.
p	A vector of probabilities at the mean values of C and F.
ll0	The initial log-likelihood given the starting values of sVN and FC.
llf	The final log-likelihood at the mean values of C and F.

Author(s)

Jeffrey Dumont <jeff.dumont@rsginc.com>, Jeff Keller <jeff.keller@rsginc.com>

References

Train, K. (2009) Discrete Choice Methods with Simulation. Cambridge University Press.

Train, K. and Sonnier G. (2005) *Mixed Logit with Bounded Distributions of Correlated Partworths*, Applications of Simulation Methods in Environmental and Resource Economics. Edited by Anna Alberini and Riccardo Scarpa. http://elsa.berkeley.edu/~train/trainsonnier.pdf

Train, K. Original Gauss and Matlab code: http://elsa.berkeley.edu/Software/abstracts/train1006mxlhb.html

See Also

plot.RSGHB, writeModel

Examples

```
# Organize choicedata for modeling
data(choicedata)
tt1 <- choicedata$tt1
tt2 <- choicedata$tt2
toll2 <- choicedata$toll2
choice1 <- (choicedata$Choice==1)
choice2 <- (choicedata$Choice==2)

# The model likelihood function
likelihood <- function(fc, b) {

     # Assign Beta vectors to named parameters for convenience
     cc    <- 1
     wtp1  <- b[, cc]; cc <- cc + 1
     price <- b[, cc]; cc <- cc + 1

     # Discrete choice utility in WTP-space
     v1 <-                 price * wtp1 * tt1
     v2 <- price * toll2 + price * wtp1 * tt2

     # Return the probability of choice
     p  <- (exp(v1)*choice1 + exp(v2)*choice2) / (exp(v1) + exp(v2))
     return(p)
}

# Estimation controls/settings
control <- list(
     modelname = "MNL_WTPSpace",
     gVarNamesNormal = c("WTP", "Price"),
     gNCREP = 300,
     gNEREP = 100,
     gINFOSKIP = 10,
     gNSKIP = 2,
     nodiagnostics = TRUE
)

# Estimate the model
set.seed(1987)

          model <- doHB(likelihood, choicedata, control)
```

---

plot.RSGHB *Plot RSGHB Model Object Components*

Description

This function plots various components of an RSGHB model object.

Usage

```
## S3 method for class 'RSGHB'
plot(x, ...)
```

Arguments

x	An RSGHB model object.
...	Additional arguments passed to methods. See Details.

Details

A type argument can be passed to control which component is plotted. If type is "A", then the population means of the underlying normal distributions are plotted. If type is "F", then the fixed parameters are plotted. If type is "Log" (the default), then the model statistics at every gINFOSKIPth iteration are plotted.

An error is returned if the type component is not found in x.

Author(s)

Jeff Keller <jeff.keller@rsginc.com>, Jeffrey Dumont <jeff.dumont@rsginc.com>

See Also

doHB

Examples

```
# plot(model, type = "A")
# plot(model, type = "F")
# plot(model, type = "Log")
```

---

writeModel *Write an RSGHB Model Object as Series of CSVs*

Description

Function for writing an RSGHB model object as a series of CSV files to the disk. Files are written in Sawtooth's CBC HB format to allow for easier integration with analysis processes built around that software.

Usage

```
writeModel(object, writeDraws = FALSE, path = getwd())
```

Arguments

object	an RSGHB model object.
writeDraws	whether to write out the individual draw files. (Defaults to FALSE)
path	the folder location to write the files. (Defaults to getwd())

Details

All output files are named with a object[["modelname"]] prefix. (E.g. modelname.log, modelname_C.csv)

If writeDraws is TRUE, a CSV file for each of object[["gNP"]] individuals is written to the disk. This may take a long time if object[["gNEREP"]] or object[["gNP"]] is large.

Value

None, though a series of files is written to path:

modelname.log
: A log file containing summary model information and the estimation iteration detail.

modelname_logPlot.png
: A faceted plot of all appropriate model statistics stored in the iteration detail of the model (object[["iter.detail"]]). Depending on the type of parameters estimated, some statistics may be NULL or NA and are omitted.

modelname_A.csv
: If object was estimated with random parameters, contains the sample-level means of the underlying normals at each iteration.

modelname_B.csv, modelname_Bsd.csv
: If object was estimated with random parameters, contains the mean individual-level draws across iterations for the underlying normals. The Bsd file provides the standard deviations of these individual draws.

modelname_C.csv, modelname_Csd.csv
: If object was estimated with random parameters, contains the mean individual-level draws across iterations for the underlying normals while including the appropriate distribution transformations. The Csd file provides the standard deviations of these individual draws. These two files are equivalent to the conditional distributions from models estimated using Maximum Simulated Likelihood methods.

modelname_D.csv
: If object was estimated with random parameters, contains a row-based representation of the sample variance-covariance matrix for each iteration.

modelname_F.csv
: If object was estimated with fixed parameters, contains the set of fixed (non-random) parameters at each iteration in object[["gNEREP"]].

modelname_pvMatrix.csv
: If object was estimated with random parameters, contains the prior variance-covariance matrix that was assumed to estimate the model.

Author(s)

Jeff Keller <jeff.keller@rsginc.com>

See Also

doHB

Examples

```
# writeModel(model)
```

Index

∗ **datasets**
choicedata, 2

choicedata, 2

doHB, 3, *8*, *10*

plot *(*plot.RSGHB*)*, 7
plot.RSGHB, 7, *7*

writeModel, *5*, *7*, 8