Package: hbbr (via r-universe)

October 14, 2026

Type Package

Title Hierarchical Bayesian Benefit-Risk Assessment Using Discrete Choice Experiment

Version 1.1.2

Date 2019-10-07

Maintainer Saurabh Mukhopadhyay <stat.mukherjee@gmail.com>

Description Implements assessment of benefit-risk balance using Bayesian Discrete Choice Experiment. For more details see the article by Mukhopadhyay et al. (2019) <DOI:10.1080/19466315.2018.1527248>.

Depends R (>= 3.2), R2jags

Suggests ggplot2, gridExtra, testthat

Note Package 'R2jags' requires JAGS executable file installed

License GPL-2

Encoding UTF-8

LazyData TRUE

RoxygenNote 6.1.1

NeedsCompilation no

Author Saurabh Mukhopadhyay [aut] (AbbVie Inc.), Saurabh Mukhopadhyay [cre]

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

Date/Publication 2019-10-25 07:20:02 UTC

RemoteUrl https://github.com/cran/hbbr

RemoteRef HEAD

RemoteSha 39f252c403b158e0fe01fe29be92960d1beeb768

Contents

hbbr.Fit 2
hbbrAug.Fit 5
hbbrPilotResp 8
simAugData 10

Index **12**

`hbbr.Fit` *hbbr.Fit (Fits processed response data to hbbr model)*

Description

Fits processed benefit-risk survey data from an appropriately designed discrete choice experiment to the hbbr (Hierarchical Bayesian Benefit-Risk) model. For details see article by Mukhopadhyay, S., Dilley, K., Oladipo, A., & Jokinen, J. (2019). Hierarchical Bayesian Benefit–Risk Modeling and Assessment Using Choice Based Conjoint. Statistics in Biopharmaceutical Research, 11(1), 52-60.

Usage

```
hbbr.Fit(brdta, design, tune.param = list(tau = 0.01, eta = NULL, df.add
  = 2), mcmc = list(burnin = 5000, iter = 1e+05, nc = 2, thin = 20),
  verbose = TRUE)
```

Arguments

`brdta`	processed and coded survey response data to be fitted to the hbbr model. It is a data frame in which 1st two columns indicate subject id and subject response (y = 0 or 1), and remaining columns contain information on design matrix (X). See Details below for more information.
`design`	design information of the experiment: design = list(b, r, bl, rl, blbls, rlbls) where, b is number of benefit attributes, r is number of risk attributes, bl and rl are vectors of integers of length b and r indicating number of levels in j-th benefit attribute and k-th risk attribute, respectively. blbls, rlbls consists of labels for benefit and risk attributes. When blbls is NULL, it uses "B1", "B2", ... and similarly for rlbls.
`tune.param`	a list of tuning hyper-parameters to be used; default tune.param=list(tau=0.01, eta=NULL). See Details below for more information.
`mcmc`	a list of mcmc parameters to be used in the Gibbs sampler to obtain posterior samples of the paramaters of interests; default: mcmc=list(burnin=5000, iter=100000, nc=2, thin=20). See Details below for more information.
`verbose`	TRUE or FALSE: flag indicating whether to print intermediate output summary which might be helpful to see convergence results.

Details

brdta is a processed and coded survey response data to be fitted to the hbbr model. It is a data frame in which 1st column contains ID of respondent, 2nd column contains response (y = 0 or 1) - each value corresponds to each choice-pair card evaluated by the respondent: y =1 if the 1st choice of the pair was preferred; 0 otherwise, 3rd column onwards contain information on design matrix (X). Each row of X is a vector of indicator variables taking values 0, 1, or -1; a value of 0 is used to denote absence of an attribute level; a value of 1 or -1 is used to indicate presence of an attribute level in the 1st choice, or in the 2nd choice, respectively in the choice-pair presented to the respondent. Note that column corresponding to the 1st level for each attribute would not be included in X as the part-worth parameter (beta) for the 1st level of each attribute is assumed to be 0 without loss of generality. So, if there are b benefit attributes and r risk attributes, and then have bl_j and rl_k levels (j=1,...,b; k=1,...,r) then total number of columns brdta is Sum_over_j(bl_j-1) + Sum_over_k(rl_k-1). If there are B respondents, each responding to k choice-pairs, then brdta will have B*k rows.

tune.param is a list of tuning hyper-parameters (tau, eta) for the hbbr model. Specifically, in the hbbr model beta.h ~ MVN(beta.bar, V.beta) where the hyper-prior of beta.bar is assumed to be MVN (beta0, B) with B = 1/tau*I; and hyper-prior of V.beta is assumed to follow inverse Wishart IW(nue, V) with V = 1/eta*I. When eta is NULL then eta will take the default value of m+3 which is the DF for the Wishart distribution. If we think the respondents have very similar part-worth vectors, then use eta=1.

mcmc is a list of MCMC specification parameters: (a) burnin - contains the number of burn-in values to be generated, (b) iter - is the total number of iterations of each chain beyond burn-in, (c) nc - is the number of independent chains, and (d) thin = posterior samples to be saved for every 'thin' values of the MCMC samples in each of the 'nc' chains. For more details see R2jags package help files.

Value

returns a list of useful output of interest and input specifications: (bbar.mcmc, bbar.means, bbar.sds, summary, logL, design, model, brdata, other.inputs).

Author(s)

Saurabh Mukhopadhyay

Examples

```
## Sample calls: fits pilot response data included with the package


  data(hbbrPilotResp)
  hbfit = hbbr.Fit(brdta=hbbrPilotResp$brdta, design=hbbrPilotResp$design,
                   mcmc=list(burnin=500, iter=10000, nc=2, thin=10))
  hb = hbfit$bbar.mcmc
  dgn = hbfit$design
  mns = hbfit$bbar.means
  sds = hbfit$bbar.sd # same as apply(hbfit$bbar.mcmc, 2, sd)

  ## Plots of MCMC draws ------------------------------------
```

```
op=par(mfrow=c(1,2), mar = c(4,2,3,1),oma=c(.1,.1,2,.1))
matplot(hb,type="l",xlab="Iterations",ylab="",
        main=paste("Average Part-Worths (beta-bars)"),
        cex.main=.8, cex.lab=0.8, axes=FALSE)
axis(1, at=seq(0,dim(hb)[1],length.out = 6),
        labels= paste(seq(0,5,1)*dim(hb)[1]/5 *hbfit$other.inputs$thin, sep=""),
        cex.axis=0.8)
axis(2, cex.axis=0.8,las=1)

plot(hbfit$logL, type="l",main="Log Likelihood", axes=FALSE,xlab="Iterations",ylab="",
    cex.main=.8,cex.lab=0.8)
axis(1, at=seq(0,dim(hb)[1],length.out = 6),
    labels= paste(seq(0,5,1)*dim(hb)[1]/5 *hbfit$other.inputs$thin, sep=""),
     cex.axis=0.8)
axis(2, cex.axis=0.8,las=1)
title(outer=TRUE, main = paste("MCMC draws plotted at every ",
     hbfit$other.inputs$thin,"-th Iteration",sep=""),cex.main=.9)
par(op)

## Plots for mean estimated part-worth utilities ------------------
require(ggplot2)
require(gridExtra)

b.mns = c()
b.sds = c()
b.atr = c()
b.lvl = c()
j.now=1
for (j in 1:dgn$b) {
  b.mns = c(b.mns,0, mns[j.now:(j.now-1+dgn$bl[j]-1)])
  b.sds = c(b.sds,0, sds[j.now:(j.now-1+dgn$bl[j]-1)])
  b.atr = c(b.atr, rep(dgn$blbls[j], dgn$bl[j]))
  b.lvl = c(b.lvl, paste("E", 1:dgn$bl[j],sep=""))
  j.now = j.now-1+dgn$bl[j]
}

r.mns = c()
r.sds = c()
r.atr = c()
r.lvl = c()
k.now=j.now
for (k in 1:dgn$r) {
  r.mns = c(r.mns,0,mns[k.now:(k.now-1+dgn$rl[k]-1)])
  r.sds = c(r.sds,0, sds[k.now:(k.now-1+dgn$rl[k]-1)])
  r.atr = c(r.atr, rep(dgn$rlbls[k], dgn$rl[k]))
  r.lvl = c(r.lvl, paste("H", 1:dgn$rl[k],sep=""))
  k.now = k.now-1+dgn$rl[k]
}

d0.b = data.frame(Attributes =b.atr, lvl=b.lvl, util = b.mns, se = b.sds)
d0.r = data.frame(Attributes =r.atr, lvl=r.lvl, util = r.mns, se = r.sds)
y.max = max(abs(mns) + max(sds))
pd <- position_dodge(0.2) # move them .2 to the left and right
```

```
  pb = ggplot(data = d0.b, aes(x=lvl, y=util, group=Attributes,color=Attributes)) +
    ylim(0, y.max) +
    geom_hline(yintercept = 0) +
    geom_line(size=1.5, position=pd) +
    geom_point(size=4, shape=22, fill="green",color="darkgreen", position=pd) +
    geom_errorbar(aes(ymin=util-se, ymax=util+se), width=0.2, position=pd) +
    xlab("Benefit-Attribute Levels") + ylab("Estimated Utility") +
    ggtitle("Estimated Partworth Utilities of Benefits") +
    scale_color_manual(values=c("deepskyblue3" , "#9999CC", "cyan3" )) +
    theme(legend.position="bottom",plot.title = element_text(size = 10))

  pr = ggplot( data = d0.r, aes(x=lvl, y=util, group=Attributes,color=Attributes)) +
    ylim(-y.max,0)+
    geom_hline(yintercept = 0) +
    geom_line(size=1.5, position=pd) +
    geom_point(size=4, shape=22, fill="pink",color="darkred", position=pd) +
    geom_errorbar(aes(ymin=util-se, ymax=util+se), width=0.2, position=pd) +
    xlab("Risk-Attribute Levels") + ylab("Estimated Utility") +
    ggtitle("Estimated Partworth Utilities of Risks") +
    scale_color_manual(values=c("orange" , "maroon" )) +
    theme(legend.position="bottom",plot.title = element_text(size = 10))

  grid.arrange(pb, pr, nrow = 1)

##-------------------------------------------------------------------
```

`hbbrAug.Fit` *hbbrAug.Fit (Fits processed response data to the augmented hbbr model)*

Description

Fits processed benefit-risk survey data from an appropriately designed discrete choice experiment to the augmented hbbr model that includes patients' baseline characteristics. For details see article by Mukhopadhyay, S., Dilley, K., Oladipo, A., & Jokinen, J. (2019). Hierarchical Bayesian Benefit–Risk Modeling and Assessment Using Choice Based Conjoint. Statistics in Biopharmaceutical Research, 11(1), 52-60.

Usage

```
hbbrAug.Fit(brdta, Z, design, tune.param = list(tau = 0.01, eta = NULL,
  df.add = 2), mcmc = list(burnin = 5000, iter = 1e+05, nc = 2, thin =
  20), verbose = TRUE)
```

Arguments

brdta	processed and coded survey response data to be fitted to the hbbr model. It is a data frame in which 1st two columns indicate subject id and subject response (y = 0 or 1), and remaining columns contain information on design matrix (X). See Details below for more information.
Z	matrix of observed baseline characteristics of the patients. If there are N patients responded to the survey and we have included g characteristics of each patient then Z is a matrix of (g+1) x N, with all elements of the first column equal to 1. Note that when g=0, the model reduces to regular hbbr model.
design	design information of the experiment: design = list(b, r, bl, rl, blbls, rlbls) where, b is number of benefit attributes, r is number of risk attributes, bl and rl are vectors of integers of length b, and r indicating number of levels in j-th benefit attribute and k-th risk attribute, respectively. blbls, rlbls consists of labels of benefit and risk attributes. When blbls is NULL, it uses "B1", "B2", ... and similarly for rlbls.
tune.param	a list of tuning hyper-parameters to be used; default tune.param=list(tau=0.01, eta=NULL). See Details below for more information.
mcmc	a list of mcmc parameters to be used in the Gibbs sampler to obtain posterior samples of the parameters of interests; default: mcmc=list(burnin=1000, iter=10000, nc=4, thin=10). See Details below for more information.
verbose	TRUE or FALSE: flag indicating whether to print intermediate output summary which might be helpful to see convergence results.

Details

brdta is a processed and coded survey response data to be fitted to the hbbr model. It is a data frame in which 1st column contains ID of respondent, 2nd column contains response (y = 0 or 1) - each value corresponds to each choice-pair card evaluated by the respondent: y =1 if the 1st choice of the pair was preferred; 0 otherwise, 3rd column onwards contain information on design matrix (X). Each row of X is a vector of indicator variables taking values 0, 1, or -1; a value of 0 is used to denote absence of an attribute level; a value of 1 or -1 is used to indicate presence of an attribute level in the 1st choice, or in the 2nd choice, respectively in the choice-pair presented to the respondent. Note that column corresponding to the 1st level for each attribute would not be included as the part-worth parameter (beta) for the 1st level of each attribute is assumed to be 0 without loss of generality. So, if there are b benefit attributes and r risk attributes, and then have bl_j and rl_k levels (j=1,...,b; k=1,...,r) then total number of columns brdta is Sum_over_j(bl_j-1) + Sum_over_k(rl_k-1). If there are B respondents each responding to k choice-pairs then brdta will have B*k rows.

tune.param is a list of tuning hyper-parameters (tau, eta) for the hbbr model. Specifically, in the hbbr model beta.h ~ MVN(beta.bar, V.beta) where the hyper-prior of beta.bar is assumed to be MVN (beta0, B) with B = 1/tau*I; and hyper-prior of V.beta is assumed to follow inverse Wishart IW(nue, V) with V = 1/eta*I. When eta is NULL then eta will take the default value of m+3 which is the DF for the Wishart distribution. If we think the respondents have very similar part-worth vectors, then use eta=1.

mcmc is a list of MCMC specification parameters to be used for rjags package: (a) burnin - contains the number of burn-in values to be generated, (b) iter - is the total number of iterations of each chain

beyond burn-in, (c) nc - is the number of independent chains, and (d) thin = posterior samples to be saved for every 'thin' values of the MCMC samples in each of the 'nc' chains. For more details see rjags package help files.

Value

returns a list of useful output of interest and input specifications: (del.mcmc, del.means, del.sds, summary, logL, design, model, brdata, other.inputs).

Author(s)

Saurabh Mukhopadhyay

Examples

```
## Sample calls:
# fits simulated response data included with this package to augmented hbbr model
# and then plots the estimated part-worth utilities.


data("simAugData")
hbA = hbbrAug.Fit(brdta= simAugData$brdtaAug, Z=simAugData$Z,
              design=simAugData$design,
              tune.param=list(tau=0.01, eta=NULL, df.add=2),
              mcmc=list(burnin=500, iter=10000, nc=2, thin=10))

# define an appropriate function to plot the part-worth values...
partworth.plot = function(attr.lvl, beta.mns, nb=3, new=TRUE, pnt =15, cl=clrs)
{
#check dimension
  k = length(attr.lvl) # no of attributes
  bk = length(unlist(attr.lvl)) # no of levels acrosss attributes
  if (bk - k != length(beta.mns)) stop("error 1")
  mns = rep(0, length(unlist(attr.lvl)))
  cntr = 0
  for (j in 1:k)
  {
    for (i in 1:length(attr.lvl[[j]])){
      cntr = cntr +1
      if (i > 1) mns[cntr]= beta.mns[cntr-1-(j-1)]
    }
  }
  indx = list()
  j0=1
  for (j in 1:k) {
    j1 = (j0+length(attr.lvl[[j]])-1)
    indx[[j]]= j0:j1
    j0=j1+1
 }
  if (new) {
    plot(c(1,bk), c(floor(min(beta.mns)*1.2),ceiling(max(beta.mns)*1.2)),
                                        type="n", axes=FALSE, xlab="",ylab="")
```

```
    axis(2, at=0:ceiling(max(beta.mns)*1.2), las=1, cex.axis=.7)
    axis(4, at=floor(min(beta.mns)*1.2):0, las=1, cex.axis=.7)
  }
  vl=c()
  for (j in 1:k)
  {
    points(indx[[j]], mns[indx[[j]]], type="b", pch =pnt, col=cl[j])
    vl=c(vl, max(indx[[j]])+.5)
  }
  abline(v=vl,col="gray", h=0)
  box()
}

# Plotting estimated betas (part-worth) for some selected baseline characteristics:

augattr.lvl = list(b1=paste("B1",1:3,sep=""),b2=paste("B2",1:3,sep=""),
         r1=paste("R1",1:3,sep=""),r2=paste("R2",1:3,sep=""))
clrs = c("blue", "green4","orange4", "red3")

mns = hbA$del.means
# est. part-worth values
betmn1 = mns %*% matrix(c(1, 0, 1), ncol=1)  # at mean age with disease staus=1
betmn2 = mns %*% matrix(c(1, 0, -1), ncol=1) # at mean age with disease staus=-1
betmn3 = mns %*% matrix(c(1, 1, -1), ncol=1) # at age = mean+1*SD, disease staus=-1

partworth.plot(attr.lvl = augattr.lvl, beta.mns = betmn1)
partworth.plot(attr.lvl = augattr.lvl, beta.mns = betmn2, new=FALSE, pnt=17)
partworth.plot(attr.lvl = augattr.lvl, beta.mns = betmn3, new=FALSE, pnt=16)

# Plotting true betas at those baseline characteristics
Del = simAugData$Del
clrs = rep("darkgrey", 4)
# true part-worth values
bmn1 = Del %*% matrix(c(1, 0, 1), ncol=1)  # at mean age with disease staus=1
bmn2 = Del %*% matrix(c(1, 0, -1), ncol=1) # at mean age with disease staus=-1
bmn3 = Del %*% matrix(c(1, 1, -1), ncol=1) # at age = mean+1*SD, disease staus=-1
partworth.plot(attr.lvl = augattr.lvl, beta.mns = bmn1)
partworth.plot(attr.lvl = augattr.lvl, beta.mns = bmn2, new=FALSE, pnt=17)
partworth.plot(attr.lvl = augattr.lvl, beta.mns = bmn3, new=FALSE, pnt=16)
```

`hbbrPilotResp` *A list consisting of pilot data and associated discrete choice design information for the HBBR model framework.*

Description

Data from 23 respondents each choosing preference from 18 choice cards. Choice cards were randomly generated from 108 total choices. For details see article by Mukhopadhyay, S., et al.

"Hierarchical Bayesian Benefit-Risk Modeling and Assessment Using Choice Based Conjoint." Statistics in Biopharmaceutical Research 11.1 (2019): 52-60.

Usage

```
data(hbbrPilotResp)
```

Format

A list consisting of pilot response data from 23 experts and design information to be used to fit the HBBR model

brdta A data frame with 23x18 = 414 rows and 15 columns consists of responses from 23 experts each providing tradeoff responses to 18 choice pairs. The 1st column consists of responders' id. The 2nd column contains binary responses (1 indicating 1st of the choice pair was selected, 0 indicating 2nd was selected). Remaining 13 columns contain the design matrix X taking values 0, 1, or -1; a value of 1 or -1 is used to indicate presence of an attribute level in the 1st choice or in the 2nd choice of the choice pair, respectively; a value of 0 is used to indicate absence of an attribute in the choice pair. See Details below for more about the discrete choice experiment that is coded as design matrix X.

design A list of structure (b, r, bl, rl), where b and r indicate number of benefit and risk attributes, bl is a vector of integers of size b consisting number of levels within each benefit attribute; similarly rl is a vector of integers of size r consisting number of levels within each risk attribute.

Details

The discrete choice experiment (DCE) included 3 benefit attributes (b=3): overall survival (OS), objective response rate (ORR), fatigue reduction (FTG); and 2 risk attributes (r=2): febrile neutropenia (FebNEU) and severe pneumonia (SevPNA). There were 4 levels for each of the benefit attributes (ORR, OS, and FTG) (i.e. bl= rep(4,3)) and 3 levels for each of the 2 risk attributes (FebNEU and SevPNA) (i.e. rl = rep(3,2)). The DCE produced b*r*(4 choose 2)*(3 choose 2) = 108 distinct non-dominant choice pairs each with one benefit and one risk attribute. Panels (questionnaires) were generated with 18 randomly selected choice pairs per panel from the set of 108 choice pairs. Since the part-worth of various levels within each attribute are to be measured relatively to the part-worth of the 1st level of the attribute, columns for the 1st level of the attributes are not required. Thus, we have sum(bl)-b + sum(br)-r = 13 columns are needed to obtain information on the X matrix which are stored as the last 13 columns of brdta.

References

Mukhopadhyay, S. et al. "Hierarchical Bayesian Benefit–Risk Modeling and Assessment Using Choice Based Conjoint." Statistics in Biopharmaceutical Research 11.1 (2019): 52-60.

Examples

```
data(hbbrPilotResp)
```

simAugData *A list consisting of simulated data, design, baseline profiles, and true part-worth matrix for the Augmented HBBR model framework.*

Description

Simulated response data and associated information from 100 respondents each choosing preference from 12 choice cards. Choice cards were randomly generated from 36 total choices.

Usage

```
data(simAugData)
```

Format

A list consisting of simulated response data from 100 subjects, design information, baseline profiles, and true part-worth matrix of the Augmented HBBR model framework.

brdtaAug A data frame of 100x12 rows and 10 columns consists of simulated responses from 100 subjects each providing tradeoff responses to 12 choice pairs. The 1st column consists of subject id. The 2nd column contains binary responses (1 indicating 1st of the choice pair was selected, 0 indicating 2nd was selected). Remaining 8 columns contain the design matrix X taking values 0, 1, or -1; a value of 1 or -1 is used to indicate presence of an attribute level in the 1st choice or in the 2nd choice of the choice pair, respectively; a value of 0 is used to indicate absence of an attribute in the choice pair. See Details below for more about the discrete choice experiment that is coded as design matrix X.

design A list of structure (b, r, bl, rl), where b and r indicate number of benefit and risk attributes, bl is a vector of integers of size b consisting number of levels within each benefit attribute; similarly rl is a vector of integers of size r consisting number of levels within each risk attribute.

Z A data frame of size 100x3 consists of baseline characteristics from 100 subjects. The 1st column of Z is vector of 1, the 2nd column consists of standardized age, and the 3rd column indicates disease status at baseline (1= present, -1 = not present). The 1st column being a constant vector of 1 indicates that the part-worth matrix to be estimated, say Del-hat, would have the overall mean part-worth in the 1st column, while 2nd and 3rd columns would consists of estimates of additive components of part-worth due to age and disease status.

Del A matrix of true part-worth values of dimension 8x3. This was used along with X and Z to generate the responses from 100 subjects captured in the 2nd column of brdtaSim.

Details

The simulated discrete choice experiment (DCE) included 3 benefit attributes (b=2): B1, B2 (say) and 2 risk attributes (r=2): R1, R2 (say). There were 3 levels for each of the benefit attributes ("Low", "Moderate", "High") (i.e. bl= rep(3,2)) and 3 levels for each of the risk attributes ("None", "Mild", "Severe") (i.e. rl = rep(3,2)). The DCE produced 36 distinct non-dominant choice pairs each with one benefit and one risk attribute. Panels (questionnaires) were generated with 12 (randomly)

selected choice pairs per panel from the set of 36 choice pairs. Since the part-worth of various levels within each attribute are to be measured relatively to the part-worth of the 1st level of the attribute, columns for the 1st level of the attributes are not required. Thus, we have sum(bl)-b + sum(br)-r = 8 columns are needed to obtain information on the X matrix which are stored as the last 8 columns of brdtaAugSim.

Examples

```
data(simAugData)
```

Index

∗ **datasets**
hbbrPilotResp, 8
simAugData, 10

hbbr.Fit, 2
hbbrAug.Fit, 5
hbbrPilotResp, 8

simAugData, 10