Package ‘BNSP’

September 27, 2018

Title Bayesian Non- And Semi-Parametric Model Fitting

Version 2.0.8

Date 2018-09-27

Author Georgios Papageorgiou

Maintainer Georgios Papageorgiou <gpapageo@gmail.com>

Description Markov chain Monte Carlo algorithms for non- and semi-parametric models: 1. Dirichlet process mixtures & 2. spike-slab for variable selection in mean/variance regression models.

Depends R (>= 3.1.0)

Imports coda, ggplot2, plot3D, threejs, gridExtra, cubature, Formula, plyr,mgcv

LinkingTo cubature

Suggests mvtnorm, np

License GPL (>= 2)

URL http://www.bbk.ac.uk/ems/faculty/papageorgiou/BNSP

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-09-27 09:10:03 UTC

R topics documented:

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

Description

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

Details

Package:	BNSP
Type:	Package
Version:	2.0.7
Date:	2018-09-27
License:	GPL (>=2)

Acknowledgments

This work was partly supported by the Medical Research Council grant number G09018401.

Author(s)

Georgios Papageorgiou (2014)

Maintainer: Georgios Papageorgiou <gpapageo@gmail.com>

References

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

Papageorgiou, G. (2018). BNSP: an R Package for Fitting Bayesian Semiparametric Regression Models and Variable Selection. arXiv:1804.10939 [stat.OT]

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.

`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: $\xi \sim$ Gamma(α_ξ, β_ξ).
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: $\xi_1 \sim$ 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 $\alpha_{1\xi}$ 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$(\alpha_{\xi_2}, \beta_{\xi_2})I[\xi_2 > 0.05]$	Alpha.xi = c(1.0,1.0), Beta.xi = c(0.1,1.0)
	where β_{ξ_2} denotes st.dev.	
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 ξ_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)
```

`mvrm` *Bayesian models for normally distributed responses and semiparametric mean and variance regression models*

Description

MCMC for normally distributed responses with additive model for the mean and variance functions achieved via spike-slab prior for variable selection. See 'Details' section for a full description of the model.

Usage

```
mvrm(formula,data,sweeps,burn=0,thin=1,seed,StorageDir,
c.betaPrior="IG(0.5,0.5*n)",c.alphaPrior="IG(1.1,1.1)",
pi.muPrior="Beta(1,1)",pi.sigmaPrior="Beta(1,1)",sigmaPrior="HN(2)",...)
```

Arguments

`formula`	a formula defining the response and the covariates in the mean and variance models e.g. `y ~ x	z` or for smooth effects `y ~ sm(x)	sm(z)`. The package uses the extended formula notation, where on the right of ~ we define two models: on the right of `|` is the mean model and on the left is the variance model.
`data`	data frame.
`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 parameters of the inverse Gamma prior of c_β. The default is "IG(0.5,0.5*n)", that is, an inverse Gamma with parameters $1/2$ and $n/2$, where n is the sample size.
`c.alphaPrior`	The parameters of the inverse Gamma prior of c_α. The default is "IG(1.1,1.1)".
`pi.muPrior`	The parameters of the Beta prior of π_μ. The default is "Beta(1,1)".
`pi.sigmaPrior`	The parameters of the Beta prior of π_σ. The default is "Beta(1,1)".
`sigmaPrior`	The prior of σ^2. The default is "HN(2)", a half-normal with variance equal to two, that is $\|\sigma\| \sim N(0,2)$. Inverse Gamma prior is also available.
`...`	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 responses and mean and variance functions modeled in terms of covariates. For instance, in the presence of 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 the effects of u_2 and w_2. Smooth functions, such as f_μ and f_σ, are represented by basis function expansion. For instance

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

where ϕ are the basis functions and β are the associated 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

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

Lastly, for σ^2 we consider inverse Gamma and half-normal priors

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

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.
`cbeta.txt`	contains samples from the posterior of c_β.
`calpha.txt`	contains samples from the posterior of c_α.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

References

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.

Papageorgiou, G. (2018). BNSP: an R Package for fitting Bayesian regression models with semiparametric mean and variance functions. arXiv:1804.10939 [stat.OT]

Examples

```
# Fit a mean/variance regression model on the cps71 dataset from package np
require(np)
require(ggplot2)
data(cps71)
model <- logwage ~ sm(age,k=30,bs="rd") | sm(age,k=30,bs="rd")
DIR<-getwd()
m1<-mvrm(formula=model,data=cps71,sweeps=10,burn=5,thin=1,seed=1,StorageDir=DIR)
m2 <- mvrm(formula=model,data=cps71,sweeps=10000,burn=5000,thin=2, seed=1,StorageDir=DIR)
#Print information and summarize the model
print(m2)
summary(m2)
#Summarize and plot one parameter of interest
alpha<-mvrm2mcmc(m2,"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=m2,model="mean",term="sm(age)",plotOptions=wagePlotOptions)
#Obtain predictions for new values of the predictor "age"
predict(m2,data.frame(age=c(21,65)),interval="credible")
```

`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", and they correspond to the parameters of the model that a call to functions `mvrm` fits.

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="mean",term,intercept=TRUE,grid=30,centre=mean,
quantiles=c(0.1, 0.9),static=TRUE,centreEffects=FALSE,plotOptions=list(), ...)
```

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.
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.
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).
...	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
```

---

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, nSamples = 100, ...)
```

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	tolerance level.
nSamples	number of samples to obtain from the posterior predictive distribution (for each sweep of the MCMC). Only relevant for "prediction intervals".
...	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 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.

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

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

where $||u||$ denotes the Euclidean norm of u and $\xi_1, \dots, \xi_{q-1}$ 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

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

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

Radial basis functions for bivariate smooths are defined by

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

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, ...)
```

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.
`...`	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