Package ‘beanz’

August 9, 2023

Title Bayesian Analysis of Heterogeneous Treatment Effect

Version 3.1

Author Chenguang Wang [aut, cre],
Ravi Varadhan [aut],

Maintainer Chenguang Wang <cwang68@jhmi.edu>

License GPL (>= 3)

Description It is vital to assess the heterogeneity of treatment effects (HTE) when making health care decisions for an individual patient or a group of patients. Nevertheless, it remains challenging to evaluate HTE based on information collected from clinical studies that are often designed and conducted to evaluate the efficacy of a treatment for the overall population. The Bayesian framework offers a principled and flexible approach to estimate and compare treatment effects across subgroups of patients defined by their characteristics. This package allows users to explore a wide range of Bayesian HTE analysis models, and produce posterior inferences about HTE. See Wang et al. (2018) <DOI:10.18637/jss.v085.i07> for further details.

Depends R (>= 3.4.0), Rcpp (>= 0.12.0), methods

Imports rstan (>= 2.18.1), rstantools (>= 1.5.0), survival, loo,
RcppParallel (>= 5.0.1)

LinkingTo StanHeaders (>= 2.18.0), rstan (>= 2.18.1), BH (>= 1.66.0),
Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1)

LazyData true

ByteCompile true

SystemRequirements GNU make

NeedsCompilation yes

Suggests knitr, shiny, rmarkdown, pander, shinythemes, DT, testthat

RoxygenNote 7.2.3

VignetteBuilder knitr

Repository CRAN

Date/Publication 2023-08-09 10:30:13 UTC
Description

This package contains the functions for running Bayesian models implemented in STAN for HTE analysis.

Notation

Consider a randomized two-arm clinical trial. Let $Y$ denote the response and $Z$ denote treatment arm assignment. For subgroup analysis, assume there are $P$ baseline covariates, $X_1, \ldots, X_P$, of interest. The covariates can be binary, ordinal with numerical values, or nominal variables. Let $\Omega = \{(X_1, \ldots, X_P)\}$ denote the collection of subgroups defined by the covariates. Let $\theta_g$ denote the treatment effect in subgroup $G = g$, and let $\hat{\theta}_g$ be the estimated $\theta$ in subgroup $G = g$ with $\hat{\sigma}_g^2$ the estimated variance associated with $\hat{\theta}_g$.

Models

We approximate the distribution of $\hat{\theta}_g$ by

$$\hat{\theta}_g | \theta_g, \sigma_g^2 \sim N(\theta_g, \sigma_g^2)$$

and assign an informative prior to $\sigma_g$.

We consider two options in the software: log-normal or uniform prior. The uniform prior is specified as:

$$\log \sigma_g | \hat{\sigma}_g, \Delta \sim Unif(\log \hat{\sigma}_g - \Delta, \log \hat{\sigma}_g + \Delta)$$

and the log-normal prior is specified as:

$$\log \sigma_g | \hat{\sigma}_g, \Delta \sim N(\log \hat{\sigma}_g, \Delta)$$

where $\Delta$ is a parameter specified by the users.

We consider a set of models together with the priors for $\theta_g$: 

Bayesian Approaches for HTE Analysis
**No subgroup effect model** This model assumes that patients in all the subgroups are exchangeable. That is, all the subgroups are statistically identical with regard to the treatment effect and there is no subgroup effect. Information about treatment effects can be directly combined from all subgroups for inference. The model is specified as follows:

$$
\theta_g = \mu \\
\mu \sim N(MU, B),
$$

where $MU$ should be set to 0 in most cases, and $B$ is large in relation to the magnitude of the treatment effect size so that the prior for $\mu$ is essentially non-informative.

**Full stratification model** The subgroups are fully distinguished from each other with regard to the treatment effect. There is no information about treatment effects shared between any subgroups. The model is specified as follows:

$$
\theta_g = \mu_g \\
\mu_g \sim N(MU, B).
$$

**Simple regression model** The model introduces a first-order, linear regression structure. This model takes into account the information that the subgroups are formulated based on the set of baseline covariates. The coefficients are assumed to be exchangeable among subgroups. Information about treatment effects are shared between subgroups with similar baseline covariates through these coefficients. The model is specified as follows:

$$
\theta_g|X_{g,i} = \mu + \sum_{j=1}^{P} X_{g,i}^{j} \gamma_j \\
\mu \sim N(MU, B) \\
\gamma_j \sim N(0, C) \quad j = 1, \ldots, P.
$$

**Basic shrinkage model** This approach assumes all subgroups are exchangeable with regards to the treatment effect. The model is specified as follows:

$$
\theta_g = \mu + \phi_g \\
\mu \sim N(MU, B) \\
\phi_g \sim N(0, \omega^2) \\
\omega \sim Half - N(D).
$$

**Simple regression and shrinkage model** This model combines basic regression with shrinkage, with a linear regression structure and a random effect term. Direct estimates are shrunken towards the regression surface. The model is specified as follows:

$$
\theta_g = \mu + \sum_{j=1}^{P} X_{g,i}^{j} \gamma_j + \phi_g \\
\mu \sim N(MU, B) \\
\gamma_j \sim N(0, 1C) \quad j = 1, \ldots, P \\
\phi_g \sim N(0, \omega^2) \\
\omega \sim Half - N(D).
$$

**Dixon and Simon model** This model assumes that the elements in coefficient are exchangeable with each other, which allows information sharing among covariate effects. Similar to the simple regression model, only the first-order interactions are considered. The model is specified as follows:
\[
\theta_g = \mu + \sum_{j=1}^{P} X_{g,j} \gamma_j \\
\mu \sim N(MU, B) \\
\gamma_j \sim N(0, \omega^2) \\
\omega \sim \text{Half-N}(D).
\]

**Extended Dixon and Simon model** This approach extends the Dixon and Simon model by introducing the higher-order interactions, with the interaction effects exchangeable. The model is specified as follows:

\[
\theta_g = \mu + \sum_{k=1}^{P} \sum_{j \in \xi(k)} X_{\xi(k),j} \gamma_j^{(k)} \\
\mu \sim N(MU, B) \\
\gamma_j^{(k)} \sim N(0, \omega_k^2) \quad k = 1, \ldots, P, \quad j \in \xi(k) \\
\omega_k \sim \text{Half-N}(D),
\]

where \(\xi(k)\) denotes the set of \(k\)th order interaction terms.

**Graphical user interface (GUI)**

This package provides a web-based Shiny GUI. See **bzShiny** for details.

**References**


---

**bzCallStan**

**Call STAN models**

**Description**

Call STAN to draw posterior samples for Bayesian HTE models.

**Usage**

```r
bzCallStan(
  mdls = c("nse", "fs", "sr", "bs", "srs", "ds", "eds"),
  dat.sub,
  var.estvar,
  var.cov,
  par.pri = c(B = 1000, C = 1000, D = 1, MU = 0),
  var.nom = NULL,
  delta = 0,
)```
prior.sig = 1,
chains = 4,
...
)

Arguments

mdls
name of the Bayesian HTE model. The options are:

nse  No subgroup effect model
fs   Full stratification model
sr   Simple regression model
bs   Basic shrinkage model
srs  Simple regression with shrinkage model
ds  Dixon-Simon model
eds  Extended Dixon-Simon model
dat.sub
dataset with subgroup treatment effect summary data
var.estvar
column names in dat.sub that corresponds to treatment effect estimation and the
estimated variance
var.cov
array of column names in dat.sub that corresponds to binary or ordinal baseline
covariates
par.pri
vector of prior parameters for each model. See beanz-package for the details
of model specification.

nse, fs  B
sr  B, C
bs, ds, eds  B, D
srs  B, C, D
nse, fs, sr, bs, srs, ds, eds  MU
var.nom
array of column names in dat.sub that corresponds to nominal baseline covari-
ates
delta
parameter for specifying the informative priors of $\sigma_g$
prior.sig
option for the informative prior on $\sigma_g$. 0: uniform prior and 1: log-normal prior
chains
STAN options. Number of chains.
...
options to call STAN sampling. These options include iter, warmup, thin, algorithm. See rstan::sampling for details.

Value

A class beanz.stan list containing

mdl  name of the Bayesian HTE model
stan.rst  raw rstan sampling results
smps  matrix of the posterior samples
get.mus  method to return the posterior sample of the subgroup treatment effects
DIC  DIC value
looic leave-one-out cross-validation information criterion
rhat  Gelman and Rubin potential scale reduction statistic
prior.sig option for the informative prior on $\sigma_g$
delta parameter for specifying the informative priors of $\sigma_g$

Examples

```r
## Not run:
var.cov <- c("sodium", "lvef", "any.vasodilator.use");
var.resp <- "y";
var.trt  <- "trt";
var.censor <- "censor";
resptype <- "survival";
var.estvar <- c("Estimate", "Variance");

subgrp.effect <- bzGetSubgrpRaw(solvd.sub,
                               var.resp   = var.resp,
                               var.trt    = var.trt,
                               var.cov    = var.cov,
                               var.censor = var.censor,
                               resptype   = resptype);

rst.nse <- bzCallStan("nse", dat.sub=subgrp.effect,
                       var.estvar = var.estvar, var.cov = var.cov,
                       par.pri = c(B=1000, MU = 0),
                       chains=4, iter=600,
                       warmup=200, thin=2, seed=1000);

rst.sr  <- bzCallStan("sr", dat.sub=subgrp.effect,
                       var.estvar=var.estvar, var.cov = var.cov,
                       par.pri=c(B=1000, C=1000),
                       chains=4, iter=600,
                       warmup=200, thin=2, seed=1000);
## End(Not run)
```

bzComp  

Comparison of posterior treatment effects

Description

Present the difference in the posterior treatment effects between subgroups

Usage

bzSummaryComp(stan.rst, sel.grps = NULL, cut = 0, digits = 3, seed = NULL)

bzPlotComp(stan.rst, sel.grps = NULL, ..., seed = NULL)
bzForestComp(  
    stan.rst,  
    sel.grps = NULL,  
    ...,  
    quants = c(0.025, 0.975),  
    seed = NULL  
)

Arguments

stan.rst  a class beanz.stan object generated by `bzCallStan`

sel.grps  an array of subgroup numbers to be included in the summary results

cut  cut point to compute the probability that the posterior subgroup treatment effects is below

digits  number of digits in the summary result table

seed  random seed

...  options for `plot` function

quants  lower and upper quantiles of the credible intervals in the forest plot

Value

`bzSummaryComp` generates a data frame with summary statistics of the difference of treatment effects between the selected subgroups. `bzPlotComp` generates the density plot of the difference in the posterior treatment effects between subgroups. `bzForestComp` generates the forest plot of the difference in the posterior treatment effects between subgroups.

See Also

`bzCallStan`

Examples

```r
# Not run:
var.cov <- c("sodium", "lvef", "any.vasodilator.use");
var.resp <- "y";
var.trt  <- "trt";
var.censor <- "censor";
resptype <- "survival";
var.estvar <- c("Estimate", "Variance");

subgrp.effect <- bzGetSubgrpRaw(solvd.sub,
    var.resp  = var.resp,
    var.trt   = var.trt,
    var.cov   = var.cov,
    var.censor = var.censor,
    resptype  = resptype);

rst.sr   <- bzCallStan("sr", dat.sub=subgrp.effect,
```
bzGailSimon

Gail-Simon Test

Description

Gail-Simon qualitative interaction test.

Usage

bzGailSimon(effects, sderr, d = 0)

Arguments

effects                subgroup treatment effects
sderr                   standard deviation of the estimated treatment effects
d                   clinically meaningful difference

Examples

## Not run:
var.cov <- c("sodium", "lvef", "any.vasodilator.use");
var.resp <- "y";
var.trt <- "trt";
var.censor <- "censor";
resptype <- "survival";
subgrp.effect <- bzGetSubgrp(solvd.sub,
                              var.resp = var.resp,
                              var.trt = var.trt,
                              var.cov = var.cov,
                              var.censor = var.censor,
                              resptype = resptype);

gs.pval <- bzGailSimon(subgrp.effect$Estimate,
                        subgrp.effect$Variance);
## End(Not run)
bzGetSubgrp

Get subgroup treatment effect estimation and variance

Description

Compute subgroup treatment effect estimation and variance for subgroup effect summary data. The estimation and variance are combined if there are multiple record of the same subgroup, defined by the covariates, in the data.

Usage

bzGetSubgrp(data.all, var.ey, var.variance, var.cov)

Arguments

data.all subject level dataset
var.ey column name in data.all for estimated treatment effect
var.variance column name in data.all for variance of subgroup treatment assignment
var.cov array of column names in dat.all that corresponds to binary or ordinal baseline covaraites

Value

A dataframe with treatment effect estimation and variance for each subgroup

bzGetSubgrpRaw

Get subgroup treatment effect estimation and variance

Description

Compute subgroup treatment effect estimation and variance from subject level data.

Usage

bzGetSubgrpRaw(
    data.all,
    var.resp,
    var.trt,
    var.cov,
    var.censor,
    resptype = c("continuous", "binary", "survival")
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data.all</td>
<td>subject level dataset</td>
</tr>
<tr>
<td>var.resp</td>
<td>column name in data.all for response</td>
</tr>
<tr>
<td>var.trt</td>
<td>column name in data.all for treatment assignment</td>
</tr>
<tr>
<td>var.cov</td>
<td>array of column names in dat.all that corresponds to binary or ordinal baseline covaraites</td>
</tr>
<tr>
<td>var.censor</td>
<td>column name in data.all for censoring if the response is time to event data</td>
</tr>
<tr>
<td>resptype</td>
<td>type of response. The options are binary, continuous or survival</td>
</tr>
</tbody>
</table>

Value

A dataframe with treatment effect estimation and variance for each subgroup

Examples

```r
## Not run:
var.cov <- c("sodium", "lvef", "any.vasodilator.use");
var.resp <- "y";
var.trt  <- "trt";
var.censor <- "censor";
resptype <- "survival";
subgrp.effect <- bzGetSubgrpRaw(solvd.sub,
                                  var.resp = var.resp,
                                  var.trt = var.trt,
                                  var.cov = var.cov,
                                  var.censor = var.censor,
                                  resptype = resptype);
## End(Not run)
```

bzPredSubgrp  Predictive Distribution

Description

Get the predictive distribution of the subgroup treatment effects

Usage

bzPredSubgrp(stan.rst, dat.sub, var.estvar)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stan.rst</td>
<td>a class beanz.stan object generated by bzCallStan</td>
</tr>
<tr>
<td>dat.sub</td>
<td>dataset with subgroup treatment effect summary data</td>
</tr>
<tr>
<td>var.estvar</td>
<td>column names in dat.sub that corresponds to treatment effect estimation and the estimated variance</td>
</tr>
</tbody>
</table>
Value

A dataframe of predicted subgroup treatment effects. That is, the distribution of

$$\theta_g | \hat{\theta}_1, \hat{\sigma}_1^2, \ldots, \hat{\theta}_G, \hat{\sigma}_G^2.$$ 

Examples

```r
## Not run:
var.cov <- c("sodium", "lvef", "any.vasodilator.use");
var.resp <- "y";
var.trt <- "trt";
var.censor <- "censor";
resptype <- "survival";
var.estvar <- c("Estimate", "Variance");
subgrp.effect <- bzGetSubgrp(solvd.sub,
    var.resp = var.resp,
    var.trt = var.trt,
    var.cov = var.cov,
    var.censor = var.censor,
    resptype = resptype);

rst.nse <- bzCallStan("nse", dat.sub=subgrp.effect,
    var.estvar = var.estvar, var.cov = var.cov,
    par.pri = c(B=1000),
    chains=4, iter=4000,
    warmup=2000, thin=2, seed=1000);

pred.effect <- bzPredSubgrp(rst.nes,
    dat.sub = solvd.sub,
    var.estvar = var.estvar);
## End(Not run)
```

---

bzRptTbl  
**Summary table of treatment effects**

Description

Compare the DIC from different models and report the summary of treatment effects based on the model with the smallest DIC value

Usage

bzRptTbl(lst.stan.rst, dat.sub, var.cov, cut = 0, digits = 3)
**bzSummary**

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lst.stan.rst</td>
<td>list of class beanz.stan results from <code>bzCallStan</code> for different models</td>
</tr>
<tr>
<td>dat.sub</td>
<td>dataset with subgroup treatment effect summary data</td>
</tr>
<tr>
<td>var.cov</td>
<td>array of column names in dat.sub that corresponds to binary or ordinal baseline covariates</td>
</tr>
<tr>
<td>cut</td>
<td>cut point to compute the probability that the posterior subgroup treatment effects is below</td>
</tr>
<tr>
<td>digits</td>
<td>number of digits in the summary result table</td>
</tr>
</tbody>
</table>

**Value**

A dataframe with summary statistics of the model selected by DIC

---

**bzShiny**

*Run Web-Based BEANZ application*

**Description**

Call Shiny to run beanz as a web-based application

**Usage**

`bzShiny()`

---

**bzSummary**

*Posterior subgroup treatment effects*

**Description**

Present the posterior subgroup treatment effects

**Usage**

```r
bzSummary(
  stan.rst,
  sel.grps = NULL,
  ref.stan.rst = NULL,
  ref.sel.grps = 1,
  cut = 0,
  digits = 3
)
```

`bzPlot(stan.rst, sel.grps = NULL, ref.stan.rst = NULL, ref.sel.grps = 1, ...)`
bzSummary

bzForest(
  stan.rst,
  sel.grps = NULL,
  ref.stan.rst = NULL,
  ref.sel.grps = 1,
  ..., 
  quants = c(0.025, 0.975)
)

Arguments

  stan.rst         a class beanz.stan object generated by bzCallStan
  sel.grps         an array of subgroup numbers to be included in the summary results
  ref.stan.rst     a class beanz.stan object from bzCallStan that is used as the reference
  ref.sel.grps     subgroups from the reference model to be included in the summary table
  cut              cut point to compute the probability that the posterior subgroup treatment effects is below
  digits           number of digits in the summary result table
  ...              options for plot function
  quants           lower and upper quantiles of the credible intervals in the forest plot

Value

  bzSummary generates a dataframe with summary statistics of the posterior treatment effect for the selected subgroups. bzPlot generates the density plot of the posterior treatment effects for the selected subgroups. bzForest generates the forest plot of the posterior treatment effects.

See Also

  bzCallStan

Examples

  ## Not run:
  sel.grps <- c(1,4,5);
  tbl.sub <- bzSummary(rst.sr, ref.stan.rst=rst.nse, ref.sel.grps=1);
  bzPlot(rst.sr, sel.grps = sel.grps, ref.stan.rst=rst.nse, ref.sel.grps=1);
  bzForest(rst.sr, sel.grps = sel.grps, ref.stan.rst=rst.nse, ref.sel.grps=1);
  ## End(Not run)
Description

Dataset for use in `beanz` examples and vignettes.

Format

A dataframe with 6 variables:

- `trt` treatment assignment
- `y` time to death or first hospitalization
- `censor` censoring status
- `sodium` level of sodium
- `lvef` level of lvef
- `any.vasodilator.use` level of use of vasodilator

Details

Subject level data from SOLVD trial. SOLVD is a randomized controlled trial of the effect of an Angiotensin-converting-enzyme inhibitor (ACE inhibitor) called enalapril on survival in patients with reduced left ventricular ejection fraction and congestive heart failure (CHF).

References

Index

beanz (beanz-package), 2
beanz-package, 2
bzCallStan, 4, 7, 10, 12, 13
bzComp, 6
bzForest (bzSummary), 12
bzForestComp (bzComp), 6
bzGailSimon, 8
bzGetSubgrp, 9
bzGetSubgrpRaw, 9
bzPlot (bzSummary), 12
bzPlotComp (bzComp), 6
bzPredSubgrp, 10
bzRptTbl, 11
bzShiny, 4, 12
bzSummary, 12
bzSummaryComp (bzComp), 6

solvd.sub, 14