Package ‘hdbayes’

April 8, 2024

Title Bayesian Analysis of Generalized Linear Models with Historical Data

Version 0.0.3

Description User-friendly functions for leveraging (multiple) historical data set(s) for generalized linear models. Contains functions for sampling from the posterior distribution of a generalized linear model using the prior induced by the Bayesian hierarchical model, power prior by Ibrahim and Chen (2000) <doi:10.1214/ss/1009212673>, normalized power prior by Duan et al. (2006) <doi:10.1002/env.752>, normalized asymptotic power prior by Ibrahim et al. (2015) <doi:10.1002/sim.6728>, commensurate prior by Hobbs et al. (2011) <doi:10.1111/j.1541-0420.2011.01564.x>, robust meta-analytic-predictive prior by Schmidli et al. (2014) <doi:10.1111/biom.12242>, and the latent exchangeability prior (LEAP) by Alt et al. (2023) <arXiv:2303.05223>. The package compiles all the 'CmdStan' models once during installation using the 'instantiate' package.

License MIT + file LICENSE

URL `https://github.com/ethan-alt/hdbayes`

BugReports `https://github.com/ethan-alt/hdbayes/issues`

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 4.2.0)

Imports instantiate (>= 0.1.0), callr, fs, formula.tools, stats, posterior, enrichwith, mclust, bridgesampling

Suggests cmdstanr (>= 0.6.0), parallel

Additional_repositories `https://mc-stan.org/r-packages/`

SystemRequirements CmdStan (https://mc-stan.org/users/interfaces/cmdstan)

LazyData true

Collate 'E1694-data.R' 'E2696-data.R' 'IBCSG_curr-data.R' 'IBCSG_hist-data.R' 'actg019-data.R' 'actg036-data.R' 'data_checks.R' 'expfam_loglik.R' 'glm_bhm.R' 'glm_commensurate.R' 'glm_leap.R' 'glm_napp.R'

'glm_npp_lognc.R' 'glm_npp.R' 'glm_pp.R' 'glm_rmap.R'
'glm_rmap_bhm.R' 'glm_rmap_bhm_approx.R' 'hdbayes-package.R'
'lm_npp.R' 'zzz.R'

NeedsCompilation yes

Author Ethan M. Alt [aut, cre, cph] (<`https://orcid.org/0000-0002-6112-9030`>),
Xinxin Chen [aut],
Luiz M. Carvalho [aut],
Joseph G. Ibrahim [aut],
Xiuya Chang [ctb]

Maintainer Ethan M. Alt <`ethanalt@live.unc.edu`>

Repository CRAN

Date/Publication 2024-04-08 14:50:08 UTC

R topics documented:

`actg019` *AIDS Clinical Trial ACTG019*

Description

A data set from the AIDS clinical trial ACTG019 (`https://clinicaltrials.gov/ct2/show/NCT00000736`) comparing zidovudine (AZT) with a placebo in adults with asymptomatic HIV infection. The study results were described in Volberding et al. (1990) `doi:10.1056/NEJM199004053221401`.

Usage

```
actg019
```

Format

A data frame with 822 rows and 5 variables:

outcome outcome variable with 1 indicating death, development of AIDS or AIDS-related complex (ARC) and 0 otherwise

age patient age in years

treatment treatment indicator, 0 = placebo, 1 = AZT

race race indicator, 0 = non-white, 1 = white

cd4 CD4 cell count

References

Volberding, P. A., Lagakos, S. W., Koch, M. A., Pettinelli, C., Myers, M. W., Booth, D. K., Balfour, H. H., Reichman, R. C., Bartlett, J. A., Hirsch, M. S., Murphy, R. L., Hardy, W. D., Soeiro, R., Fischl, M. A., Bartlett, J. G., Merigan, T. C., Hyslop, N. E., Richman, D. D., Valentine, F. T., Corey, L., and the AIDS Clinical Trials Group of the National Institute of Allergy and Infectious Diseases (1990). Zidovudine in asymptomatic human immunodeficiency virus infection. New England Journal of Medicine, 322(14), 941–949.

Chen, M.-H., Ibrahim, J. G., and Yiannoutsos, C. (1999). Prior elicitation, Variable Selection and Bayesian computation for Logistic Regression Models. Journal of the Royal Statistical Society Series B: Statistical Methodology, 61(1), 223–242.

`actg036` *AIDS Clinical Trial ACTG036*

Description

A data set from the AIDS clinical trial ACTG036 (`https://clinicaltrials.gov/study/NCT00001104`) comparing zidovudine (AZT) with a placebo in patients with hereditary coagulation disorders and HIV infection. The study results were described in Merigan et al. (1991) `doi:10.1182/blood.V78.4.900.900`. This data set has the same variables as the actg019 data set. We can use the actg019 data as the historical data and the actg036 data as the current data.

Usage

```
actg036
```

Format

A data frame with 183 rows and 5 variables:

outcome outcome variable with 1 indicating death, development of AIDS or AIDS-related complex (ARC) and 0 otherwise

age patient age in years

treatment treatment indicator, 0 = placebo, 1 = AZT

race race indicator, 0 = non-white, 1 = white

cd4 CD4 cell count

References

Merigan, T., Amato, D., Balsley, J., Power, M., Price, W., Benoit, S., Perez-Michael, A., Brownstein, A., Kramer, A., and Brettler, D. (1991). Placebo-controlled trial to evaluate zidovudine in treatment of human immunodeficiency virus infection in asymptomatic patients with hemophilia. NHF-ACTG 036 Study Group. Blood, 78(4), 900–906.

Chen, M.-H., Ibrahim, J. G., and Yiannoutsos, C. (1999). Prior elicitation, Variable Selection and Bayesian computation for Logistic Regression Models. Journal of the Royal Statistical Society Series B: Statistical Methodology, 61(1), 223–242.

`E1694` *ECOG E1694 Trial*

Description

A data set from the ECOG E1694 trial comparing the GM2-KLH/QS-21 (GMK) vaccine with high-dose interferon alfa-2b (IFN) therapy in resected high-risk melanoma patients. The study results were described in Kirkwood et al. (2001) `doi:10.1200/JCO.2001.19.9.2370`. This data set only includes patients without nodal metastasis and has the same variables as the E2696 data set. We can use the E2696 data as the historical data and the E1694 data as the current data.

Usage

```
E1694
```

Format

A data frame with 200 rows and 6 variables:

failtime relapse-free survival (RFS) times (in months)

failind relapse indicator, 0 = right censored, 1 = relapse

treatment treatment indicator, 0 = GMK, 1 = IFN

sex gender indicator, 0 = male, 1 = female

age patient age in years

perform ECOG performance status indicator, 0 = fully active patient, able to carry on all pre-disease performance without restriction, 1 = restricted in physically strenuous activity, but are ambulatory and able to carry out work of a light or sedentary nature

References

Kirkwood, J. M., Ibrahim, J. G., Sosman, J. A., Sondak, V. K., Agarwala, S. S., Ernstoff, M. S., and Rao, U. (2001). High-dose interferon alfa-2b significantly prolongs relapse-free and overall survival compared with the GM2-KLH/QS-21 vaccine in patients with resected stage IIB-III melanoma: Results of intergroup trial E1694/S9512/C509801. Journal of Clinical Oncology, 19(9), 2370–2380.

`E2696` *ECOG E2696 Trial*

Description

A data set from the ECOG E2696 trial comparing the combination of the GM2-KLH/QS-21 (GMK) vaccine and high-dose interferon alfa-2b (IFN) therapy with the GMK vaccine alone in resected high-risk melanoma patients. The study results were described in Kirkwood et al. (2001) `doi:10.1200/JCO.2001.19.5.1430`. This data set only includes patients without nodal metastasis.

Usage

```
E2696
```

Format

A data frame with 105 rows and 6 variables:

failtime relapse-free survival (RFS) times (in months)

failind relapse indicator, 0 = right censored, 1 = relapse

treatment treatment indicator, 0 = GMK, 1 = GMK and IFN

sex gender indicator, 0 = male, 1 = female

age patient age in years

perform ECOG performance status indicator, 0 = fully active patient, able to carry on all pre-disease performance without restriction, 1 = restricted in physically strenuous activity, but are ambulatory and able to carry out work of a light or sedentary nature

References

Kirkwood, J. M., Ibrahim, J., Lawson, D. H., Atkins, M. B., Agarwala, S. S., Collins, K., Mascari, R., Morrissey, D. M., and Chapman, P. B. (2001). High-dose interferon alfa-2b does not diminish antibody response to GM2 vaccination in patients with resected melanoma: Results of the multicenter eastern cooperative oncology group phase II trial E2696. Journal of Clinical Oncology, 19(5), 1430–1436.

`glm.bhm` *Posterior of Bayesian hierarchical model (BHM)*

Description

Sample from the posterior distribution of a GLM using the BHM.

Usage

```
glm.bhm(
  formula,
  family,
  data.list,
  offset.list = NULL,
  meta.mean.mean = NULL,
  meta.mean.sd = NULL,
  meta.sd.mean = NULL,
  meta.sd.sd = NULL,
  disp.mean = NULL,
  disp.sd = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.
`family`	an object of class `family`. See `?stats::family`.
`data.list`	a list of `data.frames`. The first element in the list is the current data, and the rest are the historical data sets.
`offset.list`	a list of vectors giving the offsets for each data. The length of offset.list is equal to the length of data.list. The length of each element of offset.list is equal to the number of rows in the corresponding element of data.list. Defaults to a list of vectors of 0s.
`meta.mean.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the means for the normal hyperpriors on the mean hyperparameters of regression coefficients. If a scalar is provided, meta.mean.mean will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`meta.mean.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sds for the normal hyperpriors on the mean hyperparameters of regression coefficients. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 10s.

`meta.sd.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the means for the half-normal hyperpriors on the sd hyperparameters of regression coefficients. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 0s.
`meta.sd.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sds for the half-normal hyperpriors on the sd hyperparameters of regression coefficients. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 1s.
`disp.mean`	a scalar or a vector whose dimension is equal to the number of data sets (including the current data) giving the means for the half-normal priors on the dispersion parameters. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 0s.
`disp.sd`	a scalar or a vector whose dimension is equal to the number of data sets (including the current data) giving the sds for the half-normal priors on the dispersion parameters. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 10s.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Details

The BHM assumes that the regression coefficients for the historical and current data are different, but are correlated through a common distribution, whose hyperparameters (i.e., mean and standard deviation (sd) (the covariance matrix is assumed to have a diagonal structure)) are treated as random. The number of of regression coefficients for the current data is assumed to be the same as that for the historical data.

The hyperpriors on the mean and the sd hyperparameters are independent normal and independent half-normal distributions, respectively. The priors on the dispersion parameters (if applicable) for the current and historical data sets are independent half-normal distributions.

Value

The function returns an object of class `draws_df` giving posterior samples.

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg019)
  data(actg036)
  ## take subset for speed purposes
  actg019 = actg019[1:100, ]
```

```
  actg036 = actg036[1:50, ]
  data_list = list(currdata = actg019, histdata = actg036)
  glm.bhm(
    formula = outcome ~ scale(age) + race + treatment + scale(cd4),
    family = binomial('logit'),
    data.list = data_list,
    chains = 1, iter_warmup = 500, iter_sampling = 1000
  )
}
```

`glm.commensurate` *Posterior of commensurate prior (CP)*

Description

Sample from the posterior distribution of a GLM using the CP by Hobbs et al. (2011) `doi:10.1111/j.1541-0420.2011.01564.x`.

Usage

```
glm.commensurate(
  formula,
  family,
  data.list,
  tau,
  offset.list = NULL,
  beta0.mean = NULL,
  beta0.sd = NULL,
  disp.mean = NULL,
  disp.sd = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates
`family`	an object of class `family`. See `?stats::family`
`data.list`	a list of `data.frames`. The first element in the list is the current data, and the rest are the historical data sets.
`tau`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the commensurate prior parameters. If a scalar is provided, tau will be a vector of repeated elements of the given scalar. Each element of tau must be positive, corresponding to a normal precision parameter.

`offset.list`	a list of vectors giving the offsets for each data. The length of offset.list is equal to the length of data.list. The length of each element of offset.list is equal to the number of rows in the corresponding element of data.list. Defaults to a list of vectors of 0s.
`beta0.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the mean parameters for the prior on the historical data regression coefficients. If a scalar is provided, same as for tau. Defaults to a vector of 0s.
`beta0.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sd parameters for the prior on the historical data regression coefficients. If a scalar is provided, same as for tau. Defaults to a vector of 10s.
`disp.mean`	a scalar or a vector whose dimension is equal to the number of data sets (including the current data) giving the means for the half-normal priors on the dispersion parameters. If a scalar is provided, same as for tau. Defaults to a vector of 0s.
`disp.sd`	a scalar or a vector whose dimension is equal to the number of data sets (including the current data) giving the sds for the half-normal priors on the dispersion parameters. If a scalar is provided, same as for tau. Defaults to a vector of 10s.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Details

The CP assumes that the regression coefficients for the current data conditional on those for the historical data are independent normal distributions with mean equal to the corresponding regression coefficients for the historical data and variance equal to the inverse of the corresponding elements of a user-specified vector (tau) of precision parameters. The number of current data regression coefficients is assumed to be the same as that of historical data regression coefficients. The priors on the dispersion parameters (if applicable) for the current and historical data sets are independent half-normal distributions.

Value

The function returns an object of class `draws_df` giving posterior samples.

References

Hobbs, B. P., Carlin, B. P., Mandrekar, S. J., and Sargent, D. J. (2011). Hierarchical commensurate and power prior models for adaptive incorporation of historical information in clinical trials. Biometrics, 67(3), 1047–1056.

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg019)
  data(actg036)
  ## take subset for speed purposes
  actg019 = actg019[1:100, ]
  actg036 = actg036[1:50, ]
  data_list = list(currdata = actg019, histdata = actg036)
  glm.commensurate(
    formula = cd4 ~ treatment + age + race,
    family = poisson(), data.list = data_list,
    tau = rep(5, 4),    ## 4 parameters including intercept
    chains = 1, iter_warmup = 500, iter_sampling = 1000
  )
}
```

`glm.leap` *Posterior of Latent Exchangeability Prior (LEAP)*

Description

Sample from the posterior distribution of a GLM using the LEAP by Alt et al. (2023).

Usage

```
glm.leap(
  formula,
  family,
  data.list,
  K = 2,
  prob.conc = NULL,
  offset.list = NULL,
  beta.mean = NULL,
  beta.sd = NULL,
  disp.mean = NULL,
  disp.sd = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.
`family`	an object of class `family`. See `?stats::family`.

`data.list`	a list of two `data.frames` giving the current data followed by one historical data set.
`K`	the desired number of classes to identify. Defaults to 2.
`prob.conc`	a scalar or a vector of length `K` giving the concentration parameters for Dirichlet prior. If length == 2, a `beta(prob.conc[1], prob.conc[2])` prior is used. If a scalar is provided, prob.conc will be a vector of repeated elements of the given scalar. Defaults to a vector of 1s.
`offset.list`	a list of matrices giving the offset for current data followed by historical data. For each matrix, the number of rows corresponds to observations and columns correspond to classes. Defaults to a list of matrices of 0s.
`beta.mean`	a `p x K` matrix of mean parameters for initial prior on regression coefficients (including intercept). Defaults to a matrix of 0s.
`beta.sd`	a `p x K` matrix of sd parameters for the initial prior on regression coefficients (including intercept). Defaults to a matrix of 10s.
`disp.mean`	a scalar or a vector whose dimension is equal to the number of classes (`K`) giving the means for the half-normal hyperpriors on the dispersion parameters. If a scalar is provided, disp.mean will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`disp.sd`	a scalar or a vector whose dimension is equal to the number of classes (`K`) giving the sds for the half-normal hyperpriors on the dispersion parameters. If a scalar is provided, same as for disp.mean. Defaults to a vector of 10s.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Details

The LEAP discounts the historical data by identifying the most relevant individuals from the historical data. It is equivalent to a prior induced by the posterior of a finite mixture model for the historical data set.

Value

The function returns an object of class `draws_df` giving posterior samples.

References

Alt, E. M., Chang, X., Jiang, X., Liu, Q., Mo, M., Xia, H. M., and Ibrahim, J. G. (2023). LEAP: The latent exchangeability prior for borrowing information from historical data. arXiv preprint.

Examples

```
data(actg019)
data(actg036)
# take subset for speed purposes
actg019 = actg019[1:100, ]
actg036 = actg036[1:50, ]
if (instantiate::stan_cmdstan_exists()) {
  glm.leap(
    formula = outcome ~ scale(age) + race + treatment + scale(cd4),
    family = binomial('logit'),
    data.list = list(actg019, actg036),
    K = 2,
    chains = 1, iter_warmup = 500, iter_sampling = 1000
  )
}
```

`glm.napp` *Posterior of normalized asymptotic power prior (NAPP)*

Description

Sample from the posterior distribution of a GLM using the NAPP by Ibrahim et al. (2015) doi:10.1002/sim.6728.

Usage

```
glm.napp(
  formula,
  family,
  data.list,
  offset.list = NULL,
  a0.shape1 = 1,
  a0.shape2 = 1,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.
`family`	an object of class `family`. See `?stats::family`.
`data.list`	a list of `data.frames`. The first element in the list is the current data, and the rest are the historical datasets.

`offset.list`	a list of vectors giving the offsets for each data. The length of offset.list is equal to the length of data.list. The length of each element of offset.list is equal to the number of rows in the corresponding element of data.list. Defaults to a list of vectors of 0s.
`a0.shape1`	first shape parameter for the i.i.d. beta prior on a0 vector. When `a0.shape1 == 1` and `a0.shape2 == 1`, a uniform prior is used.
`a0.shape2`	second shape parameter for the i.i.d. beta prior on a0 vector. When `a0.shape1 == 1` and `a0.shape2 == 1`, a uniform prior is used.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Details

The NAPP assumes that the regression coefficients and logarithm of the dispersion parameter are a multivariate normal distribution with mean equal to the maximum likelihood estimate of the historical data and covariance matrix equal to a_0^{-1} multiplied by the inverse Fisher information matrix of the historical data, where a_0 is the power prior parameter (treated as random).

Value

The function returns an object of class `draws_df` giving posterior samples.

References

Ibrahim, J. G., Chen, M., Gwon, Y., and Chen, F. (2015). The power prior: Theory and applications. Statistics in Medicine, 34(28), 3724–3749.

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg019)
  data(actg036)
  ## take subset for speed purposes
  actg019 = actg019[1:100, ]
  actg036 = actg036[1:50, ]
  data_list = list(currdata = actg019, histdata = actg036)
  glm.napp(
    formula = cd4 ~ treatment + age + race,
    family = poisson('log'),
    data.list = data_list,
    chains = 1, iter_warmup = 500, iter_sampling = 1000
  )
}
```

`glm.npp` *Posterior of normalized power prior (NPP)*

Description

Sample from the posterior distribution of a GLM using the NPP by Duan et al. (2006) doi:10.1002/env.752.

Usage

```
glm.npp(
  formula,
  family,
  data.list,
  a0.lognc,
  lognc,
  offset.list = NULL,
  beta.mean = NULL,
  beta.sd = NULL,
  disp.mean = NULL,
  disp.sd = NULL,
  a0.shape1 = 1,
  a0.shape2 = 1,
  a0.lower = NULL,
  a0.upper = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.
`family`	an object of class `family`. See `?stats::family`.
`data.list`	a list of `data.frames`. The first element in the list is the current data, and the rest are the historical data sets.
`a0.lognc`	a vector giving values of the power prior parameter for which the logarithm of the normalizing constant has been evaluated.
`lognc`	an S by T matrix where S is the length of a0.lognc, T is the number of historical data sets, and the j-th column, j = 1, ..., T, is a vector giving the logarithm of the normalizing constant (as estimated by `glm.npp.lognc()` for a0.lognc using the j-th historical data set.

`offset.list`	a list of vectors giving the offsets for each data. The length of offset.list is equal to the length of data.list. The length of each element of offset.list is equal to the number of rows in the corresponding element of data.list. Defaults to a list of vectors of 0s.
`beta.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the mean parameters for the initial prior on regression coefficients. If a scalar is provided, beta.mean will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`beta.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sd parameters for the initial prior on regression coefficients. If a scalar is provided, same as for beta.mean. Defaults to a vector of 10s.
`disp.mean`	mean parameter for the half-normal prior on dispersion parameter. Defaults to 0.
`disp.sd`	sd parameter for the half-normal prior on dispersion parameter. Defaults to 10.
`a0.shape1`	first shape parameter for the i.i.d. beta prior on a0 vector. When `a0.shape1 == 1` and `a0.shape2 == 1`, a uniform prior is used.
`a0.shape2`	second shape parameter for the i.i.d. beta prior on a0 vector. When `a0.shape1 == 1` and `a0.shape2 == 1`, a uniform prior is used.
`a0.lower`	a scalar or a vector whose dimension is equal to the number of historical data sets giving the lower bounds for each element of the a0 vector. If a scalar is provided, a0.lower will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`a0.upper`	a scalar or a vector whose dimension is equal to the number of historical data sets giving the upper bounds for each element of the a0 vector. If a scalar is provided, same as for a0.lower. Defaults to a vector of 1s.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Details

Before using this function, users must estimate the logarithm of the normalizing constant across a range of different values for the power prior parameter (a_0), possibly smoothing techniques over a find grid. The power prior parameters (a_0's) are treated as random with independent beta priors. The initial priors on the regression coefficients are independent normal priors. The current and historical data sets are assumed to have a common dispersion parameter with a half-normal prior (if applicable). For normal linear models, the exact normalizing constants for NPP can be computed. See the implementation in `lm.npp()`.

Value

The function returns an object of class `draws_df` giving posterior samples.

References

Duan, Y., Ye, K., and Smith, E. P. (2005). Evaluating water quality using power priors to incorporate historical information. Environmetrics, 17(1), 95–106.

See Also

`glm.npp.lognc()`

Examples

```
if(requireNamespace("parallel")){
  data(actg019)
  data(actg036)
  ## take subset for speed purposes
  actg019 = actg019[1:100, ]
  actg036 = actg036[1:50, ]

  library(parallel)
  ncores    = 2
  data.list = list(data = actg019, histdata = actg036)
  formula   = cd4 ~ treatment + age + race
  family    = poisson()
  a0        = seq(0, 1, length.out = 11)
  if (instantiate::stan_cmdstan_exists()) {
    ## call created function
    ## wrapper to obtain log normalizing constant in parallel package
    logncfun = function(a0, ...){
      hdbayes::glm.npp.lognc(
        formula = formula, family = family, a0 = a0, histdata = data.list[[2]],
        ...
      )
    }

    cl = makeCluster(ncores)
    clusterSetRNGStream(cl, 123)
    clusterExport(cl, varlist = c('formula', 'family', 'data.list'))
    a0.lognc = parLapply(
      cl = cl, X = a0, fun = logncfun, iter_warmup = 500,
      iter_sampling = 1000, chains = 1, refresh = 0
    )
    stopCluster(cl)
    a0.lognc = data.frame( do.call(rbind, a0.lognc) )

    ## sample from normalized power prior
    glm.npp(
      formula = cd4 ~ treatment + age + race,
      family = poisson(),
```

```
        data.list = data.list,
        a0.lognc = a0.lognc$a0,
        lognc = matrix(a0.lognc$lognc, ncol = 1),
        chains = 1, iter_warmup = 500, iter_sampling = 1000,
        refresh = 0
      )
    }
  }
```

`glm.npp.lognc` *Estimate the logarithm of the normalizing constant for normalized power prior (NPP) for one data set*

Description

Uses Markov chain Monte Carlo (MCMC) and bridge sampling to estimate the logarithm of the normalizing constant for the NPP for a fixed value of the power prior parameter $a_0 \in (0, 1)$ for one data set. The initial priors are independent normal priors on the regression coefficients and a half-normal prior on the dispersion parameter (if applicable).

Usage

```
glm.npp.lognc(
  formula,
  family,
  histdata,
  a0,
  offset0 = NULL,
  beta.mean = NULL,
  beta.sd = NULL,
  disp.mean = NULL,
  disp.sd = NULL,
  bridge.args = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.
`family`	an object of class `family`. See `?stats::family`.
`histdata`	a `data.frame` giving the historical data.
`a0`	the power prior parameter (a scalar between 0 and 1).

`offset0`	vector whose dimension is equal to the rows of the historical data set giving an offset for the historical data. Defaults to a vector of 0s.
`beta.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the mean parameters for the normal initial prior on regression coefficients given the dispersion parameter. If a scalar is provided, beta.mean will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`beta.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sd parameters for the initial prior on regression coefficients. The sd used is `sqrt(dispersion) * beta.sd`. If a scalar is provided, same as for beta.mean. Defaults to a vector of 10s.
`disp.mean`	mean parameter for the half-normal prior on dispersion parameter. Defaults to 0.
`disp.sd`	sd parameter for the half-normal prior on dispersion parameter. Defaults to 10.
`bridge.args`	a `list` giving arguments (other than samples, log_posterior, data, lb, ub) to pass onto `bridgesampling::bridge_sampler()`.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Value

The function returns a vector giving the value of a0, the estimated logarithm of the normalizing constant, the minimum estimated bulk effective sample size of the MCMC sampling, and the maximum Rhat.

References

Gronau, Q. F., Singmann, H., and Wagenmakers, E.-J. (2020). bridgesampling: An r package for estimating normalizing constants. Journal of Statistical Software, 92(10).

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg036)
  ## take subset for speed purposes
  actg036 = actg036[1:50, ]
  glm.npp.lognc(
    cd4 ~ treatment + age + race,
    family = poisson(), histdata = actg036, a0 = 0.5,
    chains = 1, iter_warmup = 500, iter_sampling = 5000
  )
}
```

`glm.pp` *Posterior of power prior (PP) with fixed a_0*

Description

Sample from the posterior distribution of a GLM using the PP by Ibrahim and Chen (2000) `doi:10.1214/ss/1009212673`.

Usage

```
glm.pp(
  formula,
  family,
  data.list,
  a0.vals,
  offset.list = NULL,
  beta.mean = NULL,
  beta.sd = NULL,
  disp.mean = NULL,
  disp.sd = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.
`family`	an object of class `family`. See `?stats::family`.
`data.list`	a list of `data.frames`. The first element in the list is the current data, and the rest are the historical data sets.
`a0.vals`	a scalar between 0 and 1 or a vector whose dimension is equal to the number of historical data sets giving the (fixed) power prior parameter for each historical data set. Each element of vector should be between 0 and 1. If a scalar is provided, same as for beta.mean.
`offset.list`	a list of vectors giving the offsets for each data. The length of offset.list is equal to the length of data.list. The length of each element of offset.list is equal to the number of rows in the corresponding element of data.list. Defaults to a list of vectors of 0s.
`beta.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the mean parameters for the initial prior on regression coefficients. If a scalar is provided, beta.mean will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.

`beta.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sd parameters for the initial prior on regression coefficients. If a scalar is provided, same as for beta.mean. Defaults to a vector of 10s.
`disp.mean`	mean parameter for the half-normal prior on dispersion parameter. Defaults to 0.
`disp.sd`	sd parameter for the half-normal prior on dispersion parameter. Defaults to 10.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Details

The power prior parameters (a_0's) are treated as fixed. The initial priors on the regression coefficients are independent normal priors. The current and historical data sets are assumed to have a common dispersion parameter with a half-normal prior (if applicable).

Value

The function returns an object of class `draws_df` giving posterior samples.

References

Chen, M.-H. and Ibrahim, J. G. (2000). Power prior distributions for Regression Models. Statistical Science, 15(1).

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg019)
  data(actg036)
  ## take subset for speed purposes
  actg019 = actg019[1:100, ]
  actg036 = actg036[1:50, ]
  data_list = list(currdata = actg019, histdata = actg036)
  glm.pp(
    formula = cd4 ~ treatment + age + race,
    family = poisson('log'),
    data.list = data_list,
    a0.vals = 0.5,
    chains = 1, iter_warmup = 500, iter_sampling = 1000
  )
}
```

---

glm.rmap *Posterior of robust meta-analytic predictive prior (RMAP)*

Description

Final step for sampling from the posterior distribution of a GLM using the RMAP by Schmidli et al. (2014) doi:10.1111/biom.12242.

Usage

```
glm.rmap(
  formula,
  family,
  curr.data,
  probs,
  means,
  covs,
  curr.offset = NULL,
  w = 0.1,
  norm.vague.mean = NULL,
  norm.vague.sd = NULL,
  curr.disp.mean = NULL,
  curr.disp.sd = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.
`family`	an object of class `family`. See `?stats::family`.
`curr.data`	a `data.frame` giving the current data.
`probs`	a vector of mixing proportions in the mixture approximation to the prior induced by the BHM. Obtained from the outputs of the `glm.rmap.bhm.approx()` function.
`means`	a matrix with the jth column being the mean vector for the jth component in the mixture approximation to the prior induced by the BHM. Obtained from the outputs of the `glm.rmap.bhm.approx()` function.
`covs`	a 3-dimensional array giving the covariance matrices for the mixture approximation to the prior induced by the BHM. Obtained from the outputs of the `glm.rmap.bhm.approx()` function. the means for the half-normal hyperpriors on the sd hyperparameters of regression coefficients.

`curr.offset`	a vector whose dimension is equal to the rows of the current data set giving an offset for the current data. Defaults to a vector of 0s.
`w`	a scalar between 0 and 1 giving how much weight to put on the historical data.
`norm.vague.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the means for the vague normal prior on regression coefficients. If a scalar is provided, norm.vague.mean will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`norm.vague.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sds for the vague normal prior on regression coefficients. If a scalar is provided, same as for norm.vague.mean. Defaults to a vector of 10s.
`curr.disp.mean`	a scalar giving the mean for the half-normal hyperprior on the dispersion parameter for the current data. Defaults to a vector of 0s.
`curr.disp.sd`	a scalar giving the sd for the half-normal hyperprior on the dispersion parameter for the current data. Defaults to a vector of 10s.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Details

This function samples from the posterior distribution of a GLM using the RMAP. The first component of the RMAP is a prior induced by the Bayesian hierarchical model (BHM). We approximate this component by a mixture of multivariate normal distributions where the parameters are obtained from the outputs of the `glm.rmap.bhm.approx()` function. The second component is a vague (non-informative) multivariate normal prior. We assume that the covariance matrix of the vague prior is a diagonal matrix.

Value

The function returns an object of class `draws_df` giving posterior samples.

References

Schmidli, H., Gsteiger, S., Roychoudhury, S., O'Hagan, A., Spiegelhalter, D., and Neuenschwander, B. (2014). Robust meta-analytic-predictive priors in clinical trials with historical control information. Biometrics, 70(4), 1023–1032.

See Also

`glm.rmap.bhm()` for the first step and `glm.rmap.bhm.approx()` for the second step of implementing RMAP.

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg019) ## current data
  data(actg036) ## historical data
  ## take subset for speed purposes
  actg019 = actg019[1:150, ]
  actg036 = actg036[1:100, ]
  hist_data_list = list(actg036)
  samples_bhm = glm.rmap.bhm(
    formula = outcome ~ scale(age) + race + treatment + scale(cd4),
    family = binomial('logit'),
    hist.data.list = hist_data_list,
    chains = 1, iter_warmup = 1000, iter_sampling = 2000
  )$beta_pred
  res_approx = glm.rmap.bhm.approx(
    samples.bhm = samples_bhm,
    G = 1:5, verbose = FALSE
  )
  glm.rmap(
    formula = outcome ~ scale(age) + race + treatment + scale(cd4),
    family = binomial('logit'),
    curr.data = actg019,
    probs = res_approx$probs,
    means = res_approx$means,
    covs = res_approx$covs,
    chains = 1, iter_warmup = 1000, iter_sampling = 2000
  )
}
```

`glm.rmap.bhm` *Posterior of robust meta-analytic predictive prior (RMAP)*

Description

First step for sampling from the posterior distribution of a GLM using the RMAP by Schmidli et al. (2014) doi:10.1111/biom.12242.

Usage

```
glm.rmap.bhm(
  formula,
  family,
  hist.data.list,
  hist.offset.list = NULL,
  meta.mean.mean = NULL,
  meta.mean.sd = NULL,
  meta.sd.mean = NULL,
```

```
  meta.sd.sd = NULL,
  hist.disp.mean = NULL,
  hist.disp.sd = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.
`family`	an object of class `family`. See `?stats::family`.
`hist.data.list`	a list of `data.frames`. Each element in the list is a historical data set.
`hist.offset.list`	a list of vectors giving the offsets for each historical data. The length of hist.offset.list is equal to the length of hist.data.list. The length of each element of hist.offset.list is equal to the number of rows in the corresponding element of hist.data.list. Defaults to a list of vectors of 0s.
`meta.mean.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the means for the normal hyperpriors on the mean hyperparameters of regression coefficients. If a scalar is provided, meta.mean.mean will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`meta.mean.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sds for the normal hyperpriors on the mean hyperparameters of regression coefficients. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 1s.
`meta.sd.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the means for the half-normal hyperpriors on the sd hyperparameters of regression coefficients. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 0s.
`meta.sd.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the sds for the half-normal hyperpriors on the sd hyperparameters of regression coefficients. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 10s.
`hist.disp.mean`	a scalar or a vector whose dimension is equal to the number of historical data sets giving the means for the half-normal hyperpriors on the dispersion parameters. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 0s.
`hist.disp.sd`	a scalar or a vector whose dimension is equal to the number of historical data sets giving the sds for the half-normal hyperpriors on the dispersion parameters. If a scalar is provided, same as for meta.mean.mean. Defaults to a vector of 10s.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.

`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Details

The RMAP is a mixture prior of two components where one component is a prior induced by the Bayesian hierarchical model (BHM), and the other is a vague (noninformative) prior. This function samples from the prior induced by the BHM.

Value

The function returns a matrix of the samples of regression coefficients from the prior induced by the BHM. The number of columns is equal to the number of regression coefficients, and the number of rows is equal to the number of MCMC samples.

See Also

`glm.rmap.bhm.approx()` for the second step and `glm.rmap()` for the final step of implementing RMAP.

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg036) ## historical data
  ## take subset for speed purposes
  actg036 = actg036[1:50, ]
  hist_data_list = list(actg036)
  glm.rmap.bhm(
    formula = outcome ~ scale(age) + race + treatment + scale(cd4),
    family = binomial('logit'),
    hist.data.list = hist_data_list,
    chains = 1, iter_warmup = 500, iter_sampling = 1000
  )
}
```

`glm.rmap.bhm.approx` *Posterior of robust meta-analytic predictive prior (RMAP)*

Description

Second step for sampling from the posterior distribution of a GLM using the RMAP by Schmidli et al. (2014) doi:10.1111/biom.12242.

Usage

```
glm.rmap.bhm.approx(samples.bhm, G = NULL, ...)
```

Arguments

`samples.bhm`	a matrix of the samples of regression coefficients from the prior induced by the BHM, output from the `glm.rmap.bhm()` function.
`G`	an integer vector giving the numbers of mixtures components (in the mixture approximation to the BHM) for which the BIC is to be calculated. Defaults to 1:9. See the argument `G` in `?mclust::Mclust`.
`...`	arguments passed to `mclust::Mclust()`. See `https://mclust-org.github.io/mclust/reference/Mclust.html`.

Details

This function approximates the distribution of the output "beta_pred" from the `glm.rmap.bhm()` function (i.e., samples from the prior induced by the Bayesian hierarchical model (BHM)) by a mixture of multivariate normal distributions. We use the `mclust` package by Scrucca et al. (2023) to implement the mixture approximation.

Value

The function returns a list giving the parameters estimated from the optimal (according to BIC) mixture model including the mixing proportions, the estimated mean and covariance matrix for each mixture component. An object of class `Mclust` from the mclust::Mclust() function is also included.

References

Scrucca, L., Fraley, C., Murphy, T. B., and Raftery, A. E. (2023). Model-Based Clustering, Classification, and Density Estimation Using mclust in R. Chapman and Hall/CRC. ISBN 978-1032234953. `https://mclust-org.github.io/book/`.

See Also

`glm.rmap.bhm()` for the first step and `glm.rmap()` for the final step of implementing RMAP.

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg036) ## historical data
  ## take subset for speed purposes
  actg036 = actg036[1:100, ]
  hist_data_list = list(actg036)
  samples_bhm = glm.rmap.bhm(
    formula = outcome ~ scale(age) + race + treatment + scale(cd4),
    family = binomial('logit'),
    hist.data.list = hist_data_list,
    chains = 1, iter_warmup = 1000, iter_sampling = 2000
  )$beta_pred
  glm.rmap.bhm.approx(
    samples.bhm = samples_bhm,
    G = 1:5, verbose = FALSE
```

```
  )
}
```

`IBCSG_curr` *International Breast Cancer Study Group (IBCSG) Trial VI Data*

Description

A data set from the IBCSG Trial VI investigating both the duration of adjuvant chemotherapy (3 versus 6 initial cycles of oral cyclophosphamide, methotrexate, and fluorouracil (CMF)) and the reintroduction of single courses of delayed chemotherapy in node-positive premenopausal breast cancer patients. The study results were described by IBCSG (1996) `doi:10.1200/JCO.1996.14.6.1885` and Hürny et al. (1992) `doi:10.1016/0959-8049(92)90399-m`. This data set only includes patients above the age of 40 (i.e., age $\geq$ 40) and treats the measurements of patients' physical well-being on month 18 as the outcome. The IBCSG_hist data set includes patients from the same study but with age < 40. We can use the IBCSG_hist data as the historical data and the IBCSG_curr data as the current data.

Usage

```
IBCSG_curr
```

Format

A data frame with 488 rows and 8 variables:

phys18 outcome variable, integer scores between 0 and 100 measuring the patients' physical well-being on month 18, with a higher score indicating a better physical well-being

phys1 physical well-being scores assessed at the start of the study

n_init_cycles number of initial cycles of CMF, equal to 3 or 6

reintroduction indicator of reintroduction of chemotherapy, 0 = no reintroduction, 1 = having reintroduction

age patient age in years

country country, ANZ = New Zealand/Australia, CH = Switzerland, SWED = Sweden

nodegp indicator of number of positive nodes being greater than or equal to 4, 0 = less than 4, 1 = 4+

ER estrogen receptor (ER) status indicator, 0 = negative, 1 = positive

References

International Breast Cancer Study Group. (1996). Duration and reintroduction of adjuvant chemotherapy for node-positive premenopausal breast cancer patients. Journal of Clinical Oncology, 14(6), 1885–1894.

Hürny, C., Bernhard, J., Gelber, R. D., Coates, A., Castiglione, M., Isley, M., Dreher, D., Peterson, H., Goldhirsch, A., and Senn, H.-J. (1992). Quality of life measures for patients receiving adjuvant therapy for breast cancer: An international trial. European Journal of Cancer, 28(1), 118–124.

Chi, Y.-Y. and Ibrahim, J. G. (2005). Joint models for multivariate longitudinal and Multivariate Survival Data. Biometrics, 62(2), 432–445.

`IBCSG_hist` *International Breast Cancer Study Group (IBCSG) Trial VI Data*

Description

A data set from the IBCSG Trial VI investigating both the duration of adjuvant chemotherapy (3 versus 6 initial cycles of oral cyclophosphamide, methotrexate, and fluorouracil (CMF)) and the reintroduction of single courses of delayed chemotherapy in node-positive premenopausal breast cancer patients. The study results were described by IBCSG (1996) `doi:10.1200/JCO.1996.14.6.1885` and Hürny et al. (1992) `doi:10.1016/0959-8049(92)90399-m`. This data set only includes patients under the age of 40 (i.e., age $<$ 40) and treats the measurements of patients' physical well-being on month 18 as the outcome. The IBCSG_curr data set includes patients from the same study but with age $\geq$ 40. We can use the IBCSG_hist data as the historical data and the IBCSG_curr data as the current data.

Usage

```
IBCSG_hist
```

Format

A data frame with 103 rows and 8 variables:

phys18 outcome variable, integer scores between 0 and 100 measuring the patients' physical well-being on month 18, with a higher score indicating a better physical well-being

phys1 physical well-being scores assessed at the start of the study

n_init_cycles number of initial cycles of CMF, equal to 3 or 6

reintroduction indicator of reintroduction of chemotherapy, 0 = no reintroduction, 1 = having reintroduction

age patient age in years

country country, ANZ = New Zealand/Australia, CH = Switzerland, SWED = Sweden

nodegp indicator of number of positive nodes being greater than or equal to 4, 0 = less than 4, 1 = 4+

ER estrogen receptor (ER) status indicator, 0 = negative, 1 = positive

References

International Breast Cancer Study Group. (1996). Duration and reintroduction of adjuvant chemotherapy for node-positive premenopausal breast cancer patients. Journal of Clinical Oncology, 14(6), 1885–1894.

Hürny, C., Bernhard, J., Gelber, R. D., Coates, A., Castiglione, M., Isley, M., Dreher, D., Peterson, H., Goldhirsch, A., and Senn, H.-J. (1992). Quality of life measures for patients receiving adjuvant therapy for breast cancer: An international trial. European Journal of Cancer, 28(1), 118–124.

Chi, Y.-Y. and Ibrahim, J. G. (2005). Joint models for multivariate longitudinal and Multivariate Survival Data. Biometrics, 62(2), 432–445.

`lm.npp` *Posterior of normalized power prior (NPP) for normal linear models*

Description

Sample from the posterior distribution of a normal linear model using the NPP by Duan et al. (2006) doi:10.1002/env.752. The power prior parameters (a_0's) are treated as random with independent beta priors. The current and historical data sets are assumed to have a common dispersion parameter (σ^2) with an inverse-gamma prior. Conditional on σ^2, the initial priors on the regression coefficients are independent normal distributions with variance $\propto (\sigma^2)^{-1}$. In this case, the normalizing constant for the NPP has a closed form.

Usage

```
lm.npp(
  formula,
  data.list,
  offset.list = NULL,
  beta.mean = NULL,
  beta.sd = NULL,
  sigmasq.shape = 2.1,
  sigmasq.scale = 1.1,
  a0.shape1 = 1,
  a0.shape2 = 1,
  a0.lower = NULL,
  a0.upper = NULL,
  iter_warmup = 1000,
  iter_sampling = 1000,
  chains = 4,
  ...
)
```

Arguments

`formula`	a two-sided formula giving the relationship between the response variable and covariates.

`data.list`	a list of `data.frames`. The first element in the list is the current data, and the rest are the historical data sets.
`offset.list`	a list of vectors giving the offsets for each data. The length of offset.list is equal to the length of data.list. The length of each element of offset.list is equal to the number of rows in the corresponding element of data.list. Defaults to a list of vectors of 0s.
`beta.mean`	a scalar or a vector whose dimension is equal to the number of regression coefficients giving the mean parameters for the initial prior on regression coefficients. If a scalar is provided, beta.mean will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`beta.sd`	a scalar or a vector whose dimension is equal to the number of regression coefficients. Conditional on the variance parameter sigmasq for the outcome, beta.sd * sqrt(sigmasq) gives the sd for the initial prior on regression coefficients. If a scalar is provided, same as for beta.mean. Defaults to a vector of 10s.
`sigmasq.shape`	shape parameter for inverse-gamma prior on variance parameter. Defaults to 2.1.
`sigmasq.scale`	scale parameter for inverse-gamma prior on variance parameter. Defaults to 1.1.
`a0.shape1`	first shape parameter for the i.i.d. beta prior on a0 vector. When `a0.shape1 == 1` and `a0.shape2 == 1`, a uniform prior is used.
`a0.shape2`	second shape parameter for the i.i.d. beta prior on a0 vector. When `a0.shape1 == 1` and `a0.shape2 == 1`, a uniform prior is used.
`a0.lower`	a scalar or a vector whose dimension is equal to the number of historical data sets giving the lower bounds for each element of the a0 vector. If a scalar is provided, a0.lower will be a vector of repeated elements of the given scalar. Defaults to a vector of 0s.
`a0.upper`	a scalar or a vector whose dimension is equal to the number of historical data sets giving the upper bounds for each element of the a0 vector. If a scalar is provided, same as for a0.lower. Defaults to a vector of 1s.
`iter_warmup`	number of warmup iterations to run per chain. Defaults to 1000. See the argument `iter_warmup` in `sample()` method in cmdstanr package.
`iter_sampling`	number of post-warmup iterations to run per chain. Defaults to 1000. See the argument `iter_sampling` in `sample()` method in cmdstanr package.
`chains`	number of Markov chains to run. Defaults to 4. See the argument `chains` in `sample()` method in cmdstanr package.
`...`	arguments passed to `sample()` method in cmdstanr package (e.g. seed, refresh, init).

Value

The function returns an object of class `draws_df` giving posterior samples.

References

Duan, Y., Ye, K., and Smith, E. P. (2005). Evaluating water quality using power priors to incorporate historical information. Environmetrics, 17(1), 95–106.

Examples

```
if (instantiate::stan_cmdstan_exists()) {
  data(actg019)
  data(actg036)
  data_list = list(currdata = actg019, histdata = actg036)
  lm.npp(
    formula = cd4 ~ treatment + age + race,
    data.list = data_list,
    chains = 1, iter_warmup = 500, iter_sampling = 1000
  )
}
```

Index