Package: BNSP (via r-universe)

October 12, 2026

Title Bayesian Non- And Semi-Parametric Model Fitting

Version 2.2.3

Date 2023-05-25

Author Georgios Papageorgiou

Maintainer Georgios Papageorgiou <gpapageo@gmail.com>

Description MCMC algorithms & processing functions for: 1. single response multiple regression, see Papageorgiou, G. (2018) <doi:10.32614/RJ-2018-069>, 2. multivariate response multiple regression, with nonparametric models for the means, the variances and the correlation matrix, with variable selection, see Papageorgiou, G. and Marshall, B. C. (2020) <doi:10.1080/10618600.2020.1739534>, 3. joint mean-covariance models for multivariate responses, see Papageorgiou, G. (2022) <doi:10.1002/sim.9376>, and 4.Dirichlet process mixtures, see Papageorgiou, G. (2019) <doi:10.1111/anzs.12273>.

Depends R (>= 3.5.0)

Imports coda, ggplot2, plot3D, threejs, gridExtra, cubature, Formula, plyr, mgcv, corrplot, label.switching

LinkingTo cubature

Suggests mvtnorm, np

License GPL (>= 2)

NeedsCompilation yes

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

Date/Publication 2023-05-25 09:40:07 UTC

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

RemoteRef HEAD

RemoteSha 413e0a42026002d565a3f9ade3617b75371f92f5

Contents

BNSP-package . . . 2
ami . . . 3
chol . . . 4
clustering . . . 5
continue . . . 6
dpmj . . . 7
histCorr . . . 14
lmrm . . . 15
mvrm . . . 19
mvrm2mcmc . . . 25
plot.mvrm . . . 26
plotCorr . . . 28
predict.mvrm . . . 29
print.mvrm . . . 30
s . . . 31
simD . . . 32
simD2 . . . 32
sinusoid . . . 33
sm . . . 35
summary.mvrm . . . 36
te . . . 37
ti . . . 38

Index **40**

`BNSP-package` *Bayesian non- and semi-parametric model fitting*

Description

Markov chain Monte Carlo algorithms for non- and semi-parametric models: 1. spike-slab variable selection in multivariate mean/variance regression models with function `mvrm`, 2. joint mean-covariance models for multivariate longitudinal responses with function `lmrm`, and 3. Dirichlet process mixture models with function `dpmj`.

Details

Package:	BNSP
Type:	Package
Version:	2.2.3
Date:	2023-05-25
License:	GPL (>=2)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

For details on the GNU General Public License see http://www.gnu.org/copyleft/gpl.html or write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Author(s)

Georgios Papageorgiou (2014)

Maintainer: Georgios Papageorgiou <gpapageo@gmail.com>

References

Papageorgiou, G. (2020). Bayesian semiparametric modelling of covariance matrices for multivariate longitudinal data. https://arxiv.org/abs/2012.09833

Papageorgiou, G. and Marshall, B.C. (2020). Bayesian semiparametric analysis of multivariate continuous responses, with variable selection. Journal of Computational and Graphical Statistics, 29:4, 896-909, DOI: 10.1080/10618600.2020.1739534

Papageorgiou, G. (2018). BNSP: an R Package for fitting Bayesian semiparametric regression models and variable selection. The R Journal, 10(2):526-548.

Papageorgiou, G. (2018). Bayesian density regression for discrete outcomes. Australian and New Zealand Journal of Statistics, arXiv:1603.09706v3 [stat.ME].

Papageorgiou, G., Richardson, S. and Best, N. (2015). Bayesian nonparametric models for spatially indexed data of mixed type. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 77:973-999.

---

ami *Amitriptyline dataset from Johnson and Wichern*

Description

Amitriptyline is a prescription antidepressant. The dataset consists of measurements on 17 patients who had over-dosed on amitriptyline.

Usage

```
data(ami)
```

Format

A data frame containing 17 rows and 7 columns. The columns represent

`tot` total blood plasma level.

`ami` amount of amitriptyline found in the plasma.

`gen` gender (1 for female).

`amt` amount of the drug taken.

`pr` PR wave measurement.

`bp` diastolic blood pressure.

`qrs` QRS wave measurement.

Source

Johnson, R. A., and Wichern, D. W. (2007), Applied Multivariate Statistical Analysis, Essex: Pearson, page 426.

References

Johnson, R. A., and Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*, Essex: Pearson.

`chol` *The Cholesky and modified Cholesky decompositions*

Description

Computes the Cholesky factorization and modified Cholesky factorizations of a real symmetric positive-definite square matrix.

Usage

```
chol(x, mod = TRUE, p = 1, ...)
```

Arguments

`x`	A symmetric, positive-definite matrix.
`mod`	Defaults to TRUE. With this choice, the function returns the modified Cholesky decomposition. When mod = FALSE, the function returns the usual Cholesky decomposition.
`p`	Relevant only when `mod = TRUE`. It determines the size of the blocks of the block diagonal matrix.
`...`	other arguments.

Details

The function computes the modified Cholesky decomposition of a real symmetric positive-definite square matrix Σ. This is given by

$$L\Sigma L^{\top} = D,$$

where L is a lower tringular matrix with ones on its main diagonal and D is a block diagonal matrix with block size determined by argument p.

Value

The function returns matrices L and D.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

The default function from **base**, chol

Examples

```
Sigma <- matrix(c(1.21,0.18,0.13,0.41,0.06,0.23,
                  0.18,0.64,0.10,-0.16,0.23,0.07,
                  0.13,0.10,0.36,-0.10,0.03,0.18,
                  0.41,-0.16,-0.10,1.05,-0.29,-0.08,
                  0.06,0.23,0.03,-0.29,1.71,-0.10,
                  0.23,0.07,0.18,-0.08,-0.10,0.36),6,6)
LD <- chol(Sigma)
L <- LD$L
D <- LD$D
round(L,5)
round(D,5)
solve(L) %*% D %*% solve(t(L))
LD <- chol(Sigma, p = 2)
L <- LD$L
D <- LD$D
round(L, 5)
round(D, 5)
solve(L) %*% D %*% solve(t(L))
```

---

clustering *Computes the similarity matrix*

Description

Computes the similarity matrix.

Usage

```
clustering(object, ...)
```

Arguments

object	an object of class "mvrm", usually a result of a call to mvrm.
...	other arguments.

Details

The function computes the similarity matrix for clustering based on corrrelations or variables.

Value

Similarity matrix.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

mvrm

Examples

```
#see \code{mvrm} example
```

---

continue *Continues the sampler from where it stopped*

Description

Allows the user to continue the sampler from the state it stopped in the previous call to mvrm.

Usage

```
continue(object, sweeps, burn = 0, thin, discard = FALSE,...)
```

Arguments

object	An object of class "mvrm", usually a result of a call to mvrm.
sweeps	The number of additional sweeps, maintaining the same thinning interval as specified in the original call to mvrm.
burn	length of burn-in period. Defaults to zero.
thin	thinning parameter. Defaults to the thinning parameter chosen for object.
discard	If set to true, the previous samples are discarded.
...	other arguments.

Details

The function allows the sampler to continue from the state it last stopped.

Value

The function returns an object of class `mvrm`.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

`mvrm`

Examples

```
#see \code{mvrm} example
```

`dpmj` *Dirichlet process mixtures of joint models*

Description

Fits Dirichlet process mixtures of joint response-covariate models, where the covariates are of mixed type while the discrete responses are represented utilizing continuous latent variables. See 'Details' section for a full model description and Papageorgiou (2018) for all technical details.

Usage

```
dpmj(formula, Fcdf, data, offset, sampler = "truncated", Xpred, offsetPred,
     StorageDir, ncomp, sweeps, burn, thin = 1, seed, H, Hdf, d, D,
     Alpha.xi, Beta.xi, Alpha.alpha, Beta.alpha, Trunc.alpha, ...)
```

Arguments

`formula`	a formula defining the response and the covariates e.g. `y ~ x`.
`Fcdf`	a description of the kernel of the response variable. Currently five options are supported: 1. "poisson", 2. "negative binomial", 3. "generalized poisson", 4. "binomial" and 5. "beta binomial". The first three kernels are used for count data analysis, where the third kernel allows for both over- and under-dispersion relative to the Poisson distribution. The last two kernels are used for binomial data analysis. See 'Details' section for some of the kernel details.
`data`	an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in 'data', the variables are taken from 'environment(formula)'.

`offset` this can be used to specify an a priori known component to be included in the model. This should be 'NULL' or a numeric vector of length equal to the sample size. One 'offset' term can be included in the formula, and if more are required, their sum should be used.

`sampler` the MCMC algorithm to be utilized. The two options are `sampler = "slice"` which implements a slice sampler (Walker, 2007; Papaspiliopoulos, 2008) and `sampler = "truncated"` which proceeds by truncating the countable mixture at `ncomp` components (see argument `ncomp`).

`Xpred` an optional design matrix the rows of which include the values of the covariates x for which the conditional distribution of $Y|x, D$ (where D denotes the data) is calculated. These are treated as 'new' covariates i.e. they do not contribute to the likelihood. The matrix shouldn't include a column of 1's. NA's can be included to obtain averaged effects.

`offsetPred` the offset term associated with the new covariates `Xpred`. It is of dimension one i.e. the same offset term is used for all rows of `Xpred`. If `Fcdf` is one of "poisson" or "negative binomial" or "generalized poisson", then `offsetPred` is the Poisson offset term. If `Fcdf` is one of "binomial" or "beta binomial", then `offsetPred` is the number of Binomial trials. If `offsetPred` is missing, it is taken to be the mean of `offset`, rounded to the nearest integer.

`StorageDir` a directory to store files with the posterior samples of models parameters and other quantities of interest. If a directory is not provided, files are created in the current directory and removed when the sampler completes.

`ncomp` number of mixture components. It defines where the countable mixture of densities [in (1) below] is truncated. Even if `sampler="slice"` is chosen, `ncomp` needs to be specified as it is used in the initialization process.

`sweeps` total number of posterior samples, including those discarded in burn-in period (see argument `burn`) and those discarded by the thinning process (see argument `thin`).

`burn` length of burn-in period.

`thin` thinning parameter.

`seed` optional seed for the random generator.

`H` optional scale matrix of the Wishart-like prior assigned to the restricted covariance matrices Σ_h^*. See 'Details' section.

`Hdf` optional degrees of freedom of the prior Wishart-like prior assigned to the restricted covariance matrices Σ_h^*. See 'Details' section.

`d` optional prior mean of the mean vector μ_h. See 'Details' section.

`D` optional prior covariance matrix of the mean vector μ_h. See 'Details' section.

`Alpha.xi` an optional parameter that depends on the specified `Fcdf` argument.

1. If `Fcdf = "poisson"`, this argument is parameter α_ξ of the prior of the Poisson rate: $\xi \sim \text{Gamma}(\alpha_\xi, \beta_\xi)$.
2. If `Fcdf = "negative binomial"`, this argument is a two-dimensional vector that includes parameters $\alpha_{1\xi}$ and $\alpha_{2\xi}$ of the priors: $\xi_1 \sim \text{Gamma}(\alpha_{1\xi}, \beta_{1\xi})$ and $\xi_2 \sim \text{Gamma}(\alpha_{2\xi}, \beta_{2\xi})$, where ξ_1 and ξ_2 are the two parameters of the Negative Binomial pmf.

3. If `Fcdf = "generalized poisson"`, this argument is a two-dimensional vector that includes parameters $\alpha_{1\xi}$ and $\alpha_{2\xi}$ of the priors: $\xi_1 \sim$ Gamma$(\alpha_{1\xi}, \beta_{1\xi})$ and $\xi_2 \sim$ N$(\alpha_{2\xi}, \beta_{2\xi})I[\xi_2 \in R_{\xi_2}]$, where ξ_1 and ξ_2 are the two parameters of the Generalized Poisson pmf. Parameter ξ_2 is restricted in the range $R_{\xi_2} = (0.05, \infty)$ as it is a dispersion parameter.
4. If `Fcdf = "binomial"`, this argument is parameter α_ξ of the prior of the Binomial probability: $\xi \sim$ Beta(α_ξ, β_ξ).
5. If `Fcdf = "beta binomial"`, this argument is a two-dimensional vector that includes parameters $\alpha_{1\xi}$ and $\alpha_{2\xi}$ of the priors: $\xi_1 \sim$ Gamma$(\alpha_{1\xi}, \beta_{1\xi})$ and $\xi_2 \sim$ Gamma$(\alpha_{2\xi}, \beta_{2\xi})$, where ξ_1 and ξ_2 are the two parameters of the Beta Binomial pmf.

See 'Details' section.

`Beta.xi` an optional parameter that depends on the specified family.

1. If `Fcdf = "poisson"`, this argument is parameter β_ξ of the prior of the Poisson rate: $\xi \sim$ Gamma(α_ξ, β_ξ).
2. If `Fcdf = "negative binomial"`, this argument is a two-dimensional vector that includes parameters $\beta_{1\xi}$ and $\beta_{2\xi}$ of the priors: $\xi_1 \sim$ Gamma$(\alpha_{1\xi}, \beta_{1\xi})$ and $\xi_2 \sim$ Gamma$(\alpha_{2\xi}, \beta_{2\xi})$, where ξ_1 and ξ_2 are the two parameters of the Negative Binomial pmf.
3. If `Fcdf = "generalized poisson"`, this argument is a two-dimensional vector that includes parameters $\beta_{1\xi}$ and $\beta_{2\xi}$ of the priors: $\xi_1 \sim$ Gamma$(\alpha_{1\xi}, \beta_{1\xi})$ and $\xi_2 \sim$ Normal$(\alpha_{2\xi}, \beta_{2\xi})I[\xi_2 \in R_{\xi_2}]$, where ξ_1 and ξ_2 are the two parameters of the Generalized Poisson pmf. Parameter ξ_2 is restricted in the range $R_{\xi_2} = (0.05, \infty)$ as it is a dispersion parameter. Note that $\beta_{2\xi}$ is a standard deviation.
4. If `Fcdf = "binomial"`, this argument is parameter β_ξ of the prior of the Binomial probability: $\xi \sim$ Beta(α_ξ, β_ξ).
5. If `Fcdf = "beta binomial"`, this argument is a two-dimensional vector that includes parameters $\beta_{1\xi}$ and $\beta_{2\xi}$ of the priors: $\xi_1 \sim$ Gamma$(\alpha_{1\xi}, \beta_{1\xi})$ and $\xi_2 \sim$ Gamma$(\alpha_{2\xi}, \beta_{2\xi})$, where ξ_1 and ξ_2 are the two parameters of the Beta Binomial pmf.

See 'Details' section.

`Alpha.alpha` optional shape parameter α_α of the Gamma prior assigned to the concentration parameter α. See 'Details' section.

`Beta.alpha` optional rate parameter β_α of the Gamma prior assigned to concentration parameter α. See 'Details' section.

`Trunc.alpha` optional truncation point c_α of the Gamma prior assigned to concentration parameter α. See 'Details' section.

`...` Other options that will be ignored.

Details

Function `dpmj` returns samples from the posterior distributions of the parameters of the model:

$$f(y_i, x_i) = \sum_{h=1}^{\infty} \pi_h f(y_i, x_i|\theta_h), \tag{1}$$

where y_i is a univariate discrete response, x_i is a p-dimensional vector of mixed type covariates, and $\pi_h, h \geq 1$, are obtained according to Sethuraman's (1994) stick-breaking construction: $\pi_1 = v_1$, and for $l \geq 2, \pi_l = v_l \prod_{j=1}^{l-1}(1 - v_j)$, where v_k are iid samples $v_k \sim$Beta $(1, \alpha), k \geq 1$.

Let Z denote a discrete variable (response or covariate). It is represented as discretized version of a continuous latent variable Z^*. Observed discrete Z and continuous latent variable Z^* are connected by:

$$z = q \iff c_{q-1} < z^* < c_q, q = 0, 1, 2, \ldots,$$

where the cut-points are obtained as: $c_{-1} = -\infty$, while for $q \geq 0$, $c_q = c_q(\lambda) = \Phi^{-1}\{F(q; \lambda)\}$. Here $\Phi(.)$ is the cumulative distribution function (cdf) of a standard normal variable and $F()$ denotes an appropriate cdf. Further, latent variables are assumed to independently follow a $N(0, 1)$ distribution, where the mean and variance are restricted to be zero and one as they are non-identifiable by the data. Choices for $F()$ are described next.

For counts, three options are supported. First, $F(.; \lambda_i)$ can be specified as the cdf of a Poisson$(H_i\xi_h)$ variable. Here $\lambda_i = (\xi_h, H_i)^T, \xi_h$ denotes the Poisson rate associated with cluster h, and H_i the offset term associated with sampling unit i. Second, $F(.; \lambda_i)$ can be specified as the negative binomial cdf, where $\lambda_i = (\xi_{1h}, \xi_{2h}, H_i)^T$. This option allows for overdispersion within each cluster relative to the Poisson distribution. Third, $F(.; \lambda_i)$ can be specified as the Generalized Poisson cdf, where, again, $\lambda_i = (\xi_{1h}, \xi_{2h}, H_i)^T$. This option allows for both over- and under-dispersion within each cluster.

For Binomial data, two options are supported. First, $F(.; \lambda_i)$ may be taken to be the cdf of a Binomial(H_i, ξ_h) variable, where ξ_h denotes the success probability of cluster h and H_i the number of trials associated with sampling unit i. Second, $F(.; \lambda_i)$ may be specified to be the beta-binomial cdf, where $\lambda = (\xi_{1h}, \xi_{2h}, H_i)^T$.

The special case of Binomial data is treated as

$$Z = 0 \iff z^* < 0, z^* \sim N(\mu_z^*, 1).$$

Details on all kernels are provided in the two tables below. The first table provides the probability mass functions and the mean in the presence of an offset term (which may be taken to be one). The column 'Sample' indicates for which parameters the routine provides posterior samples. The second table provides information on the assumed priors along with the default values of the parameters of the prior distributions and it also indicates the function arguments that allow the user to alter these.

Kernel	PMF	Offset	Mean	Sample
Poisson	$\exp(-H\xi)(H\xi)^y/y!$	H	$H\xi$	ξ
Negative Binomial	$\frac{\Gamma(y+\xi_1)}{\Gamma(\xi_1)\Gamma(y+1)}(\frac{\xi_2}{H+\xi_2})^{\xi_1}(\frac{H}{H+\xi_2})^y$	H	$H\xi_1/\xi_2$	ξ_1, ξ_2
Generalized Poisson	$\xi_1\{\xi_1 + (\xi_2 - 1)y\}^{y-1}\xi_2^{-y} \times \exp\{-[\xi_1 + (\xi_2 - 1)y]/\xi_2\}/y!$	H	$H\xi_1$	ξ_1, ξ_2
Binomial	$\binom{N}{y}\xi^y(1-\xi)^{N-y}$	N	$N\xi$	ξ
Beta Binomial	$\binom{N}{y}\frac{Beta(y+\xi_1, N-y+\xi_2)}{Beta(\xi_1,\xi_2)}$	N	$N\xi_1/(\xi_1 + \xi_2)$	ξ_1, ξ_2

Kernel	Priors	Default Values
Poisson	$\xi \sim$ Gamma(α_ξ, β_ξ)	Alpha.xi = 1.0, Beta.xi = 0.1
Negative Binomial	$\xi_i \sim$ Gamma$(\alpha_{\xi_i}, \beta_{\xi_i}), i = 1, 2$	Alpha.xi = c(1.0,1.0), Beta.xi = c(0.1,0.1)

Generalized Poisson	$\xi_1 \sim$ Gamma$(\alpha_{\xi_1}, \beta_{\xi_1})$ $\xi_2 \sim$ N$(\alpha_{\xi_2}, \beta_{\xi_2})I[\xi_2 > 0.05]$ where β_{ξ_2} denotes st.dev.	Alpha.xi = c(1.0,1.0), Beta.xi = c(0.1,1.0)
Binomial	$\xi \sim$ Beta(α_ξ, β_ξ)	Alpha.xi = 1.0, Beta.xi = 1.0
Beta Binomial	$\xi_i \sim$ Gamma$(\alpha_{\xi_i}, \beta_{\xi_i}), i = 1, 2$	Alpha.xi = c(1.0,1.0), Beta.xi = c(0.1,0.1)

Let $z_i = (y_i, x_i^T)^T$ denote the joint vector of observed continuous and discrete variables and z_i^* the corresponding vector of continuous observed and latent variables. With θ_h denoting model parameters associated with the hth cluster, the joint density $f(z_i|\theta_h)$ takes the form

$$f(z_i|\theta_h) = \int_{R(y)} \int_{R(x_d)} N_q(z_i^*; \mu_h^*, \Sigma_h^*) dx_d^* dy^*,$$

where

$$\mu_h^* = \begin{pmatrix} 0 \\ \mu_h \end{pmatrix}, \quad \Sigma_h^* = \begin{bmatrix} C_h & \nu_h^T \\ \nu_h & \Sigma_h \end{bmatrix},$$

where C_h is the covariance matrix of the latent continuous variables and it has diagonal elements equal to one i.e. it is a correlation matrix.

In addition to the priors defined in the table above, we specify the following:

1. The restricted covariance matrix Σ_h^* is assigned a prior distribution that is based on the Wishart distribution with degrees of freedom set by default to dimension of matrix plus two and diagonal scale matrix, with the sub-matrix that corresponds to discrete variables taken to be the identity matrix and with sub-matrix that corresponds to continuous variables having entries equal to 1/8 of the square of the observed data range. Default values can be changed using arguments `H` and `Hdf`.
2. The prior on μ_h, the non-zero part of μ_h^*, is taken to be multivariate normal $\mu_h \sim N(d, D)$. The mean d is taken to be equal to the center of the dataset. The covariance matrix D is taken to be diagonal. Its elements that correspond to continuous variables are set equal to 1/8 of the square of the observed data range while the elements that correspond to binary variables are set equal to 5. Arguments `Mu.mu` and `Sigma.mu` allow the user to change the default values.
3. The concentration parameter α is assigned a Gamma$(\alpha_\alpha, \beta_\alpha)$ prior over the range (c_α, ∞), that is, $f(\alpha) \propto \alpha^{\alpha_\alpha - 1} \exp\{-\alpha\beta_\alpha\} I[\alpha > c_\alpha]$, where $I[.]$ is the indicator function. The default values are $\alpha_\alpha = 2.0, \beta_\alpha = 5.0$, and $c_\alpha = 0.25$. Users can alter the default using using arguments `Alpha.alpha`, `Beta.alpha` and `Turnc.alpha`.

Value

Function `dpmj` returns the following:

`call`	the matched call.
`seed`	the seed that was used (in case replication of the results is needed).
`meanReg`	if `Xpred` is specified, the function returns the posterior mean of the conditional expectation of the response y given each new covariate x.
`medianReg`	if `Xpred` is specified, the function returns the posterior mean of the conditional 50% quantile of the response y given each new covariate x.

`q1Reg` if `Xpred` is specified, the function returns the posterior mean of the conditional 25% quantile of the response y given each new covariate x.

`q3Reg` if `Xpred` is specified, the function returns the posterior mean of the conditional 75% quantile of the response y given each new covariate x.

`modeReg` if `Xpred` is specified, the function returns the posterior mean of the conditional mode of the response y given each new covariate x.

`denReg` if `Xpred` is specified, the function returns the posterior mean conditional density of the response y given each new covariate x. Results are presented in a matrix the rows of which correspond to the different xs.

`denVar` if `Xpred` is specified, the function returns the posterior variance of the conditional density of the response y given each new covariate x. Results are presented in a matrix the rows of which correspond to the different xs.

Further, function `dpmj` creates files where the posterior samples are written. These files are (with all file names preceded by 'BNSP.'):

`alpha.txt` this file contains samples from the posterior of the concentration parameters α. The file is arranged in `(sweeps-burn)/thin` lines and one column, each line including one posterior sample.

`compAlloc.txt` this file contains the allocations to clusters obtained during posterior sampling. It consists of `(sweeps-burn)/thin` lines, that represent the posterior samples, and n columns, that represent the sampling units. Clusters are represented by integers ranging from 0 to `ncomp`-1.

`MeanReg.txt` this file contains the conditional means of the response y given covariates x obtained during posterior sampling. The rows represent the `(sweeps-burn)/thin` posterior samples. The columns represent the various covariate values x for which the means are obtained.

`MedianReg.txt` this file contains the 50% conditional quantile of the response y given covariates x obtained during posterior sampling. The rows represent the `(sweeps-burn)/thin` posterior samples. The columns represent the various covariate values x for which the medians are obtained.

`muh.txt` this file contains samples from the posteriors of the p-dimensional mean vectors $\mu_h, h = 1, 2, \ldots,$`ncomp`. The file is arranged in `((sweeps-burn)/thin)*ncomp` lines and p columns. In more detail, sweeps create `ncomp` lines representing samples $\mu_h^{(sw)}, h = 1, \ldots,$`ncomp`, where superscript sw represents a particular sweep. The elements of $\mu_h^{(sw)}$ are written in the columns of the file.

`nmembers.txt` this file contains `(sweeps-burn)/thin` lines and `ncomp` columns, where the lines represent posterior samples while the columns represent the components or clusters. The entries represent the number of sampling units allocated to each component.

`Q05Reg.txt` this file contains the 5% conditional quantile of the response y given covariates x obtained during posterior sampling. The rows represent the `(sweeps-burn)/thin` posterior samples. The columns represent the various covariate values x for which the quantiles are obtained.

`Q10Reg.txt` as above, for the 10% conditional quantile.

Q15Reg.txt	as above, for the 15% conditional quantile.
Q20Reg.txt	as above, for the 20% conditional quantile.
Q25Reg.txt	as above, for the 25% conditional quantile.
Q75Reg.txt	as above, for the 75% conditional quantile.
Q80Reg.txt	as above, for the 80% conditional quantile.
Q85Reg.txt	as above, for the 85% conditional quantile.
Q90Reg.txt	as above, for the 90% conditional quantile.
Q95Reg.txt	as above, for the 95% conditional quantile.
Sigmah.txt	this file contains samples from the posteriors of the $q \times q$ restricted covariance matrices $\Sigma_h^*, h = 1, 2, \ldots,$ncomp. The file is arranged in ((sweeps-burn)/thin)*ncomp lines and q^2 columns. In more detail, sweeps create ncomp lines representing samples $\Sigma_h^{(sw)}, h = 1, \ldots,$ncomp, where superscript sw represents a particular sweep. The elements of $\Sigma_h^{(sw)}$ are written in the columns of the file.
xih.txt	this file contains samples from the posteriors of parameters $\xi_h, h = 1, 2, \ldots,$ncomp. The file is arranged in ((sweeps-burn)/thin)*ncomp lines and one or two columns, depending on the number of parameters in the selected Fcdf. Sweeps write in the file ncomp lines representing samples $\xi_h^{(sw)}, h = 1, \ldots,$ncomp, where superscript sw represents a particular sweep.
Updated.txt	this file contains (sweeps-burn)/thin lines with the number of components updated at each iteration of the sampler (relevant for slice sampling).

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

References

Consul, P. C. & Famoye, G. C. (1992). Generalized Poisson regression model. Communications in Statistics - Theory and Methods, 1992, 89-109.

Papageorgiou, G. (2018). Bayesian density regression for discrete outcomes. arXiv:1603.09706v3 [stat.ME].

Papaspiliopoulos, O. (2008). A note on posterior sampling from Dirichlet mixture models. Technical report, University of Warwick.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. Statistica Sinica, 4, 639-650.

Walker, S. G. (2007). Sampling the Dirichlet mixture model with slices. Communications in Statistics Simulation and Computation, 36(1), 45-54.

Examples

```
#Bayesian nonparametric joint model with binomial response Y and one predictor X
data(simD)
pred<-seq(with(simD,min(X))+0.1,with(simD,max(X))-0.1,length.out=30)
npred<-length(pred)
# fit1 and fit2 define the same model but with different numbers of
# components and posterior samples
```

```
fit1 <- dpmj(cbind(Y,(E-Y))~X, Fcdf="binomial", data=simD, ncomp=10, sweeps=20,
             burn=10, sampler="truncated", Xpred=pred, offsetPred=30)
fit2 <- dpmj(cbind(Y,(E-Y))~X, Fcdf="binomial", data=simD, ncomp=50, sweeps=5000,
               burn=1000, sampler="truncated", Xpred=pred, offsetPred=30)
plot(with(simD,X),with(simD,Y)/with(simD,E))
lines(pred,fit2$medianReg/30,col=3,lwd=2)
# with discrete covariate
simD<-data.frame(simD,Xd=sample(c(0,1),300,replace=TRUE))
pred<-c(0,1)
fit3 <- dpmj(cbind(Y,(E-Y))~Xd, Fcdf="binomial", data=simD, ncomp=10, sweeps=20,
             burn=10, sampler="truncated", Xpred=pred, offsetPred=30)
```

`histCorr` *Creates plots of correlation matrices*

Description

This function plots the posterior distribution of the elements of correlation matrices.

Usage

```
histCorr(x, term = "R", plotOptions = list(),...)
```

Arguments

`x`	an object of class 'mvrm', as generated by function `mvrm`.
`term`	Admits two possible values: "R" to plot samples from the posterior of the correlation matrix R, and "muR" to plot samples from the posterior of the means μ_R.
`plotOptions`	ggplot type options.
`...`	other arguments.

Details

Use this function to visualize the elements of a correlation matrix.

Value

Posterior distributions of elements of correlation matrices.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

`mvrm`

Examples

```
#see \code{mvrm} example
```

`lmrm` *Bayesian semiparametric modelling of covariance matrices for multivariate longitudinal data*

Description

Implements an MCMC algorithm for posterior sampling based on a semiparametric model for continuous longitudinal multivariate responses. The overall model consists of 5 regression submodels and it utilizes spike-slab priors for variable selection and function regularization. See 'Details' section for a full description of the model.

Usage

```
lmrm(formula, data = list(), centre=TRUE, id, time,
sweeps, burn = 0, thin = 1, seed, StorageDir,
c.betaPrior = "IG(0.5,0.5*n*p)", pi.muPrior = "Beta(1,1)",
c.alphaPrior = "IG(1.1,1.1)", pi.phiPrior = "Beta(1,1)", c.psiPrior = "HN(2)",
sigmaPrior = "HN(2)", pi.sigmaPrior = "Beta(1,1)",
corr.Model = c("common", nClust = 1), DP.concPrior = "Gamma(5,2)",
c.etaPrior = "IG(0.5,0.5*samp)", pi.nuPrior = "Beta(1,1)",
pi.fiPrior = "Beta(1,1)", c.omegaPrior = "IG(1.1,1.1)", sigmaCorPrior = "HN(2)",
tuneCalpha, tuneSigma2, tuneCbeta, tuneAlpha, tuneSigma2R, tuneR, tuneCpsi,
tuneCbCor, tuneOmega, tuneComega, tau, FT = 1,...)
```

Arguments

`formula`	a formula defining the responses and the covariates in the 5 regression models e.g. `y1 | y2 ~ x | w | z | t | t` or for smooth effects `y1 | y2 ~ sm(x) | sm(w) | sm(z) | sm(t) | sm(t)`. The package uses the extended formula notation, where the responses are defined on the left of ~ and the models on the right.
`data`	a data frame.
`centre`	Binary indicator. If set equal to TRUE, the design matrices are centred, to have column mean equl to zero, otherwise, if set to FALSE, the columns are not centred.
`id`	identifiers of the individuals or other sampling units that are observed over time.
`time`	a vector input that specifies the time of observation
`sweeps`	total number of posterior samples, including those discarded in burn-in period (see argument `burn`) and those discarded by the thinning process (see argument `thin`).
`burn`	length of burn-in period.
`thin`	thinning parameter.

`seed` optional seed for the random generator.

`StorageDir` a required directory to store files with the posterior samples of models parameters.

`c.betaPrior` The inverse Gamma prior of c_β. The default is "IG(0.5,0.5*n*p)", that is, an inverse Gamma with parameters $1/2$ and $np/2$, where n is the number of sampling units and p is the length of the response vector.

`pi.muPrior` The Beta prior of π_μ. The default is "Beta(1,1)". It can be of dimension 1, of dimension K (the number of effects that enter the mean model), or of dimension pK

`c.alphaPrior` The inverse Gamma prior of c_α^2. The default is "IG(1.1,1.1)". Half-normal priors for c_α are also available, declared using "HN(a)", where "a" is a positive number. It can be of dimension 1 or p (the length of the multivariate response).

`pi.phiPrior` The Beta prior of π_ϕ. The default is "Beta(1,1)". It can be of dimension 1, of dimension B (the number of effects that enter the dependence model), or of dimension p^2B

`c.psiPrior` The prior of c_ψ^2. The default is "HN(2)", a half-normal prior for c_ψ with variance equal to two, $c_\psi \sim N(0,2)I[c_\psi > 0]$. Inverse Gamma priors for c_ψ^2 are also available, declared using "IG(a,b)". It can be of dimension 1 or p^2 (the number of dependence models).

`sigmaPrior` The prior of $\sigma_k^2, k = 1, \ldots, p$. The default is "HN(2)", a half-normal prior for σ_k with variance equal to two, $\sigma_k \sim N(0,2)I[\sigma > 0]$. Inverse Gamma priors for σ_k^2 are also available, declared using "IG(a,b)". It can be of dimension 1 or p (the length of the multivariate response).

`pi.sigmaPrior` The Beta prior of π_σ. The default is "Beta(1,1)". It can be of dimension 1, of dimension L (the number of effects that enter the variance model), or of dimension pL

`corr.Model` Specifies the model for the correlation matrices R_t. The three choices supported are "common", that specifies a common correlations model, "groupC", that specifies a grouped correlations model, and "groupV", that specifies a grouped variables model. When the model chosen is either "groupC" or "groupV", the upper limit on the number of clusters can also be specified, using corr.Model = c("groupC", nClust = d) or corr.Model = c("groupV", nClust = p). If the number of clusters is left unspecified, for the "groupV" model, it is taken to be p, the number of responses. For the "groupC" model, it is taken to be $d = p(p-1)/2$, the number of free elements in the correlation matrices.

`DP.concPrior` The Gamma prior for the Dirichlet process concentration parameter.

`c.etaPrior` The inverse Gamma prior of c_η. The default is "IG(0.5,0.5*samp)", that is, an inverse Gamma with parameters $1/2$ and $samp/2$, where $samp$ is the number of correlations observed over time, that is $samp=M*d$ where M is the number of unique observation time points and d is the number of non-redundant elements of R.

`pi.nuPrior` The Beta prior of π_ν. The default is "Beta(1,1)". It can be of dimension 1.

`pi.fiPrior` The Beta prior of π_φ. The default is "Beta(1,1)". It can be of dimension 1.

`c.omegaPrior` The prior of c_ω^2. The default is "HN(2)", a half-normal prior for c_ω with variance equal to two, $c_\omega \sim N(0,2)I[c_\omega > 0]$. Inverse Gamma priors for c_ω^2 are also available, declared using "IG(a,b)". It can be of dimension 1.

`sigmaCorPrior` The prior of σ^2. The default is "HN(2)", a half-normal prior for σ^2 with variance equal to two, $\sigma \sim N(0,2)I[\sigma > 0]$. Inverse Gamma priors for σ^2 are also available, declared using "IG(a,b)". It can be of dimension 1.

`tuneCalpha` Starting value of the tuning parameter for sampling $c_{\alpha k}, k = 1, \ldots, p$. Defaults at a vector of \$p\$ ones. It could be of dimension p.

`tuneSigma2` Starting value of the tuning parameter for sampling $\sigma_k^2, k = 1, \ldots, p$. Defaults at a vector of \$p\$ ones. It could be of dimension p.

`tuneCbeta` Starting value of the tuning parameter for sampling c_β. Defaults at 100.

`tuneAlpha` Starting value of the tuning parameter for sampling regression coefficients of the variance models $\alpha_k, k = 1, \ldots, p$. Defaults at a vector of 5s. It could be of dimension Lp

`tuneSigma2R` Starting value of the tuning parameter for sampling σ^2. Defaults at 1.

`tuneR` Starting value of the tuning parameter for sampling correlation matrices. Defaults at $40 * (p + 2)^3$. Can be of dimension 1 or M is the number of unique observation time points.

`tuneCpsi` Starting value of the tuning parameter for sampling variances c_ψ^2. Defaults at 5. Can be of dimension 1 or p^2

.

`tuneCbCor` Starting value of the tuning parameter for sampling c_η^2. Defaults at 10.

`tuneOmega` Starting value of the tuning parameter for sampling regression coefficients of the variance models ω. Defaults at 5.

`tuneComega` Starting value of the tuning parameter for sampling c_ω. Defaults at 1.

`tau` The tau of the shadow prior. Defaults at 0.01.

`FT` Binary indicator. If set equal to 1, the Fisher's z transform of the correlations is modelled, otherwise if set equal to 0, the untransformed correlations are modelled.

`...` Other options that will be ignored.

Details

Function `lmrm` returns samples from the posterior distributions of the parameters of a regression model with normally distributed multivariate longitudinal responses. To describe the model, let $Y_{ij} = (Y_{ij1}, \ldots, Y_{ijp})^\top$ denote the vector of p responses observed on individual $i, i = 1, \ldots, n$, at time point $t_{ij}, j = 1, \ldots, n_i$. The observational time points t_{ij} are allowed to be unequally spaced. Further, let u_{ij} denote the covariate vector that is observed along with Y_{ij} and that may include time, other time-dependent covariates and time-independent ones. In addition, let $Y_i = (Y_{i1}^\top, \ldots, Y_{in_i}^\top)^\top$ denote the ith response vector. With $\mu_i = E(Y_i)$ and $\Sigma_i = \text{cov}(Y_i)$, the model assumes multivariate normality, $Y_i \sim N(\mu_i, \Sigma_i), i = 1, 2, \ldots, n$. The means μ_i and covariance matrices Σ_i are modelled semiparametrically in terms of covariates. For the means one can specify semiparametric models,

$$\mu_{ijk} = \beta_{k0} + \sum_{l=1}^{K_1} u_{ijl}\beta_{kl} + \sum_{l=K_1+1}^{K} f_{\mu,k,l}(u_{ijl}).$$

This is the first of the 5 regression submodels.

To model the covariance matrix, first consider the modified Cholesky decomposition, $L_i \Sigma_i L_i^\top = D_i$, where L_i is a unit block lower triangular matrix and D_i is a block diagonal matrix,

$$L_i = \begin{bmatrix} I & 0 & \dots & 0 \\ -\Phi_{i21} & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\Phi_{in_i1} & -\Phi_{in_i1} & \dots & I \end{bmatrix}, \quad D_i = \begin{bmatrix} D_1 & 0 & \dots & 0 \\ 0 & D_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & D_{n_i} \end{bmatrix},$$

For modelling $D_{ij}, i = 1, \dots, n, j = 1, \dots, n_i$ in terms of covariates, first we separate the variances and the correlations $D_{ij} = S_{ij}^{1/2} R_{ij} S_{ij}^{1/2}$. It is easy to model matrix S_{ij} in terms of covariates as the only requirement on its diagonal elements is that they are nonnegative,

$$\log \sigma_{ijk}^2 = \alpha_{k0} + \sum_{l=1}^{L_1} w_{ijl} \alpha_{kl} + \sum_{l=L_1+1}^{L} f_{\sigma,k,l}(w_{ijl})$$

This is the second of the 5 regression submodels.

For ϕ_{ijklm}, the (l, m) element of $\Phi_{ijk}, l, m = 1, \dots, p$, one can specify semiparametric models

$$\phi_{ijklm} = \psi_{lm0} + \sum_{b=1}^{B_1} v_{ijkb} \psi_{lmb} + \sum_{b=B_1+1}^{B} f_{\phi,l,m,b}(v_{ijkb})$$

This is the third of the 5 regression submodels.

The elements of the correlations matrices R_{ij} are modelled in terms of covariate time only, hence they are denoted by R_t. Subject to the positive definiteness constraint, the elements of R_t are modelled using a normal distribution with location and scale parameters, μ_{ct} and σ_{ct}^2, modelled as

$$\mu_{ct} = \eta_0 + f_\mu(t),$$

$$\log \sigma_{ct}^2 = \omega_0 + f_\sigma(t),$$

and these are the last 2 of the 5 submodels.

Value

Function `lmrm` returns samples from the posteriors of the model parameters.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

References

Papageorgiou, G. (2020). Bayesian semiparametric modelling of covariance matrices for multivariate longitudinal data. arXiv:2012.09833.

Examples

```
# Fit a joint mean-covariance model on the simulated dataset simD2
require(ggplot2)
data(simD2)
model <- Y1 | Y2 ~ time | sm(time) | sm(lag) | sm(time) | 1
# the above defines the responses and the regression models on the left and
# right of "~", respectively
# the first model, for the mean, is a linear function of time, this is sufficient as
# the 2 responses have constant mean.
# the second model, for the variances, is a smooth function of time
# the third model, for the dependence structure, is a smooth function of lag,
# that lmrm figures out and it does not need to be computed by the user
# the fourth model, for location of the correlations, is a smooth function of time
# the fifth model, for scale of the correlations, is just an intercept model
## Not run:
m1 <- lmrm(formula = model, corr.Model = c("common", nClust = 1), data = simD2,
        id = id, time = time, sweeps = 2500, burn = 500, thin = 2,
        StorageDir = getwd(), seed = 1)
plot(m1)

## End(Not run)
```

`mvrm` *Bayesian semiparametric analysis of multivariate continuous responses, with variable selection*

Description

Implements an MCMC algorithm for posterior sampling based on a semiparametric model for continuous multivariate responses and additive models for the mean and variance functions. The model utilizes spike-slab priors for variable selection and regularization. See 'Details' section for a full description of the model.

Usage

```
mvrm(formula, distribution = "normal", data = list(), centre = TRUE, sweeps,
burn = 0, thin = 1, seed, StorageDir, c.betaPrior = "IG(0.5, 0.5 * n * p)",
pi.muPrior = "Beta(1, 1)", c.alphaPrior = "IG(1.1, 1.1)", sigmaPrior = "HN(2)",
pi.sigmaPrior = "Beta(1, 1)", c.psiPrior = "HN(1)", phiPrior = "HN(2)",
pi.omegaPrior = "Beta(1, 1)", mu.RPrior = "N(0, 1)", sigma.RPrior = "HN(1)",
corr.Model = c("common", nClust = 1), DP.concPrior = "Gamma(5, 2)",
breaksPrior = "SBeta(1, 2)", tuneCbeta, tuneCalpha, tuneAlpha, tuneSigma2,
tuneCpsi, tunePsi, tunePhi, tuneR, tuneSigma2R, tuneHar, tuneBreaks, tunePeriod, tau,
FT = 1, compDeviance = FALSE, ...)
```

Arguments

formula	a formula defining the responses and the covariates in the mean and variance models e.g. y1 | y2 ~ x | z or for smooth effects y1 | y2 ~ sm(x) | sm(z). The package uses the extended formula notation, where the responses are defined on the left of ~ and the mean and variance models on the right.
distribution	The distribution for the response variables. Currently two options are supported: "normal" and "t".
data	a data frame.
centre	Binary indicator. If set equal to TRUE, the design matrices are centred, to have column mean equl to zero, otherwise, if set to FALSE, the columns are not centred.
sweeps	total number of posterior samples, including those discarded in burn-in period (see argument burn) and those discarded by the thinning process (see argument thin).
burn	length of burn-in period.
thin	thinning parameter.
seed	optional seed for the random generator.
StorageDir	a required directory to store files with the posterior samples of models parameters.
c.betaPrior	The inverse Gamma prior of c_β. The default is "IG(0.5,0.5*n*p)", that is, an inverse Gamma with parameters $1/2$ and $np/2$, where n is the number of sampling units and p is the length of the response vector.
pi.muPrior	The Beta prior of π_μ. The default is "Beta(1,1)". It can be of dimension 1, of dimension K (the number of effects that enter the mean model), or of dimension pK
c.alphaPrior	The prior of c_α. The default is "IG(1.1,1.1)". Half-normal priors for $\sqrt{c_\alpha}$ are also available, declared using "HN(a)", where "a" is a positive number. It can be of dimension 1 or p (the length of the multivariate response).
sigmaPrior	The prior of σ. The default is "HN(2)", a half-normal prior for σ with variance equal to two, $\sigma \sim N(0,2)I[\sigma > 0]$. Inverse Gamma priors for σ^2 are also available, declared using "IG(a,b)". It can be of dimension 1 or p (the length of the multivariate response).
pi.sigmaPrior	The Beta prior of π_σ. The default is "Beta(1,1)". It can be of dimension 1, of dimension Q (the number of effects that enter the variance model), or of dimension pQ
c.psiPrior	The prior of c_ψ. The default is half-normal for $\sqrt{c_\psi}$, declared using "HN(a)", where "a" is a positive number. The default value for "a" is one. Inverse-gamma priors are also available and they can be declared using "IG(a,b)", where "a" and "b" are positive constants. The prior can be of dimension 1 or p (the length of the multivariate response).
phiPrior	The prior of $varphi^2$. The default is half-normal for $varphi$, declared using "HN(a)", where "a" is a positive number. The default value for "a" is two. Inverse-gamma priors are also available and they can be declared using "IG(a,b)", where "a" and "b" are positive constants. The prior can be of dimension 1 or p (the length of the multivariate response).

pi.omegaPrior	The Beta prior of π_ω. The default is "Beta(1,1)". It can be of dimension 1, of dimension B (the number of effects that enter the shape parameter model), or of dimension pB
mu.RPrior	The normal prior for μ_R. The default is the standard normal distribution.
sigma.RPrior	The half normal prior for σ_R. The default is the half normal distribution with variance one.
corr.Model	Specifies the model for the correlation matrix R. The three choices supported are "common", that specifies a common correlations model, "groupC", that specifies a grouped correlations model, and "groupV", that specifies a grouped variables model. When the model chosen is either "groupC" or "groupV", the upper limit on the number of clusters can also be specified, using corr.Model = c("groupC", nClust = d) or corr.Model = c("groupV", nClust = p). If the number of clusters is left unspecified, for the "groupV" model, it is taken to be p, the number of responses. For the "groupC" model, it is taken to be $d = p(p-1)/2$, the number of free elements in the correlation matrix.
DP.concPrior	The Gamma prior for the Dirichlet process concentration parameter.
breaksPrior	The prior for the shifts associated with the growth break points. The shift is taken to have a scaled Beta prior with support the (0, p) interval, where p is the period of the sin curve. The default SBeta(1, 2) is a scaled Beta(1, 2) distribution supported in the (0, p) interval. The shifts are in increasing order.
tuneCbeta	Starting value of the tuning parameter for sampling c_β. Defaults at 20.
tuneCalpha	Starting value of the tuning parameter for sampling c_α. Defaults at 1.
tuneAlpha	Starting value of the tuning parameter for sampling regression coefficients of the variance model α. Defaults at 5.
tuneSigma2	Starting value of the tuning parameter for sampling variances σ_j^2. Defaults at 1.
tuneCpsi	Starting value of the tuning parameter for sampling c_ψ. Defaults at 1.
tunePsi	Starting value of the tuning parameter for sampling ψ. Defaults at 5.
tunePhi	Starting value of the tuning parameter for sampling φ. Defaults at 0.5.
tuneR	Starting value of the tuning parameter for sampling correlation matrices. Defaults at $40 * (p + 2)^3$.
tuneSigma2R	Starting value of the tuning parameter for sampling σ_R^2. Defaults at 1.
tuneHar	Starting value of the tuning parameter for sampling the regression coefficients of the harmonics. Defaults at 100.
tuneBreaks	Starting value of the tuning parameter for sampling the shift parameters associated with growth breaks. Defaults at 0.01 times the period of the sin wave.
tunePeriod	Starting value of the tuning parameter for sampling the period parameter of the sin curve. Defaults to 0.01.
tau	The tau of the shadow prior. Defaults at 0.01.
FT	Binary indicator. If set equal to 1, the Fisher's z transform of the correlations is modelled, otherwise if set equal to 0, the untransformed correlations are modelled.
compDeviance	Binary indicator. If set equal to 1, the deviance is computed.
...	Other options that will be ignored.

Details

Function `mvrm` returns samples from the posterior distributions of the parameters of a regression model with normally distributed multivariate responses and mean and variance functions modeled in terms of covariates. For instance, in the presence of two responses (y_1, y_2) and two covariates in the mean model (u_1, u_2) and two in the variance model (w_1, w_2), we may choose to fit

$$\mu_u = \beta_0 + \beta_1 u_1 + f_\mu(u_2),$$

$$\log(\sigma^2_W) = \alpha_0 + \alpha_1 w_1 + f_\sigma(w_2),$$

parametrically modelling the effects of u_1 and w_1 and non-parametrically modelling the effects of u_2 and w_2. Smooth functions, such as f_μ and f_σ, are represented by basis function expansion,

$$f_\mu(u_2) = \sum_j \beta_j \phi_j(u_2),$$

$$f_\sigma(w_2) = \sum_j \alpha_j \phi_j(w_2),$$

where ϕ are the basis functions and β and α are regression coefficients.

The variance model can equivalently be expressed as

$$\sigma^2_W = \exp(\alpha_0)\exp(\alpha_1 w_1 + f_\sigma(w_2)) = \sigma^2 \exp(\alpha_1 w_1 + f_\sigma(w_2)),$$

where $\sigma^2 = \exp(\alpha_0)$. This is the parameterization that we adopt in this implementation.

Positive prior probability that the regression coefficients in the mean model are exactly zero is achieved by defining binary variables γ that take value $\gamma = 1$ if the associated coefficient $\beta \neq 0$ and $\gamma = 0$ if $\beta = 0$. Indicators δ that take value $\delta = 1$ if the associated coefficient $\alpha \neq 0$ and $\delta = 0$ if $\alpha = 0$ for the variance function are defined analogously. We note that all coefficients in the mean and variance functions are subject to selection except the intercepts, β_0 and α_0.

Prior specification:

For the vector of non-zero regression coefficients β_γ we specify a g-prior

$$\beta_\gamma | c_\beta, \sigma^2, \gamma, \alpha, \delta \sim N(0, c_\beta \sigma^2 (\tilde{X}_\gamma^\top \tilde{X}_\gamma)^{-1}).$$

where $\tilde{X}$ is a scaled version of design matrix X of the mean model.

For the vector of non-zero regression coefficients α_δ we specify a normal prior

$$\alpha_\delta | c_\alpha, \delta \sim N(0, c_\alpha I).$$

Independent priors are specified for the indicators variables γ and δ as $P(\gamma = 1|\pi_\mu) = \pi_\mu$ and $P(\delta = 1|\pi_\sigma) = \pi_\sigma$. Further, Beta priors are specified for π_μ and π_σ

$$\pi_\mu \sim Beta(c_\mu, d_\mu), \pi_\sigma \sim Beta(c_\sigma, d_\sigma).$$

We note that blocks of regression coefficients associated with distinct covariate effects have their own probability of selection (π_μ or π_σ) and this probability has its own prior distribution.

Further, we specify inverse Gamma priors for c_β and c_α

$$c_\beta \sim IG(a_\beta, b_\beta), c_\alpha \sim IG(a_\alpha, b_\alpha)$$

For σ^2 we consider inverse Gamma and half-normal priors

$$\sigma^2 \sim IG(a_\sigma, b_\sigma), |\sigma| \sim N(0, \phi_\sigma^2).$$

Lastly, for the elements of the correlation matrix, we specify normal distributions with mean μ_R and variance σ_R^2, with the priors on these two parameters being normal and half-normal, respectively. This is the common correlations model. Further, the grouped correlations model can be specified. It considers a mixture of normal distributions for the means μ_R. The grouped correlations model can also be specified. It clusters the variables instead of the correlations.

Value

Function `mvrm` returns the following:

`call`	the matched call.
`formula`	model formula.
`seed`	the seed that was used (in case replication of the results is needed).
`data`	the dataset
`X`	the mean model design matrix.
`Z`	the variance model design matrix.
`LG`	the length of the vector of indicators γ.
`LD`	the length of the vector of indicators δ.
`mcpar`	the MCMC parameters: length of burn in period, total number of samples, thinning period.
`nSamples`	total number of posterior samples
`DIR`	the storage directory

Further, function `mvrm` creates files where the posterior samples are written. These files are (with all file names preceded by 'BNSP.'):

`alpha.txt`	contains samples from the posterior of vector α. Rows represent posterior samples and columns represent the regression coefficient, and they are in the same order as the columns of design matrix Z.
`beta.txt`	contains samples from the posterior of vector β. Rows represent posterior samples and columns represent the regression coefficients, and they are in the same order as the columns of design matrix X.
`gamma.txt`	contains samples from the posterior of the vector of the indicators γ. Rows represent posterior samples and columns represent the indicator variables, and they are in the same order as the columns of design matrix X.
`delta.txt`	contains samples from the posterior of the vector of the indicators δ. Rows represent posterior samples and columns represent the indicator variables, and they are in the same order as the columns of design matrix Z.
`sigma2.txt`	contains samples from the posterior of the error variance σ^2 of each response.
`cbeta.txt`	contains samples from the posterior of c_β.
`calpha.txt`	contains samples from the posterior of c_α.

`R.txt`	contains samples from the posterior of the correlation matrix R.
`theta.txt`	contains samples from the posterior of θ of the shadow prior (probably not needed).
`muR.txt`	contains samples from the posterior of μ_R.
`sigma2R.txt`	contains samples from the posterior of σ^2_R.
`deviance.txt`	contains the deviance, minus twice the log likelihood evaluated at the sampled values of the parameters.

In addition to the above, for models that cluster the correlations we have

`compAlloc.txt`	The cluster at which the correlations were allocated, λ_{kl}. These are integers from zero to the specified number of clusters minus one.
`nmembers.txt`	The numbers of correlations assigned to each cluster.
`DPconc.txt`	Contains samples from the posterior of the Dirichlet process concentration parameter.

In addition to the above, for models that cluster the variables we have

`compAllocV.txt`	The cluster at which the variables were allocated, λ_k. These are integers from zero to the specified number of clusters minus one.
`nmembersV.txt`	The numbers of variables assigned to each cluster.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

References

Papageorgiou, G. and Marshall, B.C. (2019). Bayesian semiparametric analysis of multivariate continuous responses, with variable selection. arXiv.

Papageorgiou, G. (2018). BNSP: an R Package for fitting Bayesian semiparametric regression models and variable selection. The R Journal, 10(2):526-548.

Chan, D., Kohn, R., Nott, D., & Kirby, C. (2006). Locally adaptive semiparametric estimation of the mean and variance functions in regression models. Journal of Computational and Graphical Statistics, 15(4), 915-936.

Examples

```
# Fit a mean/variance regression model on the cps71 dataset from package np.
#This is a univariate response model
require(np)
require(ggplot2)
data(cps71)
model <- logwage ~ sm(age, k = 30, bs = "rd") | sm(age, k = 30, bs = "rd")
DIR <- getwd()
## Not run: m1 <- mvrm(formula = model, data = cps71, sweeps = 10000, burn = 5000,
                                        thin = 2, seed = 1, StorageDir = DIR)
#Print information and summarize the model
print(m1)
```

```
summary(m1)
#Summarize and plot one parameter of interest
alpha <- mvrm2mcmc(m1, "alpha")
summary(alpha)
plot(alpha)
#Obtain a plot of a term in the mean model
wagePlotOptions <- list(geom_point(data = cps71, aes(x = age, y = logwage)))
plot(x = m1, model = "mean", term = "sm(age)", plotOptions = wagePlotOptions)
plot(m1)
#Obtain predictions for new values of the predictor "age"
predict(m1, data.frame(age = c(21, 65)), interval = "credible")

# Fit a bivariate mean/variance model on the marks dataset from package ggm
# two responses: marks mechanics and vectors, and one covariate: marks on algebra
model2 <- mechanics | vectors ~ sm(algebra, k = 5) | sm(algebra, k = 3)
m2 <- mvrm(formula = model2, data = marks, sweeps = 100000, burn = 50000,
                       thin = 2, seed = 1, StorageDir = DIR)
plot(m2)

## End(Not run)
```

`mvrm2mcmc` *Convert posterior samples from function* `mvrm` *into an object of class 'mcmc'*

Description

Reads in files where the posterior samples were written and creates an object of class 'mcmc' so that functions like `summary` and `plot` from package `coda` can be used

Usage

```
mvrm2mcmc(mvrmObj, labels)
```

Arguments

`mvrmObj`	An object of class 'mvrm' as created by a call to function `mvrm`.
`labels`	The labels of the files to be read in. These can be one or more of: "alpha", "beta", "gamma", "delta", "sigma2", "cbeta", "calpha", "R", "muR", "sigma2R", "nmembers", "nmembersV", "compAlloc", "compAllocV", and "DPconc" and they correspond to the parameters of the model that a call to functions `mvrm` fits. In addition, "deviance" can be read in. If left unspecified, all files are read in.

Value

An object of class 'mcmc' that holds the samples from the posterior of the selected parameter.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

mvrm

Examples

```
#see \code{mvrm} example
```

---

plot.mvrm *Creates plots of terms in the mean and/or variance models*

Description

This function plots estimated terms that appear in the mean and variance models.

Usage

```
## S3 method for class 'mvrm'
plot(x, model, term, response, response2, intercept = TRUE, grid = 30,
centre = mean, quantiles = c(0.1, 0.9), contour = TRUE, static = TRUE,
centreEffects = FALSE, plotOptions = list(), nrow, ask = FALSE,
plotEmptyCluster = FALSE, combine = FALSE, ...)
```

Arguments

x	an object of class 'mvrm' as generated by function mvrm.
model	one of "mean", "stdev", or "both", specifying which model to be visualized.
term	the term in the selected model to be plotted.
response	integer number denoting the response variable to be plotted (in case there is more than one).
response2	only relevant for multivariate longitudinal data.
intercept	specifies if an intercept should be included in the calculations.
grid	the length of the grid on which the term will be evaluated.
centre	a description of how the centre of the posterior should be measured. Usually mean or median.
quantiles	the quantiles to be used when plotting credible regions. Plots without credible intervals may be obtained by setting this argument to NULL.
contour	relevant for 3D plots only. If contour=TRUE then plot.mvrm creates contour plots. contour=FALSE is allowed only for creating one plot at a time. The plot can be static or dynamic. See argument 'static'.
static	relevant for 3D plots only. If static=TRUE then plot.mvrm calls function ribbon3D from package plot3D to create the plot. If static=FALSE then plot.mvrm calls function scatterplot3js from package threejs to create the plot.

`centreEffects` if TRUE then the effects in the mean functions are centred around zero over the range of the predictor while the effects in the variance function are scaled around one.

`plotOptions` for plots of univariate smooth terms or for plots of bivariate smooth terms where one of the two covariates is discrete, this is a list of plot elements to give to `ggplot`. For smooths of bivariate continuous covariates, this is a list of plot elements to give to `ribbon3D` (if `static=FALSE`) or to `scatterplot3js` (if `static=TRUE`).

`nrow` the number of rows in the figure with the plots.

`ask` if set to TRUE, plots will be displayed one at a time.

`plotEmptyCluster` if set to TRUE, plots of empty clusters will be displayed. Relevant for multivariate longitudinal datasets.

`combine` Binary indicator. It can be set to TRUE to simultaneously plot two terms. One of the terms must be continuous and the other must be discrete. This makes sense to set to TRUE when wanting to visualize groups that have a common slope.

`...` other arguments.

Details

Use this function to obtain predictions.

Value

Predictions along with credible/pediction intervals

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

`mvrm`

Examples

```
#see \code{mvrm} example
```

plotCorr *Creates plots of the correlation matrices*

Description

This function plots the posterior mean and credible intervals of the elements of correlation matrices.

Usage

```
plotCorr(x, term = "R", centre = mean, quantiles = c(0.1, 0.9), ...)
```

Arguments

x	an object of class 'mvrm' as generated by function mvrm.
term	R or muR,
centre	a description of how the centre of the posterior should be measured. Usually mean or median.
quantiles	the quantiles to be used when plotting credible regions. Plots without credible intervals may be obtained by setting this argument to NULL.
...	other arguments.

Details

Use this function to visualize the elements of a correlation matrix.

Value

Posterior means and credible intervals of elements of correlation matrices.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

mvrm

Examples

```
#see \code{mvrm} example
```

predict.mvrm *Model predictions*

Description

Provides predictions and posterior credible/prediction intervals for given feature vectors.

Usage

```
## S3 method for class 'mvrm'
predict(object, newdata, interval = c("none", "credible", "prediction"),
                        level = 0.95, ind.preds=FALSE, ...)
```

Arguments

object	an object of class "mvrm", usually a result of a call to mvrm.
newdata	data frame of feature vectors to obtain predictions for. If newdata is missing, the function will use the feature vectors in the data frame used to fit the mvrm object.
interval	type of interval calculation.
level	the level of the credible interval.
ind.preds	Binary indicator. If set to TRUE the function returns additionally the predictions per individual MCMC sample.
...	other arguments.

Details

The function returns predictions of new responses or the means of the responses for given feature vectors. Predictions for new responses or the means of new responses are the same. However, the two differ in the associated level of uncertainty: response predictions are associated with wider (prediction) intervals than mean response predictions. To obtain prediction intervals (for new responses) the function samples from the normal distributions with means and variances as sampled during the MCMC run.

Value

Predictions for given covariate/feature vectors.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

mvrm

Examples

```
#see \code{mvrm} example
```

`print.mvrm` *Prints an mvrm fit*

Description

Provides basic information from an mvrm fit.

Usage

```
## S3 method for class 'mvrm'
print(x, digits = 5, ...)
```

Arguments

`x`	an object of class "mvrm", usually a result of a call to `mvrm`.
`digits`	the number of significant digits to use when printing.
`...`	other arguments.

Details

The function prints information about mvrm fits.

Value

The function provides a matched call, the number of posterior samples obtained and marginal inclusion probabilities of the terms in the mean and variance models.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

`mvrm`

Examples

```
#see \code{mvrm} example
```

s *mgcv constructor* s

Description

Provides interface between mgcv::s and BNSP. `s(...)` calls `mgcv::smoothCon(mgcv::s(...),...`

Usage

```
s(..., data, knots = NULL, absorb.cons = FALSE, scale.penalty = TRUE,
n = nrow(data), dataX = NULL, null.space.penalty = FALSE, sparse.cons = 0,
diagonal.penalty = FALSE, apply.by = TRUE, modCon = 0, k = -1, fx = FALSE,
bs = "tp", m = NA, by = NA, xt = NULL, id = NULL, sp = NULL, pc = NULL)
```

Arguments

`...`	a list of variables. See mgcv::s
`data`	see mgcv::smoothCon
`knots`	see mgcv::knots
`absorb.cons`	see mgcv::smoothCon
`scale.penalty`	see mgcv::smoothCon
`n`	see mgcv::smoothCon
`dataX`	see mgcv::smoothCon
`null.space.penalty`	see mgcv::smoothCon
`sparse.cons`	see mgcv::smoothCon
`diagonal.penalty`	see mgcv::smoothCon
`apply.by`	see mgcv::smoothCon
`modCon`	see mgcv::smoothCon
`k`	see mgcv::s
`fx`	see mgcv::s
`bs`	see mgcv::s
`m`	see mgcv::s
`by`	see mgcv::s
`xt`	see mgcv::s
`id`	see mgcv::s
`sp`	see mgcv::s
`pc`	see mgcv::s

Details

The most relevant arguments for BNSP users are the list of variables `...`, `knots`, `absorb.cons`, `bs`, and `by`.

Value

A design matrix that specifies a smooth term in a model.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

---

simD *Simulated dataset*

Description

Just a simulated dataset to illustrate the DO mixture model. The success probability and the covariate have a non-linear relationship.

Usage

```
data(simD)
```

Format

A data frame with 300 independent observations. Three numerical vectors contain information on

`Y` number of successes.

`E` number of trials.

`X` explanatory variable.

---

simD2 *Simulated dataset*

Description

A simulated dataset to illustrate the multivariate longitudinal model. It consists of a bivariate vector of responses observed over 6 time points.

Usage

```
data(simD2)
```

Format

A data frame that includes observations on 50 sampling units. The data frame has 300 rows for the 50 sampling units observed over 6 time points. It has 4 columns

`Y1` first response.

`Y2` second response.

`time` the time of observation.

`id` unique sampling unit identifier.

`sinusoid` *Sinusoid terms in mvrm formulae*

Description

Function used to define sinusoidal curves in the mean formula of function `mvrm`. The function is used internally to construct design matrices.

Usage

```
sinusoid(..., harmonics = 1, amplitude = 1, period = 0, periodRange = NULL,
breaks = NULL, knots = NULL)
```

Arguments

`...`	a single covariate that the sinusoid term is a function of.
`harmonics`	an integer value that denotes the number of sins and cosines to be utilized in the representation of a sinusoidal curve.
`amplitude`	a positive integer. If set equal to one, it denotes a fixed amplitude. Otherwise, if set to an integer that is greater than one, it denotes the number of knots to be utilized in the representation of the time-varying amplitude.
`period`	the period of the sinusoidal wave. Values less than or equal to zero signify that the period is unknown. Positive values signify that the period is known and fixed.
`periodRange`	a vector of length two with the range of possible period values. It is required when the period is unknown.
`breaks`	the growth break points.
`knots`	the knots to be utilized in the representation of the time-varying amplitude. Relevant only when `amplitude` is greater than 1.

Details

Use this function within calls to function mvrm to specify sinusoidal waves in the mean function of a regression model.

Consider the sinusoidal curve

$$y_t = \beta_0 + A(t)\sin(2\pi t/\omega + \varphi) + \epsilon_t,$$

where y_t is the response at time t, β_0 is an intercept term, $A(t)$ is a time-varying amplitude, $\varphi \in [0, 2\pi]$ is the phase shift parameter, ω is the period taken to be known, and ϵ_t is the error term.

The period ω is defined by the argument period.

The time-varying amplitude is represented using $A(t) = \sum_{j=1}^{K} \beta_{Aj}\phi_{Aj}(t)$, where K, the number of knots, is defined by argument amplitude. If amplitude = 1, then the amplitude is taken to be fixed: $A(t) = A$.

Further, $\sin(2\pi t/\omega + \varphi)$ is represented utilizing $\sin(2\pi t/\omega + \varphi) = \sum_{k=1}^{L} a_k \sin(2k\pi t/\omega) + b_k \cos(2k\pi t/\omega)$, where L, the number of harmonics, is defined by argument harmonics.

Value

Specifies the design matrices of an mvrm call

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

mvrm

Examples

```
# Simulate and fit a sinusoidal curve
# First load releveant packages
require(BNSP)
require(ggplot2)
require(gridExtra)
require(Formula)
# Simulate the data
mu <- function(u) {cos(0.5 * u) * sin(2 * pi * u + 1)}
set.seed(1)
n <- 100
u <- sort(runif(n, min = 0, max = 2*pi))
y <- rnorm(n, mu(u), 0.1)
data <- data.frame(y, u)
# Define the model and call function \code{mvrm} that perfomes posterior sampling for the given
# dataset and defined model
model <- y ~ sinusoid(u, harmonics = 2, amplitude = 20, period = 1)
## Not run:
m1 <- mvrm(formula = model, data = data, sweeps = 10000, burn = 5000, thin = 2, seed = 1,
           StorageDir = getwd())
# Plot
```

```
x1 <- seq(min(u), max(u), length.out = 100)
plotOptionsM <- list(geom_line(aes_string(x = x1, y = mu(x1)), col = 2, alpha = 0.5, lty = 2),
                     geom_point(data = data, aes(x = u, y = y)))
plot(x = m1, term = 1, plotOptions = plotOptionsM, intercept = TRUE,
     quantiles = c(0.005, 0.995), grid = 100, combine = 1)

## End(Not run)
```

---

sm *Smooth terms in mvrm formulae*

Description

Function used to define smooth effects in the mean and variance formulae of function mvrm. The function is used internally to construct the design matrices.

Usage

```
sm(..., k = 10, knots = NULL, bs = "rd")
```

Arguments

...	one or two covariates that the smooth term is a function of. If two covariates are used, they may be both continuous or one continuous and one discrete. Discrete variables should be defined as factor in the data argument of the calling mvrm function.
k	the number of knots to be utilized in the basis function expansion.
knots	the knots to be utilized in the basis function expansion.
bs	a two letter character indicating the basis functions to be used. Currently, the options are "rd" that specifies radial basis functions and is available for univariate and bivariate smooths, and "pl" that specifies thin plate splines that are available for univariate smooths.

Details

Use this function within calls to function mvrm to specify smooth terms in the mean and/or variance function of the regression model.

Univariate radial basis functions with q basis functions or $q-1$ knots are defined by

$$\mathcal{B}_1 = \left\{\phi_1(u) = u, \phi_2(u) = ||u-\xi_1||^2 \log\left(||u-\xi_1||^2\right), \ldots, \phi_q(u) = ||u-\xi_{q-1}||^2 \log\left(||u-\xi_{q-1}||^2\right)\right\},$$

where $||u||$ denotes the Euclidean norm of u and $\xi_1, \ldots, \xi_{q-1}$ are the knots that are chosen as the quantiles of the observed values of explanatory variable u, with $\xi_1 = \min(u_i), \xi_{q-1} = \max(u_i)$ and the remaining knots chosen as equally spaced quantiles between ξ_1 and ξ_{q-1}.

Thin plate splines are defined by

$$\mathcal{B}_2 = \left\{\phi_1(u) = u, \phi_2(u) = (u-\xi_1)_+, \ldots, \phi_q(u) = (u-\xi_q)_+\right\},$$

where $(a)_+ = \max(a, 0)$.

Radial basis functions for bivariate smooths are defined by

$$\mathcal{B}_3 = \left\{u_1, u_2, \phi_3(u) = ||u - \xi_1||^2 \log\left(||u - \xi_1||^2\right), \ldots, \phi_q(u) = ||u - \xi_{q-1}||^2 \log\left(||u - \xi_{q-1}||^2\right)\right\}.$$

Value

Specifies the design matrices of an `mvrm` call

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

`mvrm`

Examples

```
#see \code{mvrm} example
```

`summary.mvrm` *Summary of an mvrm fit*

Description

Provides basic information from an mvrm fit.

Usage

```
## S3 method for class 'mvrm'
summary(object, nModels = 5, digits = 5, printTuning = FALSE, ...)
```

Arguments

`object`	an object of class "mvrm", usually a result of a call to `mvrm`.
`nModels`	integer number of models with the highest posterior probability to be displayed.
`digits`	the number of significant digits to use when printing.
`printTuning`	if set to TRUE, the starting and finishig values of the tuninf parameters are displayed.
`...`	other arguments.

Details

Use this function to summarize mvrm fits.

Value

The functions provides a detailed description of the specified model and priors. In addition, the function provides information about the Markov chain ran (length, burn-in, thinning) and the folder where the files with posterior samples are stored. Lastly, the function provides the mean posterior and null deviance and the mean/variance models visited most often during posterior sampling.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

mvrm

Examples

```
#see \code{mvrm} example
```

---

te *mgcv constructor* te

Description

Provides interface between mgcv::te and BNSP. `te(...)` calls `mgcv::smoothCon(mgcv::te(...),...`

Usage

```
te(..., data, knots = NULL, absorb.cons = FALSE, scale.penalty = TRUE,
n = nrow(data), dataX = NULL, null.space.penalty = FALSE, sparse.cons = 0,
diagonal.penalty = FALSE, apply.by = TRUE, modCon = 0, k = NA, bs = "cr",
m = NA, d = NA, by = NA, fx = FALSE, np = TRUE, xt = NULL, id = NULL,
sp = NULL, pc = NULL)
```

Arguments

`...`	a list of variables. See mgcv::te
`data`	see mgcv::smoothCon
`knots`	see mgcv::knots
`absorb.cons`	see mgcv::smoothCon
`scale.penalty`	see mgcv::smoothCon
`n`	see mgcv::smoothCon
`dataX`	see mgcv::smoothCon
`null.space.penalty`	see mgcv::smoothCon
`sparse.cons`	see mgcv::smoothCon

diagonal.penalty	see mgcv::smoothCon
apply.by	see mgcv::smoothCon
modCon	see mgcv::smoothCon
k	see mgcv::te
bs	see mgcv::te
m	see mgcv::te
d	see mgcv::te
by	see mgcv::te
fx	see mgcv::te
np	see mgcv::te
xt	see mgcv::te
id	see mgcv::te
sp	see mgcv::te
pc	see mgcv::te

Details

The most relevant arguments for BNSP users are the list of variables `...`, `knots`, `absorb.cons`, `bs`, and `by`.

Value

A design matrix that specifies a smooth term in a model.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

---

ti *mgcv constructor* ti

Description

Provides interface between mgcv::ti and BNSP. `ti(...)` calls `mgcv::smoothCon(mgcv::ti(...),...`

Usage

```
ti(..., data, knots = NULL, absorb.cons = FALSE, scale.penalty = TRUE,
n = nrow(data), dataX = NULL, null.space.penalty = FALSE, sparse.cons = 0,
diagonal.penalty = FALSE, apply.by = TRUE, modCon = 0, k = NA, bs = "cr",
m = NA, d = NA, by = NA, fx = FALSE, np = TRUE, xt = NULL, id = NULL,
sp = NULL, mc = NULL, pc = NULL)
```

Arguments

...	a list of variables. See mgcv::ti
data	see mgcv::smoothCon
knots	see mgcv::knots
absorb.cons	see mgcv::smoothCon
scale.penalty	see mgcv::smoothCon
n	see mgcv::smoothCon
dataX	see mgcv::smoothCon
null.space.penalty	see mgcv::smoothCon
sparse.cons	see mgcv::smoothCon
diagonal.penalty	see mgcv::smoothCon
apply.by	see mgcv::smoothCon
modCon	see mgcv::smoothCon
k	see mgcv::ti
bs	see mgcv::ti
m	see mgcv::ti
d	see mgcv::ti
by	see mgcv::ti
fx	see mgcv::ti
np	see mgcv::ti
xt	see mgcv::ti
id	see mgcv::ti
sp	see mgcv::ti
mc	see mgcv::ti
pc	see mgcv::ti

Details

The most relevant arguments for BNSP users are the list of variables ..., knots, absorb.cons, bs, and by.

Value

A design matrix that specifies a smooth term in a model.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

Index

∗ **cluster**
  dpmj, 7
∗ **datasets**
  ami, 3
  simD, 32
  simD2, 32
∗ **models**
  sinusoid, 33
  sm, 35
∗ **nonparametric**
  dpmj, 7
  lmrm, 15
  mvrm, 19
∗ **regression**
  lmrm, 15
  mvrm, 19
  sinusoid, 33
  sm, 35
∗ **smooth**
  lmrm, 15
  mvrm, 19
  sinusoid, 33
  sm, 35

ami, 3

BNSP *(*BNSP-package*)*, 2
BNSP-package, 2

chol, 4, *5*
clustering, 5
continue, 6

dpmj, 7

histCorr, 14

lmrm, 15

mvrm, *6, 7*, *14*, 19, *26–30*, *34*, *36, 37*
mvrm2mcmc, 25

plot.mvrm, 26
plotCorr, 28
predict.mvrm, 29
print.mvrm, 30

s, 31
simD, 32
simD2, 32
sinusoid, 33
sm, 35
summary.mvrm, 36

te, 37
ti, 38