Package 'BayesPPD'

May 9, 2023

Title Bayesian Power Prior Design

Version 1.1.1

Date 2023-05-08

Description Bayesian power/type I error calculation and model fitting using
the power prior and the normalized power prior for generalized linear models.
Detailed examples of applying the package are available at <doi:10.32614/RJ-2023-016>.
The Bayesian clinical trial design methodology is described in Chen et al. (2011)
<doi:10.1111/j.1541-0420.2011.01561.x>, and Psioda and Ibrahim (2019)
<doi:10.1093/biostatistics/kxy009>. The normalized power prior is described in Duan et al. (2006)
<doi:10.1002/env.752> and Ibrahim et al. (2015) <doi:10.1002/sim.6728>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.1

LazyData true

Imports Rcpp

LinkingTo Rcpp, RcppArmadillo, RcppEigen, RcppNumerical

Suggests rmarkdown, knitr, testthat (>= 3.0.0), ggplot2, kableExtra

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Yueqi Shen [aut, cre],
Matthew A. Psioda [aut],
Joseph G. Ibrahim [aut]

Maintainer Yueqi Shen <ys137@live.unc.edu>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2023-05-09 02:20:02 UTC

R topics documented:

`BayesPPD-package` *Bayesian sample size determination using the power and normalized power prior for generalized linear models*

Description

The **BayesPPD** (Bayesian Power Prior Design) package provides two categories of functions: functions for Bayesian power/type I error calculation and functions for model fitting. Supported distributions include normal, binary (Bernoulli/binomial), Poisson and exponential. The power parameter a_0 can be fixed or modeled as random using a normalized power prior.

Details

Following Chen et al.(2011), for two group models (i.e., treatment and control group with no covariates), denote the parameter for the treatment group by μ_t and the parameter for the control group by μ_c. Suppose there are K historical datasets $D_0 = (D_{01}, \cdots, D_{0K})'$. We consider the following normalized power prior for μ_c given multiple historical datasets D_0

$$\pi(\mu_c|D_0, a_0) = \frac{1}{C(a_0)} \prod_{k=1}^{K} [L(\mu_c|D_{0k})^{a_{0k}}]\, \pi_0(\mu_c)$$

where $a_0 = (a_{01}, \cdots, a_{0K})'$, $0 \leq a_{0k} \leq 1$ for $k = 1, \cdots, K$, $L(\mu_c|D_{0k})$ is the historical data likelihood, $\pi_0(\mu_c)$ is an initial prior, and $C(a_0) = \int \prod_{k=1}^{K} [L(\mu_c|D_{0k})^{a_{0k}}] \pi_0(\mu_c) d\mu_c$. When a_0 is fixed, the normalized power prior is equivalent to the power prior

$$\pi(\mu_c|D_0, a_0) = \prod_{k=1}^{K} [L(\mu_c|D_{0k})^{a_{0k}}]\, \pi_0(\mu_c).$$

By default, the power/type I error calculation algorithm assumes the null and alternative hypotheses are given by

$$H_0 : \mu_t - \mu_c \geq \delta$$

and

$$H_1 : \mu_t - \mu_c < \delta,$$

where δ is a prespecified constant. To test hypotheses of the opposite direction, i.e., $H_0 : \mu_t - \mu_c \leq \delta$ and $H_1 : \mu_t - \mu_c > \delta$, one can set the parameter `nullspace.ineq` to "<". To determine Bayesian sample size, we estimate the quantity

$$\beta_{sj}^{(n)} = E_s[I\{P(\mu_t - \mu_c < \delta | y^{(n)}, \pi^{(f)}) \geq \gamma\}]$$

where $\gamma > 0$ is a prespecified posterior probability threshold for rejecting the null hypothesis (e.g., 0.975), the probability is computed with respect to the posterior distribution given the data $y^{(n)}$ and the fitting prior $\pi^{(f)}$, and the expectation is taken with respect to the marginal distribution of $y^{(n)}$ defined based on the sampling prior $\pi^{(s)}(\theta)$, where $\theta = (\mu_t, \mu_c, \eta)$ and η denotes any nuisance parameter in the model. Let Θ_0 and Θ_1 denote the parameter spaces corresponding to H_0 and H_1. Let $\pi_0^{(s)}(\theta)$ denote a sampling prior that puts mass in the null region, i.e., $\theta \subset \Theta_0$. Let $\pi_1^{(s)}(\theta)$ denote a sampling prior that puts mass in the alternative region, i.e., $\theta \subset \Theta_1$. Then $\beta_{s0}^{(n)}$ corresponding to $\pi^{(s)}(\theta) = \pi_0^{(s)}(\theta)$ is a Bayesian type I error, while $\beta_{s1}^{(n)}$ corresponding to $\pi^{(s)}(\theta) = \pi_1^{(s)}(\theta)$ is a Bayesian power. We compute $n_{\alpha_0} = \min\{n : \beta_{s0}^{(n)} \leq \alpha_0\}$ and $n_{\alpha_1} = \min\{n : \beta_{s1}^{(n)} \geq 1 - \alpha_1\}$. Then Bayesian sample size is $\max\{n_{\alpha_0}, n_{\alpha_1}\}$. Choosing $\alpha_0 = 0.05$ and $\alpha_1 = 0.2$ guarantees that the Bayesian type I error rate is at most 0.05 and the Bayesian power is at least 0.8.

To compute $\beta_{sj}^{(n)}$, the following algorithm is used:

Step 1: Generate $\theta \sim \pi_j^{(s)}(\theta)$

Step 2: Generate $y^{(n)} \sim f(y^{(n)}|\theta)$

Step 3: Compute $P(\mu_t < \mu_c + \delta | y^{(n)}, \pi^{(f)})$

Step 4: Check whether $P(\mu_t < \mu_c + \delta | y^{(n)}, \pi^{(f)}) \geq \gamma$

Step 5: Repeat Steps 1-4 N times

Step 6: Compute the proportion of times that $\{\mu_t < \mu_c + \delta | y^{(n)}, \pi^{(f)} \geq \gamma\}$ is true out of the N simulated datasets, which gives an estimate of $\beta_{sj}^{(n)}$.

For positive continuous data assumed to follow exponential distribution, the hypotheses are given by

$$H_0 : \mu_t / \mu_c \geq \delta$$

and

$$H_1 : \mu_t / \mu_c < \delta,$$

where μ_t and μ_c are the hazards for the treatment and the control group, respectively. The definition of $\beta_{sj}^{(n)}$ and the algorithm change accordingly.

If there are covariates to adjust for, we assume the first column of the covariate matrix is the treatment indicator, and the corresponding parameter is β_1, which, for example, corresponds to a difference in means for the linear regression model and a log hazard ratio for the exponential regression model. The hypotheses are given by

$$H_0 : \beta_1 \geq \delta$$

and

$$H_1 : \beta_1 < \delta.$$

The definition of $\beta_{sj}^{(n)}$ and the algorithm change accordingly.

By default, the package assumes the historical data is composed of control group subjects only. If the user wants to use historical data to inform treatment effect, one can set `borrow.treat=TRUE` and include the treatment indicator in the historical covariate matrix.

This implementation of the method does not assume any particular distribution for the sampling priors. The user is allowed to specify a vector or matrix of samples for θ (matrix if θ is of dimension >1) from any distribution, and the algorithm samples with replacement from the vector or matrix at each iteration of data simulation. In order to accurately approximate a joint distribution for multiple parameters, the number of iterations should be large (e.g., 10,000).

Gibbs sampling is used for normally distributed data. Slice sampling is used for all other data distributions. For two group models with fixed a_0, numerical integration using the **RcppNumerical** package is used.

References

Chen, Ming-Hui, et al. "Bayesian design of noninferiority trials for medical devices using historical data." Biometrics 67.3 (2011): 1163-1170.

`actg019` *AIDS Clinical Trial ACTG019 (1990).*

Description

A dataset containing the ACTG019 clinical trial placebo group data (1990) in adults with asymptomatic HIV.

Usage

```
actg019
```

Format

A data frame with 404 rows and 4 variables:

outcome binary variable with 1 indicating death, development of AIDS or ARC and 0 otherwise

age patient age in years

race binary variable with 1 indicating white and 0 otherwise

T4count CD4 cell count (cell count per cubicmillimetre of serum)

Source

Chen, Ming-Hui, et al. "Prior Elicitation, Variable Selection and Bayesian Computation for Logistic Regression Models." Journal of the Royal Statistical Society. Series B, vol. 61, no. 1, 1999, pp. 223-242.

actg036 *AIDS Clinical Trial ACTG036 (1991).*

Description

A dataset containing the ACTG036 clinical trial data (1991) comparing zidovudine (AZT) with a placebo in asymptomatic patients with hereditary coagulation disorders and HIV infection. The ACTG036 trial had the same response variable and covariates as the ACTG019 study. The ATCG019 data can be used as a historical dataset.

Usage

```
actg036
```

Format

A data frame with 183 rows and 5 variables:

outcome binary variable with 1 indicating death, development of AIDS or ARC and 0 otherwise

treat binary variable with 1 indicating Zidovudine (AZT) treatment and 0 indicating placebo

age patient age in years

race binary variable with 1 indicating white and 0 otherwise

T4count CD4 cell count (cell count per cubicmillimetre of serum)

Source

Chen, Ming-Hui, et al. "Prior Elicitation, Variable Selection and Bayesian Computation for Logistic Regression Models." Journal of the Royal Statistical Society. Series B, vol. 61, no. 1, 1999, pp. 223-242.

glm.fixed.a0 *Model fitting for generalized linear models with fixed a0*

Description

Model fitting using power priors for generalized linear models with fixed a_0

Usage

```
glm.fixed.a0(
  data.type,
  data.link,
  y = 0,
  n = 1,
  x = matrix(),
  borrow.treat = FALSE,
  historical = list(),
  lower.limits = rep(-100, 50),
  upper.limits = rep(100, 50),
  slice.widths = rep(1, 50),
  nMC = 10000,
  nBI = 250,
  current.data = TRUE,
  prior.beta.var = rep(10, 50)
)
```

Arguments

`data.type`	Character string specifying the type of response. The options are "Normal", "Bernoulli", "Binomial", "Poisson" and "Exponential".
`data.link`	Character string specifying the link function. The options are "Logistic", "Probit", "Log", "Identity-Positive", "Identity-Probability" and "Complementary Log-Log". Does not apply if `data.type` is "Normal".
`y`	Vector of responses.
`n`	(For binomial data only) vector of integers specifying the number of subjects who have a particular value of the covariate vector. If the data is binary and all covariates are discrete, collapsing Bernoulli data into a binomial structure can make the slice sampler much faster.
`x`	Matrix of covariates. The first column should be the treatment indicator with 1 indicating treatment group. The number of rows should equal the length of the response vector `y`.
`borrow.treat`	Logical value indicating whether the historical information is used to inform the treatment effect parameter. The default value is FALSE. If TRUE, the first column of the historical covariate matrix must be the treatment indicator. If FALSE, the historical covariate matrix must NOT have the treatment indicator, since the historical data is assumed to be from the control group only.
`historical`	(Optional) list of historical dataset(s). East historical dataset is stored in a list which contains three *named* elements: `y0`, `x0` and `a0`. • `y0` is a vector of responses. • `x0` is a matrix of covariates. If `borrow.treat` is FALSE (the default), `x0` should NOT have the treatment indicator. Apart from missing the treatment indicator, `x0` should have the same set of covariates in the same order as `x`. If `borrow.treat` is TRUE, `x0` should have the same set of covariates in the same order as `x`, where the first column of `x0` must be the treatment indicator.

- a0 is a number between 0 and 1 indicating the discounting parameter value for that historical dataset.

lower.limits — Vector of lower limits for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is -100 for all parameters (may not be appropriate for all situations). Does not apply if data.type is "Normal".

upper.limits — Vector of upper limits for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is 100 for all parameters (may not be appropriate for all situations). Does not apply if data.type is "Normal".

slice.widths — Vector of initial slice widths for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is 1 for all parameter (may not be appropriate for all situations). Does not apply if data.type is "Normal".

nMC — Number of iterations (excluding burn-in samples) for the slice sampler or Gibbs sampler. The default is 10,000.

nBI — Number of burn-in samples for the slice sampler or Gibbs sampler. The default is 250.

current.data — Logical value indicating whether current data is included. The default is TRUE. If FALSE, only historical data is included in the analysis, and the posterior samples can be used as a discrete approximation to the sampling prior in power.glm.fixed.a0.

prior.beta.var — Only applies if current.data = FALSE. If no current data is provided, the initial priors used for β are i.i.d. normal distributions with mean zero and variance equal to prior.beta.var. The length of the vector should be equal to the length of β. The default variance is 10.

Details

If data.type is "Normal", the response y_i is assumed to follow $N(x_i'\beta, \tau^{-1})$ where x_i is the vector of covariates for subject i. Each historical dataset D_{0k} is assumed to have a different precision parameter τ_k. The initial prior for τ is the Jeffery's prior, τ^{-1}, and the initial prior for τ_k is τ_k^{-1}. The initial prior for β is the uniform improper prior. Posterior samples are obtained through Gibbs sampling.

For all other data types, posterior samples are obtained through slice sampling. The default lower limits for the parameters are -100. The default upper limits for the parameters are 100. The default slice widths for the parameters are 1. The defaults may not be appropriate for all situations, and the user can specify the appropriate limits and slice width for each parameter.

When current.data is set to FALSE, only historical data is included in the analysis, and the posterior samples can be used as a discrete approximation to the sampling prior in power.glm.fixed.a0.

Value

The function returns a S3 object with a summary method. If data.type is "Normal", posterior samples of β, τ and τ_k's (if historical data is given) are returned. For all other data types, a matrix of posterior samples of β is returned. The first column contains posterior samples of the intercept. The second column contains posterior samples of β_1, the parameter for the treatment indicator.

References

Neal, Radford M. Slice sampling. Ann. Statist. 31 (2003), no. 3, 705–767.

See Also

power.glm.fixed.a0

Examples

```
data.type <- "Bernoulli"
data.link <- "Logistic"

# Simulate current data
set.seed(1)
p <- 3
n_total <- 100
y <- rbinom(n_total,size=1,prob=0.6)
# The first column of x is the treatment indicator.
x <- cbind(rbinom(n_total,size=1,prob=0.5),
           matrix(rnorm(p*n_total),ncol=p,nrow=n_total))

# Simulate two historical datasets
# Note that x0 does not have the treatment indicator
historical <- list(list(y0=rbinom(n_total,size=1,prob=0.2),
                        x0=matrix(rnorm(p*n_total),ncol=p,nrow=n_total), a0=0.2),
                   list(y0=rbinom(n_total, size=1, prob=0.5),
                        x0=matrix(rnorm(p*n_total),ncol=p,nrow=n_total), a0=0.3))

# Set parameters of the slice sampler
lower.limits <- rep(-100, 5) # The dimension is the number of columns of x plus 1 (intercept)
upper.limits <- rep(100, 5)
slice.widths <- rep(1, 5)

nMC <- 1000 # nMC should be larger in practice
nBI <- 250
result <- glm.fixed.a0(data.type=data.type, data.link=data.link, y=y, x=x, historical=historical,
                       lower.limits=lower.limits, upper.limits=upper.limits,
                       slice.widths=slice.widths, nMC=nMC, nBI=nBI)

summary(result)
```

`glm.random.a0` *Model fitting for generalized linear models with random a0*

Description

Model fitting using normalized power priors for generalized linear models with random a_0

Usage

```
glm.random.a0(
  data.type,
  data.link,
  y,
  n = 1,
  x,
  borrow.treat = FALSE,
  historical,
  prior.beta.var = rep(10, 50),
  prior.a0.shape1 = rep(1, 10),
  prior.a0.shape2 = rep(1, 10),
  a0.coefficients,
  lower.limits = rep(-100, 50),
  upper.limits = rep(100, 50),
  slice.widths = rep(0.1, 50),
  nMC = 10000,
  nBI = 250
)
```

Arguments

`data.type`	Character string specifying the type of response. The options are "Normal", "Bernoulli", "Binomial", "Poisson" and "Exponential".
`data.link`	Character string specifying the link function. The options are "Logistic", "Probit", "Log", "Identity-Positive", "Identity-Probability" and "Complementary Log-Log". Does not apply if `data.type` is "Normal".
`y`	Vector of responses.
`n`	(For binomial data only) vector of integers specifying the number of subjects who have a particular value of the covariate vector. If the data is binary and all covariates are discrete, collapsing Bernoulli data into a binomial structure can make the slice sampler much faster.
`x`	Matrix of covariates. The first column should be the treatment indicator with 1 indicating treatment group. The number of rows should equal the length of the response vector `y`.
`borrow.treat`	Logical value indicating whether the historical information is used to inform the treatment effect parameter. The default value is FALSE. If TRUE, the first column of the historical covariate matrix must be the treatment indicator. If FALSE, the historical covariate matrix must NOT have the treatment indicator, since the historical data is assumed to be from the control group only.
`historical`	List of historical dataset(s). East historical dataset is stored in a list which contains two *named* elements: `y0` and `x0`. • `y0` is a vector of responses. • `x0` is a matrix of covariates. If `borrow.treat` is FALSE (the default), `x0` should NOT have the treatment indicator. Apart from missing the treatment indicator, `x0` should have the same set of covariates in the same order as

x. If borrow.treat is TRUE, x0 should have the same set of covariates in the same order as x, where the first column of x0 must be the treatment indicator.

For binomial data, an additional element n0 is required.

- n0 is vector of integers specifying the number of subjects who have a particular value of the covariate vector.

prior.beta.var Vector of variances of the independent normal initial priors on β with mean zero. The length of the vector should be equal to the length of β. The default variance is 10.

prior.a0.shape1

Vector of the first shape parameters of the independent beta priors for a_0. The length of the vector should be equal to the number of historical datasets. The default is a vector of one's.

prior.a0.shape2

Vector of the second shape parameters of the independent beta priors for a_0. The length of the vector should be equal to the number of historical datasets. The default is a vector of one's.

a0.coefficients

Vector of coefficients for a_0 returned by the function normalizing.constant. This is necessary for estimating the normalizing constant for the normalized power prior. Does not apply if data.type is "Normal".

lower.limits Vector of lower limits for parameters to be used by the slice sampler. If data.type is "Normal", slice sampling is used for a_0, and the length of the vector should be equal to the number of historical datasets. For all other data types, slice sampling is used for β and a_0. The first P+1 elements apply to the sampling of β and the rest apply to the sampling of a_0. The length of the vector should be equal to the sum of the total number of parameters (i.e. P+1 where P is the number of covariates) and the number of historical datasets. The default is -100 for all parameters (may not be appropriate for all situations).

upper.limits Vector of upper limits for parameters to be used by the slice sampler. If data.type is "Normal", slice sampling is used for a_0, and the length of the vector should be equal to the number of historical datasets. For all other data types, slice sampling is used for β and a_0. The first P+1 elements apply to the sampling of β and the rest apply to the sampling of a_0. The length of the vector should be equal to the sum of the total number of parameters (i.e. P+1 where P is the number of covariates) and the number of historical datasets. The default is 100 for all parameters (may not be appropriate for all situations).

slice.widths Vector of initial slice widths used by the slice sampler. If data.type is "Normal", slice sampling is used for a_0, and the length of the vector should be equal to the number of historical datasets. For all other data types, slice sampling is used for β and a_0. The first P+1 elements apply to the sampling of β and the rest apply to the sampling of a_0. The length of the vector should be equal to the sum of the total number of parameters (i.e. P+1 where P is the number of covariates) and the number of historical datasets. The default is 0.1 for all parameter (may not be appropriate for all situations).

nMC Number of iterations (excluding burn-in samples) for the slice sampler or Gibbs sampler. The default is 10,000.

nBI — Number of burn-in samples for the slice sampler or Gibbs sampler. The default is 250.

Details

The user should use the function normalizing.constant to obtain a0.coefficients (does not apply if data.type is "Normal").

If data.type is "Normal", the response y_i is assumed to follow $N(x_i'\beta, \tau^{-1})$ where x_i is the vector of covariates for subject i. Historical datasets are assumed to have the same precision parameter as the current dataset for computational simplicity. The initial prior for τ is the Jeffery's prior, τ^{-1}. Independent normal priors with mean zero and variance prior.beta.var are used for β to ensure the propriety of the normalized power prior. Posterior samples for β and τ are obtained through Gibbs sampling. Independent beta(prior.a0.shape1, prior.a0.shape1) priors are used for a_0. Posterior samples for a_0 are obtained through slice sampling.

For all other data types, posterior samples are obtained through slice sampling. The default lower limits for the parameters are -100. The default upper limits for the parameters are 100. The default slice widths for the parameters are 0.1. The defaults may not be appropriate for all situations, and the user can specify the appropriate limits and slice width for each parameter.

Value

The function returns a S3 object with a summary method. If data.type is "Normal", posterior samples of β, τ and a_0 are returned. For all other data types, posterior samples of β and a_0 are returned. The first column of the matrix of posterior samples of β contains posterior samples of the intercept. The second column contains posterior samples of β_1, the parameter for the treatment indicator.

References

Neal, Radford M. Slice sampling. Ann. Statist. 31 (2003), no. 3, 705–767.

See Also

normalizing.constant and power.glm.random.a0

Examples

```
data.type <- "Bernoulli"
data.link <- "Logistic"

# Simulate current data
set.seed(1)
p <- 3
n_total <- 100
y <- rbinom(n_total,size=1,prob=0.6)
# The first column of x is the treatment indicator.
x <- cbind(rbinom(n_total,size=1,prob=0.5),
           matrix(rnorm(p*n_total),ncol=p,nrow=n_total))
```

```
# Simulate two historical datasets
# Note that x0 does not have the treatment indicator
historical <- list(list(y0=rbinom(n_total,size=1,prob=0.2),
                        x0=matrix(rnorm(p*n_total),ncol=p,nrow=n_total)),
                   list(y0=rbinom(n_total, size=1, prob=0.5),
                        x0=matrix(rnorm(p*n_total),ncol=p,nrow=n_total)))

# Please see function "normalizing.constant" for how to obtain a0.coefficients
# Here, suppose one-degree polynomial regression is chosen by the "normalizing.constant"
# function. The coefficients are obtained for the intercept, a0_1 and a0_2.
a0.coefficients <- c(1, 0.5, -1)

# Set parameters of the slice sampler
# The dimension is the number of columns of x plus 1 (intercept)
# plus the number of historical datasets
lower.limits <- rep(-100, 7)
upper.limits <- rep(100, 7)
slice.widths <- rep(0.1, 7)

nMC <- 500 # nMC should be larger in practice
nBI <- 100
result <- glm.random.a0(data.type=data.type, data.link=data.link, y=y, x=x,
                        historical=historical, a0.coefficients=a0.coefficients,
                        lower.limits=lower.limits, upper.limits=upper.limits,
                        slice.widths=slice.widths, nMC=nMC, nBI=nBI)
summary(result)
```

`normalizing.constant` *Function for approximating the normalizing constant for generalized linear models with random a0*

Description

This function returns a vector of coefficients that defines a function $f(a_0)$ that approximates the normalizing constant for generalized linear models with random a_0. The user should input the values returned to `glm.random.a0` or `power.glm.random.a0`.

Usage

```
normalizing.constant(
  grid,
  historical,
  data.type,
  data.link,
  prior.beta.var = rep(10, 50),
  lower.limits = rep(-100, 50),
  upper.limits = rep(100, 50),
  slice.widths = rep(1, 50),
```

```
  nMC = 10000,
  nBI = 250
)
```

Arguments

grid	Matrix of potential values for a_0, where the number of columns should equal the number of historial datasets. Note that the algorithm may fail if some grid values are close to zero. See *Details* below.
historical	List of historical dataset(s). East historical dataset is stored in a list which constains two *named* elements: `y0` and `x0`. • `y0` is a vector of responses. • `x0` is a matrix of covariates. For binomial data, an additional element `n0` is required. • `n0` is vector of integers specifying the number of subjects who have a particular value of the covariate vector.
data.type	Character string specifying the type of response. The options are "Bernoulli", "Binomial", "Poisson" and "Exponential".
data.link	Character string specifying the link function. The options are "Logistic", "Probit", "Log", "Identity-Positive", "Identity-Probability" and "Complementary Log-Log". Does not apply if `data.type` is "Normal".
prior.beta.var	Vector of variances of the independent normal initial priors on β with mean zero. The length of the vector should be equal to the length of β. The default variance is 10.
lower.limits	Vector of lower limits for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is -100 for all parameters (may not be appropriate for all situations). Does not apply if `data.type` is "Normal".
upper.limits	Vector of upper limits for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is 100 for all parameters (may not be appropriate for all situations). Does not apply if `data.type` is "Normal".
slice.widths	Vector of initial slice widths for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is 1 for all parameter (may not be appropriate for all situations). Does not apply if `data.type` is "Normal".
nMC	Number of iterations (excluding burn-in samples) for the slice sampler or Gibbs sampler. The default is 10,000.
nBI	Number of burn-in samples for the slice sampler or Gibbs sampler. The default is 250.

Details

This function performs the following steps:

1. Suppose there are K historical datasets. The user inputs a grid of M rows and K columns of potential values for a_0. For example, one can choose the vector `v = c(0.1, 0.25, 0.5, 0.75, 1)` and use `expand.grid(a0_1=v, a0_2=v, a0_3=v)` when $K = 3$ to get a grid with $M = 5^3 = 125$ rows and 3 columns. If there are more than three historical datasets, the dimension of `v` can be reduced to limit the size of the grid. A large grid will increase runtime.
2. For each row of a_0 values in the grid, obtain M samples for β from the power prior associated with the current values of a_0 using the slice sampler.
3. For each of the M sets of posterior samples, execute the PWK algorithm (Wang et al., 2018) to estimate the log of normalizing constant $d_1, ..., d_M$ for the normalized power prior.
4. At this point, one has a dataset with outcomes $d_1, ..., d_M$ and predictors corresponding to the rows of the a_0 grid matrix. A polynomial regression is applied to estimate a function $d = f(a0)$. The degree of the polynomial regression is determined by the algorithm to ensure $R^2 > 0.99$.
5. The vector of coefficients from the polynomial regression model is returned by the function, which the user must input into `glm.random.a0` or `power.glm.random.a0`.

When a row of the `grid` contains elements that are close to zero, the resulting power prior will be flat and estimates of normalizing constants may be inaccurate. Therefore, it is recommended that `grid` values should be at least 0.05.

If one encounters the error message "some coefficients are not defined because of singularities", it could be due to the following factors: number of `grid` rows too large or too small, insufficient sample size of the historical data, insufficient number of iterations for the slice sampler, or near-zero `grid` values.

Note that due to computational intensity, the `normalizing.constant` function has not been evaluated for accuracy for high dimensional β (e.g., dimension > 10) or high dimensional a_0 (e.g., dimension > 5).

Value

Vector of coefficients for a_0 that defines a function $f(a_0)$ that approximates the normalizing constant, necessary for functions `glm.random.a0` and `power.glm.random.a0`. The length of the vector is equal to 1+K*L where K is the number of historical datasets and L is the degree of the polynomial regression determined by the algorithm.

References

Wang, Yu-Bo; Chen, Ming-Hui; Kuo, Lynn; Lewis, Paul O. A New Monte Carlo Method for Estimating Marginal Likelihoods. Bayesian Anal. 13 (2018), no. 2, 311–333.

See Also

`glm.random.a0` and `power.glm.random.a0`

Examples

```
data.type <- "Bernoulli"
data.link <- "Logistic"
```

```
data.size <- 50

# Simulate two historical datasets
p <- 1
set.seed(111)
x1 <- matrix(rnorm(p*data.size),ncol=p,nrow=data.size)
set.seed(222)
x2 <- matrix(rnorm(p*data.size),ncol=p,nrow=data.size)
beta <- c(1,2)
mean1 <- exp(x1*beta)/(1+exp(x1*beta))
mean2 <- exp(x2*beta)/(1+exp(x2*beta))
historical <- list(list(y0=rbinom(data.size,size=1,prob=mean1),x0=x1),
                   list(y0=rbinom(data.size, size=1, prob=mean2),x0=x2))

# Create grid of possible values of a0 with two columns corresponding to a0_1 and a0_2
g <- c(0.1, 0.25, 0.5, 0.75, 1)
grid <- expand.grid(a0_1=g, a0_2=g)

nMC <- 100 # nMC should be larger in practice
nBI <- 50
result <- normalizing.constant(grid=grid, historical=historical,
                               data.type=data.type, data.link=data.link,
                               nMC=nMC, nBI=nBI)
```

`power.glm.fixed.a0` *Power/type I error calculation for generalized linear models with fixed a0*

Description

Power/type I error calculation for generalized linear models with fixed a_0 using power priors

Usage

```
power.glm.fixed.a0(
  data.type,
  data.link = "",
  data.size,
  n = 1,
  borrow.treat = FALSE,
  treat.assign.prob = 0.5,
  historical = list(),
  nullspace.ineq = ">",
  x.samples = matrix(),
  samp.prior.beta,
  samp.prior.var = 0,
  lower.limits = rep(-100, 50),
  upper.limits = rep(100, 50),
  slice.widths = rep(1, 50),
```

```
  delta = 0,
  gamma = 0.95,
  nMC = 10000,
  nBI = 250,
  N = 10000,
  approximate = FALSE,
  nNR = 10000,
  tol = 1e-05
)
```

Arguments

`data.type`	Character string specifying the type of response. The options are "Normal", "Bernoulli", "Binomial", "Poisson" and "Exponential".
`data.link`	Character string specifying the link function. The options are "Logistic", "Probit", "Log", "Identity-Positive", "Identity-Probability" and "Complementary Log-Log". Does not apply if `data.type` is "Normal".
`data.size`	Sample size of the simulated datasets.
`n`	(For binomial data only) vector of integers specifying the number of subjects who have a particular value of the covariate vector. If the data is binary and all covariates are discrete, collapsing Bernoulli data into a binomial structure can make the slice sampler much faster.
`borrow.treat`	Logical value indicating whether the historical information is used to inform the treatment effect parameter. The default value is FALSE. If TRUE, the first column of the historical covariate matrix must be the treatment indicator. If FALSE, the historical covariate matrix must NOT have the treatment indicator, since the historical data is assumed to be from the control group only.
`treat.assign.prob`	Probability of being assigned to the treatment group. The default value is 0.5. Only applies if `borrow.treat=FALSE`.
`historical`	(Optional) list of historical dataset(s). East historical dataset is stored in a list which contains three *named* elements: `y0`, `x0` and `a0`. • `y0` is a vector of responses. • `x0` is a matrix of covariates. If `borrow.treat` is FALSE (the default), `x0` should NOT have the treatment indicator. If `borrow.treat` is TRUE, the first column of `x0` must be the treatment indicator. • `a0` is a number between 0 and 1 indicating the discounting parameter value for that historical dataset. For binomial data, an additional element `n0` is required. • `n0` is vector of integers specifying the number of subjects who have a particular value of the covariate vector.
`nullspace.ineq`	Character string specifying the inequality of the null hypothesis. The options are ">" and "<". If ">" is specified, the null hypothesis is H_0: $\beta_1 \geq \delta$. If "<" is specified, the null hypothesis is H_0: $\beta_1 \leq \delta$. The default choice is ">".
`x.samples`	(Only applies when there is no historical dataset) matrix of possible values of covariates from which covariate vectors are sampled with replacement.

`samp.prior.beta`
: Matrix of possible values of β to sample (with replacement) from. Each row is a possible β vector (a realization from the sampling prior for β), where the first element is the coefficient for the intercept and the second element is the coefficient for the treatment indicator. The length of the vector should be equal to the total number of parameters. If P is the number of columns of `x0` in `historical`, the total number of parameters is P+2 if `borrow.treat=FALSE`, and is P+1 if `borrow.treat=TRUE`.

`samp.prior.var`
: Vector of possible values of σ^2 to sample (with replacement) from. Only applies if `data.type` is "Normal". The vector contains realizations from the sampling prior (e.g. inverse-gamma distribution) for σ^2.

`lower.limits`
: Vector of lower limits for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is -100 for all parameters (may not be appropriate for all situations). Does not apply if `data.type` is "Normal".

`upper.limits`
: Vector of upper limits for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is 100 for all parameters (may not be appropriate for all situations). Does not apply if `data.type` is "Normal".

`slice.widths`
: Vector of initial slice widths for parameters to be used by the slice sampler. The length of the vector should be equal to the total number of parameters, i.e. P+1 where P is the number of covariates. The default is 1 for all parameter (may not be appropriate for all situations). Does not apply if `data.type` is "Normal".

`delta`
: Prespecified constant that defines the boundary of the null hypothesis. The default is zero.

`gamma`
: Posterior probability threshold for rejecting the null. The null hypothesis is rejected if posterior probability is greater `gamma`. The default is 0.95.

`nMC`
: Number of iterations (excluding burn-in samples) for the slice sampler or Gibbs sampler. The default is 10,000.

`nBI`
: Number of burn-in samples for the slice sampler or Gibbs sampler. The default is 250.

`N`
: Number of simulated datasets to generate. The default is 10,000.

`approximate`
: Logical value indicating whether the approximation method based on asymptotic theory is used. The default is FALSE. If TRUE, an approximation method based on the Newton-Raphson algorithm (assuming canonical links) is used. This feature helps users quickly obtain a rough estimate of the sample size required for the desired level of power or type I error rate.

`nNR`
: (Only applies if `approximate=TRUE`) number of iterations of the Newton-Raphson algorithm. The default value is 10,000.

`tol`
: (Only applies if `approximate=TRUE`) absolute tolerance of the Newton-Raphson algorithm. The default value is 0.00001.

Details

If historical datasets are provided, the algorithm samples with replacement from the historical covariates to construct the simulated datasets. Otherwise, the algorithm samples with replacement from `x.samples`. One of the arguments `historical` and `x.samples` must be provided.

The sampling prior for the treatment parameter can be generated from a normal distribution (see examples). For example, suppose one wants to compute the power for the hypotheses $H_0 : \beta_1 \geq 0$ and $H_1 : \beta_1 < 0$. To approximate the sampling prior for β_1, one can simply sample from a normal distribution with negative mean, so that the mass of the prior falls in the alternative space. Conversely, to compute the type I error rate, one can sample from a normal distribution with positive mean, so that the mass of the prior falls in the null space. The sampling prior for the other parameters can be generated by using the `glm.fixed.a0` function with `current.data` set to FALSE. The posterior samples based on only historical data can be used as a discrete approximation to the sampling prior.

`samp.prior.var` is necessary for generating normally distributed data.

If `data.type` is "Normal", the response y_i is assumed to follow $N(x_i'\beta, \tau^{-1})$ where x_i is the vector of covariates for subject i. Each historical dataset D_{0k} is assumed to have a different precision parameter τ_k. The initial prior for τ is the Jeffery's prior, τ^{-1}, and the initial prior for τ_k is τ_k^{-1}. The initial prior for β is the uniform improper prior. Posterior samples are obtained through Gibbs sampling.

For all other data types, posterior samples are obtained through slice sampling. The default lower limits for the parameters are -100. The default upper limits for the parameters are 100. The default slice widths for the parameters are 1. The defaults may not be appropriate for all situations, and the user can specify the appropriate limits and slice width for each parameter.

If a sampling prior with support in the null space is used, the value returned is a Bayesian type I error rate. If a sampling prior with support in the alternative space is used, the value returned is a Bayesian power.

Because running `power.glm.fixed.a0()` and `power.glm.random.a0()` is potentially time-consuming, an approximation method based on asymptotic theory has been implemented for the model with fixed a_0. In order to attain the exact sample size needed for the desired power, the user can start with the approximation to get a rough estimate of the sample size required, using `power.glm.fixed.a0()` with `approximate=TRUE`.

Value

The function returns a S3 object with a `summary` method. Power or type I error is returned, depending on the sampling prior used. The posterior probabilities of the alternative hypothesis are returned. The average posterior mean of β and its corresponding bias are returned. If `data.type` is "Normal", average posterior means of τ and τ_k's (if historical data is given) are also returned. The first column of β contains posterior samples of the intercept. The second column contains posterior samples of β_1, the parameter for the treatment indicator.

References

Chen, Ming-Hui, et al. "Bayesian design of noninferiority trials for medical devices using historical data." Biometrics 67.3 (2011): 1163-1170.

Neal, Radford M. Slice sampling. Ann. Statist. 31 (2003), no. 3, 705–767.

See Also

`glm.fixed.a0`

Examples

```
data.type <- "Bernoulli"
data.link <- "Logistic"
data.size <- 100

# Simulate two historical datasets
p <- 3
historical <- list(list(y0=rbinom(data.size,size=1,prob=0.2),
                        x0=matrix(rnorm(p*data.size),ncol=p,nrow=data.size), a0=0.2),
                   list(y0=rbinom(data.size, size=1, prob=0.5),
                        x0=matrix(rnorm(p*data.size),ncol=p,nrow=data.size), a0=0.3))

# Generate sampling priors

# The null hypothesis here is H0: beta_1 >= 0. To calculate power,
# we can provide samples of beta_1 such that the mass of beta_1 < 0.
# To calculate type I error, we can provide samples of beta_1 such that
# the mass of beta_1 >= 0.
samp.prior.beta1 <- rnorm(100, mean=-3, sd=1)
# Here, mass is put on the alternative region, so power is calculated.
samp.prior.beta <- cbind(rnorm(100), samp.prior.beta1, matrix(rnorm(100*p), 100, p))

nMC <- 100 # nMC should be larger in practice
nBI <- 50
N <- 5 # N should be larger in practice
result <- power.glm.fixed.a0(data.type=data.type, data.link=data.link,
                             data.size=data.size, historical=historical,
                             samp.prior.beta=samp.prior.beta,
                             delta=0, nMC=nMC, nBI=nBI, N=N)
summary(result)
```

`power.glm.random.a0` *Power/type I error calculation for generalized linear models with random a0*

Description

Power/type I error calculation using normalized power priors for generalized linear models with random a_0

Usage

```
power.glm.random.a0(
  data.type,
  data.link,
  data.size,
  n = 1,
```

```
  treat.assign.prob = 0.5,
  borrow.treat = FALSE,
  historical,
  nullspace.ineq = ">",
  samp.prior.beta,
  samp.prior.var,
  prior.beta.var = rep(10, 50),
  prior.a0.shape1 = rep(1, 10),
  prior.a0.shape2 = rep(1, 10),
  a0.coefficients,
  lower.limits = rep(-100, 50),
  upper.limits = rep(100, 50),
  slice.widths = rep(0.1, 50),
  delta = 0,
  gamma = 0.95,
  nMC = 10000,
  nBI = 250,
  N = 10000
)
```

Arguments

`data.type`	Character string specifying the type of response. The options are "Normal", "Bernoulli", "Binomial", "Poisson" and "Exponential".
`data.link`	Character string specifying the link function. The options are "Logistic", "Probit", "Log", "Identity-Positive", "Identity-Probability" and "Complementary Log-Log". Does not apply if `data.type` is "Normal".
`data.size`	Sample size of the simulated datasets.
`n`	(For binomial data only) vector of integers specifying the number of subjects who have a particular value of the covariate vector. If the data is binary and all covariates are discrete, collapsing Bernoulli data into a binomial structure can make the slice sampler much faster.
`treat.assign.prob`	Probability of being assigned to the treatment group. The default value is 0.5. Only applies if `borrow.treat=FALSE`.
`borrow.treat`	Logical value indicating whether the historical information is used to inform the treatment effect parameter. The default value is FALSE. If TRUE, the first column of the historical covariate matrix must be the treatment indicator. If FALSE, the historical covariate matrix must NOT have the treatment indicator, since the historical data is assumed to be from the control group only.
`historical`	List of historical dataset(s). East historical dataset is stored in a list which contains two *named* elements: `y0` and `x0`. • `y0` is a vector of responses. • `x0` is a matrix of covariates. If `borrow.treat` is FALSE (the default), `x0` should NOT have the treatment indicator. If `borrow.treat` is TRUE, the first column of `x0` must be the treatment indicator.

For binomial data, an additional element `n0` is required.

- `n0` is vector of integers specifying the number of subjects who have a particular value of the covariate vector.

`nullspace.ineq` Character string specifying the inequality of the null hypothesis. The options are ">" and "<". If ">" is specified, the null hypothesis is H_0: $\beta_1 \geq \delta$. If "<" is specified, the null hypothesis is H_0: $\beta_1 \leq \delta$. The default choice is ">".

`samp.prior.beta`

Matrix of possible values of β to sample (with replacement) from. Each row is a possible β vector (a realization from the sampling prior for β), where the first element is the coefficient for the intercept and the second element is the coefficient for the treatment indicator. The length of the vector should be equal to the total number of parameters. If P is the number of columns of `x0` in `historical`, the total number of parameters is P+2 if `borrow.treat=FALSE`, and is P+1 if `borrow.treat=TRUE`.

`samp.prior.var` Vector of possible values of σ^2 to sample (with replacement) from. Only applies if `data.type` is "Normal". The vector contains realizations from the sampling prior (e.g. inverse-gamma distribution) for σ^2.

`prior.beta.var` Vector of variances of the independent normal initial priors on β with mean zero. The length of the vector should be equal to the length of β. The default variance is 10.

`prior.a0.shape1`

Vector of the first shape parameters of the independent beta priors for a_0. The length of the vector should be equal to the number of historical datasets. The default is a vector of one's.

`prior.a0.shape2`

Vector of the second shape parameters of the independent beta priors for a_0. The length of the vector should be equal to the number of historical datasets. The default is a vector of one's.

`a0.coefficients`

Vector of coefficients for a_0 returned by the function `normalizing.constant`. This is necessary for estimating the normalizing constant for the normalized power prior. Does not apply if `data.type` is "Normal".

`lower.limits` Vector of lower limits for parameters to be used by the slice sampler. If `data.type` is "Normal", slice sampling is used for a_0, and the length of the vector should be equal to the number of historical datasets. For all other data types, slice sampling is used for β and a_0. The first P+1 elements apply to the sampling of β and the rest apply to the sampling of a_0. The length of the vector should be equal to the sum of the total number of parameters (i.e. P+1 where P is the number of covariates) and the number of historical datasets. The default is -100 for all parameters (may not be appropriate for all situations).

`upper.limits` Vector of upper limits for parameters to be used by the slice sampler. If `data.type` is "Normal", slice sampling is used for a_0, and the length of the vector should be equal to the number of historical datasets. For all other data types, slice sampling is used for β and a_0. The first P+1 elements apply to the sampling of β and the rest apply to the sampling of a_0. The length of the vector should be equal to the sum of the total number of parameters (i.e. P+1 where P is the number

of covariates) and the number of historical datasets. The default is 100 for all parameters (may not be appropriate for all situations).

`slice.widths` Vector of initial slice widths used by the slice sampler. If `data.type` is "Normal", slice sampling is used for a_0, and the length of the vector should be equal to the number of historical datasets. For all other data types, slice sampling is used for β and a_0. The first P+1 elements apply to the sampling of β and the rest apply to the sampling of a_0. The length of the vector should be equal to the sum of the total number of parameters (i.e. P+1 where P is the number of covariates) and the number of historical datasets. The default is 0.1 for all parameter (may not be appropriate for all situations).

`delta` Prespecified constant that defines the boundary of the null hypothesis. The default is zero.

`gamma` Posterior probability threshold for rejecting the null. The null hypothesis is rejected if posterior probability is greater `gamma`. The default is 0.95.

`nMC` Number of iterations (excluding burn-in samples) for the slice sampler or Gibbs sampler. The default is 10,000.

`nBI` Number of burn-in samples for the slice sampler or Gibbs sampler. The default is 250.

`N` Number of simulated datasets to generate. The default is 10,000.

Details

The user should use the function `normalizing.constant` to obtain `a0.coefficients` (does not apply if `data.type` is "Normal").

The sampling prior for the treatment parameter can be generated from a normal distribution (see examples). For example, suppose one wants to compute the power for the hypotheses $H_0 : \beta_1 \geq 0$ and $H_1 : \beta_1 < 0$. To approximate the sampling prior for β_1, one can simply sample from a normal distribution with negative mean, so that the mass of the prior falls in the alternative space. Conversely, to compute the type I error rate, one can sample from a normal distribution with positive mean, so that the mass of the prior falls in the null space. The sampling prior for the other parameters can be generated by using the `glm.fixed.a0` function with `current.data` set to FALSE. The posterior samples based on only historical data can be used as a discrete approximation to the sampling prior.

`samp.prior.var` is necessary for generating normally distributed data.

If `data.type` is "Normal", the response y_i is assumed to follow $N(x_i'\beta, \tau^{-1})$ where x_i is the vector of covariates for subject i. Historical datasets are assumed to have the same precision parameter as the current dataset for computational simplicity. The initial prior for τ is the Jeffery's prior, τ^{-1}. Independent normal priors with mean zero and variance `prior.beta.var` are used for β to ensure the propriety of the normalized power prior. Posterior samples for β and τ are obtained through Gibbs sampling. Independent beta(`prior.a0.shape1`, `prior.a0.shape1`) priors are used for a_0. Posterior samples for a_0 are obtained through slice sampling.

For all other data types, posterior samples are obtained through slice sampling. The default lower limits for the parameters are -100. The default upper limits for the parameters are 100. The default slice widths for the parameters are 0.1. The defaults may not be appropriate for all situations, and the user can specify the appropriate limits and slice width for each parameter.

If a sampling prior with support in the null space is used, the value returned is a Bayesian type I error rate. If a sampling prior with support in the alternative space is used, the value returned is a Bayesian power.

Because running `power.glm.fixed.a0()` and `power.glm.random.a0()` is potentially time-consuming, an approximation method based on asymptotic theory has been implemented for the model with fixed a_0. In order to attain the exact sample size needed for the desired power, the user can start with the approximation to get a rough estimate of the sample size required, using `power.glm.fixed.a0()` with `approximate=TRUE`.

Value

The function returns a S3 object with a `summary` method. Power or type I error is returned, depending on the sampling prior used. The posterior probabilities of the alternative hypothesis are returned. The average posterior mean of β and its corresponding bias are returned. The average posterior mean of a_0 is returned. If `data.type` is "Normal", the average posterior mean of τ is also returned. The first element of the average posterior means of β is the average posterior mean of the intercept. The second element is the average posterior mean of β_1, the parameter for the treatment indicator.

References

Chen, Ming-Hui, et al. "Bayesian design of noninferiority trials for medical devices using historical data." Biometrics 67.3 (2011): 1163-1170.

Neal, Radford M. Slice sampling. Ann. Statist. 31 (2003), no. 3, 705–767.

See Also

`normalizing.constant` and `glm.random.a0`

Examples

```
data.type <- "Bernoulli"
data.link <- "Logistic"
data.size <- 100

# Simulate two historical datasets
p <- 3
historical <- list(list(y0=rbinom(data.size,size=1,prob=0.2),
                        x0=matrix(rnorm(p*data.size),ncol=p,nrow=data.size)),
                   list(y0=rbinom(data.size, size=1, prob=0.5),
                        x0=matrix(rnorm(p*data.size),ncol=p,nrow=data.size)))

# Generate sampling priors

# The null hypothesis here is H0: beta_1 >= 0. To calculate power,
# we can provide samples of beta_1 such that the mass of beta_1 < 0.
# To calculate type I error, we can provide samples of beta_1 such that
# the mass of beta_1 >= 0.
samp.prior.beta1 <- rnorm(100, mean=-3, sd=1)
```

```
# Here, mass is put on the alternative region, so power is calculated.
samp.prior.beta <- cbind(rnorm(100), samp.prior.beta1, matrix(rnorm(100*p), 100, p))

# Please see function "normalizing.constant" for how to obtain a0.coefficients
# Here, suppose one-degree polynomial regression is chosen by the "normalizing.constant"
# function. The coefficients are obtained for the intercept, a0_1 and a0_2.
a0.coefficients <- c(1, 0.5, -1)

nMC <- 100 # nMC should be larger in practice
nBI <- 50
N <- 3 # N should be larger in practice
result <- power.glm.random.a0(data.type=data.type, data.link=data.link,
                              data.size=data.size, historical=historical,
                        samp.prior.beta=samp.prior.beta, a0.coefficients=a0.coefficients,
                              delta=0, nMC=nMC, nBI=nBI, N=N)
summary(result)
```

`power.two.grp.fixed.a0` *Power/type I error calculation for data with two groups (treatment and control group, no covariates) with fixed a0*

Description

Power/type I error calculation for data with two groups (treatment and control group, no covariates) with fixed a_0 using power priors

Usage

```
power.two.grp.fixed.a0(
  data.type,
  n.t,
  n.c,
  historical = matrix(0, 1, 4),
  nullspace.ineq = ">",
  samp.prior.mu.t,
  samp.prior.mu.c,
  samp.prior.var.t,
  samp.prior.var.c,
  prior.mu.t.shape1 = 1,
  prior.mu.t.shape2 = 1,
  prior.mu.c.shape1 = 1,
  prior.mu.c.shape2 = 1,
  delta = 0,
  gamma = 0.95,
  nMC = 10000,
  nBI = 250,
```

```
  N = 10000
)
```

Arguments

data.type	Character string specifying the type of response. The options are "Normal", "Bernoulli", "Poisson" and "Exponential".
n.t	Sample size of the treatment group for the simulated datasets.
n.c	Sample size of the control group for the simulated datasets.
historical	(Optional) matrix of historical dataset(s). If data.type is "Normal", historical is a matrix with four columns: • The first column contains the sum of responses for the control group. • The second column contains the sample size of the control group. • The third column contains the sample variance of responses for the control group. • The fourth column contains the discounting parameter value a_0 (between 0 and 1). For all other data types, historical is a matrix with three columns: • The first column contains the sum of responses for the control group. • The second column contains the sample size of the control group. • The third column contains the discounting parameter value a_0 (between 0 and 1). Each row represents a historical dataset.
nullspace.ineq	Character string specifying the inequality of the null hypothesis. The options are ">" and "<". If ">" is specified, the null hypothesis (for non-exponential data) is H_0: μ_t - $\mu_c \geq \delta$. If "<" is specified, the null hypothesis is H_0: μ_t - $\mu_c \leq \delta$. The default choice is ">".
samp.prior.mu.t	Vector of possible values of μ_t to sample (with replacement) from. The vector contains realizations from the sampling prior (e.g. normal distribution) for μ_t.
samp.prior.mu.c	Vector of possible values of μ_c to sample (with replacement) from. The vector contains realizations from the sampling prior (e.g. normal distribution) for μ_c.
samp.prior.var.t	Vector of possible values of σ_t^2 to sample (with replacement) from. Only applies if data.type is "Normal". The vector contains realizations from the sampling prior (e.g. inverse-gamma distribution) for σ_t^2.
samp.prior.var.c	Vector of possible values of σ_c^2 to sample (with replacement) from. Only applies if data.type is "Normal". The vector contains realizations from the sampling prior (e.g. inverse-gamma distribution) for σ_c^2
prior.mu.t.shape1	First hyperparameter of the initial prior for μ_t. The default is 1. Does not apply if data.type is "Normal".

`prior.mu.t.shape2`
: Second hyperparameter of the initial prior for μ_t. The default is 1. Does not apply if `data.type` is "Normal".

`prior.mu.c.shape1`
: First hyperparameter of the initial prior for μ_c. The default is 1. Does not apply if `data.type` is "Normal".

`prior.mu.c.shape2`
: Second hyperparameter of the initial prior for μ_c. The default is 1. Does not apply if `data.type` is "Normal".

`delta`
: Prespecified constant that defines the boundary of the null hypothesis. The default is zero.

`gamma`
: Posterior probability threshold for rejecting the null. The null hypothesis is rejected if posterior probability is greater `gamma`. The default is 0.95.

`nMC`
: Number of iterations (excluding burn-in samples) for the slice sampler or Gibbs sampler. The default is 10,000.

`nBI`
: Number of burn-in samples for the slice sampler or Gibbs sampler. The default is 250.

`N`
: Number of simulated datasets to generate. The default is 10,000.

Details

If `data.type` is "Bernoulli", "Poisson" or "Exponential", a single response from the treatment group is assumed to follow Bern(μ_t), Pois(μ_t) or Exp(rate=μ_t), respectively, where μ_t is the mean of responses for the treatment group. If `data.type` is "Normal", a single response from the treatment group is assumed to follow $N(\mu_t, \tau^{-1})$ where τ is the precision parameter. The distributional assumptions for the control group data are analogous.

`samp.prior.mu.t` and `samp.prior.mu.c` can be generated using the sampling priors (see example).

If `data.type` is "Bernoulli", the initial prior for μ_t is beta(`prior.mu.t.shape1`, `prior.mu.t.shape2`).
If `data.type` is "Poisson", the initial prior for μ_t is Gamma(`prior.mu.t.shape1`, rate=`prior.mu.t.shape2`).
If `data.type` is "Exponential", the initial prior for μ_t is Gamma(`prior.mu.t.shape1`, rate=`prior.mu.t.shape2`).
The initial priors used for the control group data are analogous.

If `data.type` is "Normal", each historical dataset D_{0k} is assumed to have a different precision parameter τ_k. The initial prior for τ is the Jeffery's prior, τ^{-1}, and the initial prior for τ_k is τ_k^{-1}. The initial prior for the μ_c is the uniform improper prior.

If a sampling prior with support in the null space is used, the value returned is a Bayesian type I error rate. If a sampling prior with support in the alternative space is used, the value returned is a Bayesian power.

If `data.type` is "Normal", Gibbs sampling is used for model fitting. For all other data types, numerical integration is used for modeling fitting.

Value

The function returns a S3 object with a `summary` method. Power or type I error is returned, depending on the sampling prior used. The posterior probabilities of the alternative hypothesis are returned. Average posterior means of μ_t and μ_c and their corresponding biases are returned. If

data.type is "Normal", average posterior means of τ and τ_k's (if historical data is given) are also returned.

References

Yixuan Qiu, Sreekumar Balan, Matt Beall, Mark Sauder, Naoaki Okazaki and Thomas Hahn (2019). RcppNumerical: 'Rcpp' Integration for Numerical Computing Libraries. R package version 0.4-0. https://CRAN.R-project.org/package=RcppNumerical

Chen, Ming-Hui, et al. "Bayesian design of noninferiority trials for medical devices using historical data." Biometrics 67.3 (2011): 1163-1170.

See Also

two.grp.fixed.a0

Examples

```
data.type <- "Bernoulli"
n.t <- 100
n.c <- 100

# Simulate three historical datasets
historical <- matrix(0, ncol=3, nrow=3)
historical[1,] <- c(70, 100, 0.3)
historical[2,] <- c(60, 100, 0.5)
historical[3,] <- c(50, 100, 0.7)

# Generate sampling priors
set.seed(1)
b_st1 <- b_st2 <- 1
b_sc1 <- b_sc2 <- 1
samp.prior.mu.t <- rbeta(50000, b_st1, b_st2)
samp.prior.mu.c <- rbeta(50000, b_st1, b_st2)
# The null hypothesis here is H0: mu_t - mu_c >= 0. To calculate power,
# we can provide samples of mu.t and mu.c such that the mass of mu_t - mu_c < 0.
# To calculate type I error, we can provide samples of mu.t and mu.c such that
# the mass of mu_t - mu_c >= 0.
sub_ind <- which(samp.prior.mu.t < samp.prior.mu.c)
# Here, mass is put on the alternative region, so power is calculated.
samp.prior.mu.t <- samp.prior.mu.t[sub_ind]
samp.prior.mu.c <- samp.prior.mu.c[sub_ind]

N <- 1000 # N should be larger in practice
result <- power.two.grp.fixed.a0(data.type=data.type, n.t=n.t, n.c=n.t, historical=historical,
                        samp.prior.mu.t=samp.prior.mu.t, samp.prior.mu.c=samp.prior.mu.c,
                                  delta=0, N=N)
summary(result)
```

power.two.grp.random.a0 *Power/type I error calculation for two groups (treatment and control group, no covariates) with random a0*

Description

Power/type I error calculation using normalized power priors for two groups (treatment and control group, no covariates) with random a_0

Usage

```
power.two.grp.random.a0(
  data.type,
  n.t,
  n.c,
  historical,
  nullspace.ineq = ">",
  samp.prior.mu.t,
  samp.prior.mu.c,
  samp.prior.var.t = 0,
  samp.prior.var.c = 0,
  prior.mu.t.shape1 = 1,
  prior.mu.t.shape2 = 1,
  prior.mu.c.shape1 = 1,
  prior.mu.c.shape2 = 1,
  prior.a0.shape1 = rep(1, 10),
  prior.a0.shape2 = rep(1, 10),
  lower.limits = rep(0, 10),
  upper.limits = rep(1, 10),
  slice.widths = rep(0.1, 10),
  delta = 0,
  gamma = 0.95,
  nMC = 10000,
  nBI = 250,
  N = 10000
)
```

Arguments

data.type	Character string specifying the type of response. The options are "Normal", "Bernoulli", "Poisson" and "Exponential".
n.t	Sample size of the treatment group for the simulated datasets.
n.c	Sample size of the control group for the simulated datasets.
historical	Matrix of historical dataset(s). If data.type is "Normal", historical is a matrix with three columns:

- The first column contains the sum of responses for the control group.
- The second column contains the sample size of the control group.
- The third column contains the sample variance of responses for the control group.

For all other data types, `historical` is a matrix with two columns:

- The first column contains the sum of responses for the control group.
- The second column contains the sample size of the control group.

Each row represents a historical dataset.

`nullspace.ineq` Character string specifying the inequality of the null hypothesis. The options are ">" and "<". If ">" is specified, the null hypothesis (for non-exponential data) is H_0: μ_t - $\mu_c \geq \delta$. If "<" is specified, the null hypothesis is H_0: μ_t - $\mu_c \leq \delta$. The default choice is ">".

`samp.prior.mu.t`
Vector of possible values of μ_t to sample (with replacement) from. The vector contains realizations from the sampling prior (e.g. normal distribution) for μ_t.

`samp.prior.mu.c`
Vector of possible values of μ_c to sample (with replacement) from. The vector contains realizations from the sampling prior (e.g. normal distribution) for μ_c.

`samp.prior.var.t`
Vector of possible values of σ_t^2 to sample (with replacement) from. Only applies if `data.type` is "Normal". The vector contains realizations from the sampling prior (e.g. inverse-gamma distribution) for σ_t^2.

`samp.prior.var.c`
Vector of possible values of σ_c^2 to sample (with replacement) from. Only applies if `data.type` is "Normal". The vector contains realizations from the sampling prior (e.g. inverse-gamma distribution) for σ_c^2

`prior.mu.t.shape1`
First hyperparameter of the initial prior for μ_t. The default is 1. Does not apply if `data.type` is "Normal".

`prior.mu.t.shape2`
Second hyperparameter of the initial prior for μ_t. The default is 1. Does not apply if `data.type` is "Normal".

`prior.mu.c.shape1`
First hyperparameter of the initial prior for μ_c. The default is 1. Does not apply if `data.type` is "Normal".

`prior.mu.c.shape2`
Second hyperparameter of the initial prior for μ_c. The default is 1. Does not apply if `data.type` is "Normal".

`prior.a0.shape1`
Vector of the first shape parameters of the independent beta priors for a_0. The length of the vector should be equal to the number of historical datasets. The default is a vector of one's.

`prior.a0.shape2`
Vector of the second shape parameters of the independent beta priors for a_0. The length of the vector should be equal to the number of historical datasets. The default is a vector of one's.

lower.limits	Vector of lower limits for parameters to be used by the slice sampler. The length of the vector should be equal to the number of historical datasets. The default is 0 for all parameters (may not be appropriate for all situations).
upper.limits	Vector of upper limits for parameters to be used by the slice sampler. The length of the vector should be equal to the number of historical datasets. The default is 1 for all parameters (may not be appropriate for all situations).
slice.widths	Vector of initial slice widths used by the slice sampler. The length of the vector should be equal to the number of historical datasets. The default is 0.1 for all parameter (may not be appropriate for all situations).
delta	Prespecified constant that defines the boundary of the null hypothesis. The default is zero.
gamma	Posterior probability threshold for rejecting the null. The null hypothesis is rejected if posterior probability is greater gamma. The default is 0.95.
nMC	Number of iterations (excluding burn-in samples) for the slice sampler or Gibbs sampler. The default is 10,000.
nBI	Number of burn-in samples for the slice sampler or Gibbs sampler. The default is 250.
N	Number of simulated datasets to generate. The default is 10,000.

Details

If data.type is "Bernoulli", "Poisson" or "Exponential", a single response from the treatment group is assumed to follow Bern(μ_t), Pois(μ_t) or Exp(rate=μ_t), respectively, where μ_t is the mean of responses for the treatment group. If data.type is "Normal", a single response from the treatment group is assumed to follow $N(\mu_t, \tau^{-1})$ where τ is the precision parameter. The distributional assumptions for the control group data are analogous.

samp.prior.mu.t and samp.prior.mu.c can be generated using the sampling priors (see example).

If data.type is "Bernoulli", the initial prior for μ_t is beta(prior.mu.t.shape1, prior.mu.t.shape2).
If data.type is "Poisson", the initial prior for μ_t is Gamma(prior.mu.t.shape1, rate=prior.mu.t.shape2).
If data.type is "Exponential", the initial prior for μ_t is Gamma(prior.mu.t.shape1, rate=prior.mu.t.shape2).
The initial priors used for the control group data are analogous.

If data.type is "Normal", historical datasets are assumed to have the same precision parameter as the current dataset for computational simplicity. The initial prior for τ is the Jeffery's prior, τ^{-1}. The initial prior for the μ_c is the uniform improper prior. Posterior samples of μ_c and τ are obtained through Gibbs sampling.

Independent beta(prior.a0.shape1,prior.a0.shape1) priors are used for a_0. Posterior samples of a_0 are obtained through slice sampling. The default lower limits for the parameters are 0. The default upper limits for the parameters are 1. The default slice widths for the parameters are 0.1. The defaults may not be appropriate for all situations, and the user can specify the appropriate limits and slice width for each parameter.

If a sampling prior with support in the null space is used, the value returned is a Bayesian type I error rate. If a sampling prior with support in the alternative space is used, the value returned is a Bayesian power.

Value

The function returns a S3 object with a `summary` method. Power or type I error is returned, depending on the sampling prior used. The posterior probabilities of the alternative hypothesis are returned. Average posterior means of μ_t and μ_c and their corresponding biases are returned. The average posterior mean of a_0 is returned. If `data.type` is "Normal", the average posterior mean of τ is also returned.

References

Chen, Ming-Hui, et al. "Bayesian design of noninferiority trials for medical devices using historical data." Biometrics 67.3 (2011): 1163-1170.

Neal, Radford M. Slice sampling. Ann. Statist. 31 (2003), no. 3, 705–767.

See Also

`two.grp.random.a0`

Examples

```
data.type <- "Bernoulli"
n.t <- 100
n.c <- 100

# Simulate three historical datasets
historical <- matrix(0, ncol=2, nrow=3)
historical[1,] <- c(70, 100)
historical[2,] <- c(60, 100)
historical[3,] <- c(50, 100)

# Generate sampling priors
set.seed(1)
b_st1 <- b_st2 <- 1
b_sc1 <- b_sc2 <- 1
samp.prior.mu.t <- rbeta(50000, b_st1, b_st2)
samp.prior.mu.c <- rbeta(50000, b_st1, b_st2)
# The null hypothesis here is H0: mu_t - mu_c >= 0. To calculate power,
# we can provide samples of mu.t and mu.c such that the mass of mu_t - mu_c < 0.
# To calculate type I error, we can provide samples of mu.t and mu.c such that
# the mass of mu_t - mu_c >= 0.
sub_ind <- which(samp.prior.mu.t < samp.prior.mu.c)
# Here, mass is put on the alternative region, so power is calculated.
samp.prior.mu.t <- samp.prior.mu.t[sub_ind]
samp.prior.mu.c <- samp.prior.mu.c[sub_ind]

N <- 10 # N should be larger in practice
result <- power.two.grp.random.a0(data.type=data.type, n.t=n.t, n.c=n.c, historical=historical,
                        samp.prior.mu.t=samp.prior.mu.t, samp.prior.mu.c=samp.prior.mu.c,
                                  delta=0, nMC=10000, nBI=250, N=N)
summary(result)
```

`two.grp.fixed.a0` *Model fitting for two groups (treatment and control group, no covariates) with fixed a0*

Description

Model fitting using power priors for two groups (treatment and control group, no covariates) with fixed a_0

Usage

```
two.grp.fixed.a0(
  data.type,
  y.c,
  n.c,
  v.c,
  historical = matrix(0, 1, 4),
  prior.mu.c.shape1 = 1,
  prior.mu.c.shape2 = 1,
  nMC = 10000,
  nBI = 250
)
```

Arguments

`data.type`	Character string specifying the type of response. The options are "Normal", "Bernoulli", "Poisson" and "Exponential".
`y.c`	Sum of responses for the control group.
`n.c`	Sample size of the control group.
`v.c`	(For normal data only) sample variance of responses for the control group.
`historical`	(Optional) matrix of historical dataset(s). If `data.type` is "Normal", `historical` is a matrix with four columns: • The first column contains the sum of responses for the control group. • The second column contains the sample size of the control group. • The third column contains the sample variance of responses for the control group. • The fourth column contains the discounting parameter value a_0 (between 0 and 1). For all other data types, `historical` is a matrix with three columns: • The first column contains the sum of responses for the control group. • The second column contains the sample size of the control group. • The third column contains the discounting parameter value a_0 (between 0 and 1). Each row represents a historical dataset.

`prior.mu.c.shape1`
: First hyperparameter of the initial prior for μ_c. The default is 1. Does not apply if `data.type` is "Normal".

`prior.mu.c.shape2`
: Second hyperparameter of the initial prior for μ_c. The default is 1. Does not apply if `data.type` is "Normal".

`nMC`
: (For normal data only) number of iterations (excluding burn-in samples) for the Gibbs sampler. The default is 10,000.

`nBI`
: (For normal data only) number of burn-in samples for the Gibbs sampler. The default is 250.

Details

The power prior is applied on the data of the control group only. Therefore, only summaries of the responses of the control group need to be entered.

If `data.type` is "Bernoulli", "Poisson" or "Exponential", a single response from the treatment group is assumed to follow Bern(μ_t), Pois(μ_t) or Exp(rate=μ_t), respectively, where μ_t is the mean of responses for the treatment group. The distributional assumptions for the control group data are analogous.

If `data.type` is "Bernoulli", the initial prior for μ_t is beta(`prior.mu.t.shape1`, `prior.mu.t.shape2`).
If `data.type` is "Poisson", the initial prior for μ_t is Gamma(`prior.mu.t.shape1`, rate=`prior.mu.t.shape2`).
If `data.type` is "Exponential", the initial prior for μ_t is Gamma(`prior.mu.t.shape1`, rate=`prior.mu.t.shape2`).
The initial priors used for the control group data are analogous.

If `data.type` is "Normal", the responses are assumed to follow $N(\mu_c, \tau^{-1})$ where μ_c is the mean of responses for the control group and τ is the precision parameter. Each historical dataset D_{0k} is assumed to have a different precision parameter τ_k. The initial prior for τ is the Jeffery's prior, τ^{-1}, and the initial prior for τ_k is τ_k^{-1}. The initial prior for the μ_c is the uniform improper prior. Posterior samples are obtained through Gibbs sampling.

Value

The function returns a S3 object with a `summary` method. If `data.type` is "Normal", posterior samples of μ_c, τ and τ_k's (if historical data is given) are returned in the list item named `posterior.params`. For all other data types, two scalars, c_1 and c_2, are returned in the list item named `posterior.params`, representing the two parameters of the posterior distribution of μ_c. For Bernoulli responses, the posterior distribution of μ_c is beta(c_1, c_2). For Poisson responses, the posterior distribution of μ_c is Gamma(c_1, c_2) where c_2 is the rate parameter. For exponential responses, the posterior distribution of μ_c is Gamma(c_1, c_2) where c_2 is the rate parameter.

References

Chen, Ming-Hui, et al. "Bayesian design of noninferiority trials for medical devices using historical data." Biometrics 67.3 (2011): 1163-1170.

See Also

`power.two.grp.fixed.a0`

Examples

```
data.type <- "Bernoulli"
y.c <- 70
n.c <- 100

# Simulate three historical datasets
historical <- matrix(0, ncol=3, nrow=3)
historical[1,] <- c(70, 100, 0.3)
historical[2,] <- c(60, 100, 0.5)
historical[3,] <- c(50, 100, 0.7)

set.seed(1)
result <- two.grp.fixed.a0(data.type=data.type, y.c=y.c, n.c=n.c, historical=historical)
summary(result)
```

`two.grp.random.a0` *Model fitting for two groups (treatment and control group, no covariates) with random a0*

Description

Model fitting using normalized power priors for two groups (treatment and control group, no covariates) with random a_0

Usage

```
two.grp.random.a0(
  data.type,
  y.c,
  n.c,
  v.c,
  historical,
  prior.mu.c.shape1 = 1,
  prior.mu.c.shape2 = 1,
  prior.a0.shape1 = rep(1, 10),
  prior.a0.shape2 = rep(1, 10),
  lower.limits = rep(0, 10),
  upper.limits = rep(1, 10),
  slice.widths = rep(0.1, 10),
  nMC = 10000,
  nBI = 250
)
```

Arguments

`data.type`	Character string specifying the type of response. The options are "Normal", "Bernoulli", "Poisson" and "Exponential".

`y.c` Sum of responses for the control group.

`n.c` Sample size of the control group.

`v.c` (For normal data only) sample variance of responses for the control group.

`historical` Matrix of historical dataset(s). If `data.type` is "Normal", `historical` is a matrix with three columns:

- The first column contains the sum of responses for the control group.
- The second column contains the sample size of the control group.
- The third column contains the sample variance of responses for the control group.

For all other data types, `historical` is a matrix with two columns:

- The first column contains the sum of responses for the control group.
- The second column contains the sample size of the control group.

Each row represents a historical dataset.

`prior.mu.c.shape1`
First hyperparameter of the initial prior for μ_c. The default is 1. Does not apply if `data.type` is "Normal".

`prior.mu.c.shape2`
Second hyperparameter of the initial prior for μ_c. The default is 1. Does not apply if `data.type` is "Normal".

`prior.a0.shape1`
Vector of the first shape parameters of the independent beta priors for a_0. The length of the vector should be equal to the number of historical datasets. The default is a vector of one's.

`prior.a0.shape2`
Vector of the second shape parameters of the independent beta priors for a_0. The length of the vector should be equal to the number of historical datasets. The default is a vector of one's.

`lower.limits` Vector of lower limits for parameters to be used by the slice sampler. The length of the vector should be equal to the number of historical datasets. The default is 0 for all parameters (may not be appropriate for all situations).

`upper.limits` Vector of upper limits for parameters to be used by the slice sampler. The length of the vector should be equal to the number of historical datasets. The default is 1 for all parameters (may not be appropriate for all situations).

`slice.widths` Vector of initial slice widths used by the slice sampler. The length of the vector should be equal to the number of historical datasets. The default is 0.1 for all parameter (may not be appropriate for all situations).

`nMC` Number of iterations (excluding burn-in samples) for the slice sampler or Gibbs sampler. The default is 10,000.

`nBI` Number of burn-in samples for the slice sampler or Gibbs sampler. The default is 250.

Details

If `data.type` is "Bernoulli", "Poisson" or "Exponential", a single response from the treatment group is assumed to follow Bern(μ_t), Pois(μ_t) or Exp(rate=μ_t), respectively, where μ_t is the mean of responses for the treatment group. If `data.type` is "Normal", a single response from the treatment group is assumed to follow $N(\mu_t, \tau^{-1})$ where τ is the precision parameter. The distributional assumptions for the control group data are analogous.

If `data.type` is "Bernoulli", the initial prior for μ_t is beta(`prior.mu.t.shape1`, `prior.mu.t.shape2`).
If `data.type` is "Poisson", the initial prior for μ_t is Gamma(`prior.mu.t.shape1`, rate=`prior.mu.t.shape2`).
If `data.type` is "Exponential", the initial prior for μ_t is Gamma(`prior.mu.t.shape1`, rate=`prior.mu.t.shape2`).
The initial priors used for the control group data are analogous.

If `data.type` is "Normal", historical datasets are assumed to have the same precision parameter τ as the current dataset for computational simplicity. The initial prior for τ is the Jeffery's prior, τ^{-1}. The initial prior for the μ_c is the uniform improper prior. Posterior samples of μ_c and τ are obtained through Gibbs sampling.

Independent beta(`prior.a0.shape1`,`prior.a0.shape1`) priors are used for a_0. Posterior samples of a_0 are obtained through slice sampling. The default lower limits for the parameters are 0. The default upper limits for the parameters are 1. The default slice widths for the parameters are 0.1. The defaults may not be appropriate for all situations, and the user can specify the appropriate limits and slice width for each parameter.

Value

The function returns a S3 object with a `summary` method. If `data.type` is "Normal", posterior samples of μ_c, τ and a_0 are returned. For all other data types, posterior samples of μ_c and a_0 are returned. If there are K historical datasets, then $a_0 = (a_{01}, \cdots, a_{0K})$.

References

Neal, Radford M. Slice sampling. Ann. Statist. 31 (2003), no. 3, 705–767.

See Also

`power.two.grp.random.a0`

Examples

```
data.type <- "Bernoulli"
y.c <- 70
n.c <- 100

# Simulate three historical datasets
historical <- matrix(0, ncol=2, nrow=3)
historical[1,] <- c(70, 100)
historical[2,] <- c(60, 100)
historical[3,] <- c(50, 100)

# Set parameters of the slice sampler
lower.limits <- rep(0, 3) # The dimension is the number of historical datasets
```

```
upper.limits <- rep(1, 3)
slice.widths <- rep(0.1, 3)

set.seed(1)
result <- two.grp.random.a0(data.type=data.type, y.c=y.c, n.c=n.c, historical=historical,
                            lower.limits=lower.limits, upper.limits=upper.limits,
                            slice.widths=slice.widths, nMC=10000, nBI=250)
summary(result)
```

Index