Package: DDPstar (via r-universe)

October 10, 2026

Type Package

Title Density Regression via Dirichlet Process Mixtures of Normal Structured Additive Regression Models

Version 1.0-1

Date 2025-01-30

Imports stats, grDevices, graphics, splines, moments, Matrix, parallel, MASS

Depends R (>= 3.5.0)

Description Implements a flexible, versatile, and computationally tractable model for density regression based on a single-weights dependent Dirichlet process mixture of normal distributions model for univariate continuous responses. The model assumes an additive structure for the mean of each mixture component and the effects of continuous covariates are captured through smooth nonlinear functions. The key components of our modelling approach are penalised B-splines and their bivariate tensor product extension. The proposed method can also easily deal with parametric effects of categorical covariates, linear effects of continuous covariates, interactions between categorical and/or continuous covariates, varying coefficient terms, and random effects. Please see Rodriguez-Alvarez, Inacio et al. (2025) for more details.

License GPL

NeedsCompilation no

Author Maria Xose Rodriguez-Alvarez [aut, cre] (<https://orcid.org/0000-0002-1329-9238>), Vanda Inacio [aut] (<https://orcid.org/0000-0001-8084-1616>)

Maintainer Maria Xose Rodriguez-Alvarez <mxrodriguez@uvigo.gal>

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

Date/Publication 2025-01-31 11:10:09 UTC

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

RemoteRef HEAD

RemoteSha 487196244e9cd148ec6d644c0f72eb3b4611021b

Contents

DDPstar-package . . . 2
dde . . . 3
DDPstar . . . 4
f . . . 8
mcmccontrol . . . 10
predict.DDPstar . . . 11
predictive.checks.DDPstar . . . 15
print.DDPstar . . . 17
priorcontrol . . . 18
quantileResiduals . . . 20
rae . . . 21
summary.DDPstar . . . 23

Index **24**

`DDPstar-package` *Density Regression via Dirichlet Process Mixtures of Normal Structured Additive Regression Models*

Description

Implements a flexible, versatile, and computationally tractable model for density regression based on a single-weights dependent Dirichlet process mixture of normal distributions model for univariate continuous responses. The model assumes an additive structure for the mean of each mixture component and the effects of continuous covariates are captured through smooth nonlinear functions. The key components of our modelling approach are penalised B-splines and their bivariate tensor product extension. The proposed method can also easily deal with parametric effects of categorical covariates, linear effects of continuous covariates, interactions between categorical and/or continuous covariates, varying coefficient terms, and random effects. Please see Rodriguez-Alvarez, Inacio et al. (2025) for more details.

Details

Index of help topics:

```
DDPstar                 Density Regression via Dirichlet Process
                        Mixtures (DDP) of Normal Structured Additive
                        Regression (STAR) Models
DDPstar-package         Density Regression via Dirichlet Process
                        Mixtures of Normal Structured Additive
                        Regression Models
dde                     Dichlorodiphenyldichloroethylene (DDE) and
                        preterm delivery data
f                       Defining smooth terms in DDPstar formulae
mcmccontrol             Markov chain Monte Carlo (MCMC) parameters
predict.DDPstar         Predictions from fitted DDPstar models
```

```
predictive.checks.DDPstar
                        Posterior predictive checks.
print.DDPstar           Print method for DDPstar objects
priorcontrol            Prior information for the DDPstar model
quantileResiduals       Quantile residuals.
rae                     Defining random effects in DDPstar formulae
summary.DDPstar         Summary method for DDPstar objects
```

Author(s)

Maria Xose Rodriguez-Alvarez [aut, cre] (<https://orcid.org/0000-0002-1329-9238>), Vanda Inacio [aut] (<https://orcid.org/0000-0001-8084-1616>)

Maintainer: Maria Xose Rodriguez-Alvarez <mxrodriguez@uvigo.gal>

References

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. Statistical Science, **11**(2), 89-121.

Eilers, P.H.C. and Marx, B.D. (2003). Multidimensional calibration with temperature interaction using two- dimensional penalized signal regression. Chemometrics and Intelligence Laboratory Systems, **66**, 159-174.

De Iorio, M., Muller, P., Rosner, G. L., and MacEachern, S. N. (2004). An anova model for dependent random measures. Journal of the American Statistical Association, **99**(465), 205-215

De Iorio, M., Johnson, W. O., Muller, P., and Rosner, G. L. (2009). Bayesian nonparametric nonproportional hazards survival modeling. Biometrics, **65**, 762–775.

Lang, S. and Brezger, A. (2004). Bayesian P-splines. Journal of Computational and Graphical Statistics, **13**(1), 183-212.

Lee, D.-J., Durban, M., and Eilers, P. (2013). Efficient two-dimensional smoothing with P- spline ANOVA mixed models and nested bases. Computational Statistics & Data Analysis, **61**, 22-37.

Rodriguez-Alvarez, M. X, Inacio, V. and Klein N. (2025). Density regression via Dirichlet process mixtures of normal structured additive regression models. Accepted for publication in Statistics and Computing (DOI: 10.1007/s11222-025-10567-0).

---

dde *Dichlorodiphenyldichloroethylene (DDE) and preterm delivery data*

Description

Data set associated with a study aimed to evaluate the association between maternal serum concentration of the DDT metabolite DDE and preterm delivery. For details, see Longnecker et al. (2001).

Usage

```
data("dde")
```

Format

A data frame with 2312 observations on the following 2 variables.

`DDE` the level of the Dichlorodiphenyldichloroethylene (DDE)

`GAD` the gestational age at delivery (GAD), in days.

Source

The dataset can be downloaded from https://github.com/tommasorigon/LSBP/blob/master/Tutorial/dde.RData.

References

Longnecker, M. P., Klebanoff, M. A., Zhou, H., and Brock, J. W. (2001). Association between maternal serum concentration of the DDT metabolite DDE and preterm and small-for-gestational-age babies at birth. The Lancet, 358(9276):110-114.

Examples

```
library(DDPstar)
data(dde)
summary(dde)
```

`DDPstar` *Density Regression via Dirichlet Process Mixtures (DDP) of Normal Structured Additive Regression (STAR) Models*

Description

This function estimates the DDPstar model proposed by Rodriguez-Alvarez, Inacio, et al. (2025).

Usage

```
DDPstar(formula, data, standardise = TRUE, compute.lpml = FALSE, compute.WAIC = FALSE,
  compute.DIC = FALSE, prior = priorcontrol(), mcmc = mcmccontrol(), verbose = FALSE)
```

Arguments

`formula`	A `formula` object specifying the regression function associated to each component of the DDPstar model used to estimate the conditional density function of the response variable. Regarding the modelling of continuous covariates, both linear and nonlinear effects are allowed, with nonlinear effects being modelled through P-splines (see Note)
`data`	A data frame representing the data and containing all needed variables.
`standardise`	A logical value. If TRUE both the responses and the continuous covariates are standardised (i.e., the resulting variables have mean zero and standard deviation of one). The default is TRUE.

`compute.lpml`	A logical value. If TRUE, the log pseudo marginal likelihood (LPML, Geisser and Eddy, 1979) and the conditional predictive ordinates (CPO) are computed.
`compute.WAIC`	A logical value. If TRUE, the widely applicable information criterion (WAIC, Gelman et al., 2014; Watanabe, 2010) is computed.
`compute.DIC`	A logical value. If TRUE, the deviance information criterion (DIC)(Celeux et al., 2006, Spiegelhalter et al., 2002) is computed.
`prior`	Hyparameter specification. A list of control values to replace the default values returned by the function `priorcontrol`. See `priorcontrol` for details.
`mcmc`	A list of control values to replace the default values returned by the function `mcmccontrol`. See `mcmccontrol` for details.
`verbose`	A logical value indicating whether additional information should be displayed while the MCMC algorithm is running. . The default is FALSE

Details

Our DDPstar model for the conditional density takes the following form

$$p(y|\boldsymbol{x}) = \sum_{l=1}^{L} \omega_l \phi(y \mid \mu_l(\boldsymbol{x}), \sigma_l^2),$$

where $\phi(y|\mu, \sigma^2)$ denotes the density function of the normal distribution, evaluated at y, with mean μ and variance σ^2. The regression function $\mu_l(\boldsymbol{x})$ can incorporate both linear and nonlinear (through P-splines) effects of continuous covariates, random effects, categorical covariates (factors) as well as interactions. Interactions between categorical and (nonlinear) continuous covariates are also allowed (factor-by curve interactions).

The mean (regression function) of each mixture component, $\mu_l(\cdot)$, is compactly specified (see Rodriguez-Alvarez, Inacio, et al. (2025) for details) as

$$\mu_l(\boldsymbol{x}) = \underbrace{\mathbf{x}^\top \boldsymbol{\beta}_l}_{\text{Parametric}} + \underbrace{\sum_{r=1}^{R} \mathbf{z}_r(\boldsymbol{v}_r)^\top \boldsymbol{\gamma}_{lr}}_{\text{Smooth/Random}}$$

where $\mathbf{x}^\top \boldsymbol{\beta}_l$ contains all parametric effects, including, by default, the intercept, parametric effects of categorical covariates and linear or parametric effects of continuous covariates and their interactions, as well as, the unpenalised terms associated with the P-splines formulation, and $\sum_{r=1}^{R} \mathbf{z}_r(\boldsymbol{v}_r)^\top \boldsymbol{\gamma}_{lr}$ contains the P-splines' penalised terms, as well as, the random effects.

In our DDPstar model, L is a pre-specified upper bound on the number of mixture components. The ω_l's result from a truncated version of the stick-breaking construction ($\omega_1 = \eta_1; \omega_l = \eta_l \prod_{r<l}(1 - \eta r)$, $l = 2, \ldots, L$; $\eta_1, \ldots, \eta_{L-1} \sim$ Beta $(1, \alpha)$; $\eta_L = 1$). Further, for $\boldsymbol{\beta}_l$ we consider

$$p\left(\boldsymbol{\beta}_l \mid \mathbf{m}, \mathbf{S}\right) \propto \exp\left(-\frac{1}{2}\left(\boldsymbol{\beta}_l - \mathbf{m}\right)^\top \mathbf{S}^{-1}\left(\boldsymbol{\beta}_l - \mathbf{m}\right)\right),$$

with conjugate hyperpriors $\mathbf{m} \sim N_Q(\mathbf{m}_0, \mathbf{S}_0)$ and $\mathbf{S}^{-1} \sim W_Q(\nu, (\nu\Psi)^{-1})$, where $W(\nu, (\nu\Psi)^{-1})$ denotes a Wishart distribution with ν degrees of freedom and expectation Ψ^{-1}, and Q denotes the

length of vector $\boldsymbol{\beta}_l$. Regarding the prior distributions for the P-splines' penalised coefficients and random effect coefficients we have

$$p(\boldsymbol{\gamma}_{lr} \mid \tau_{lr}^2) \propto \exp\left(-\frac{1}{2\tau_{lr}^2}\boldsymbol{\gamma}_{lr}^\top \mathbf{K}_r \boldsymbol{\gamma}_{lr}\right),$$

where the specific forms of $\mathbf{K}_r$ depend on the components type (see Rodriguez-Alvarez, Inacio, et al. (2025) for details). Finally, $\sigma_l^{-2} \sim \Gamma(a_{\sigma^2}, b_{\sigma^2})$, $\tau_{lr}^{-2} \sim \Gamma(a_{\tau^2}, b_{\tau^2})$ and $\alpha \sim \Gamma(a_\alpha, b_\alpha)$, where $\Gamma(a, b)$ denotes a Gamma distribution with shape parameter a and rate parameter b. For a detailed description, we refer to Rodriguez-Alvarez, Inacio, et al. (2025).

It is worth referring that with respect to the computation of the DIC, when $L = 1$, it is computed as in Spiegelhalter et al. (2002), and when $L > 1$, DIC3 as described in Celeux et al. (2006) is computed. Also, for the computation of the conditional predictive ordinates (CPO) we follow the stable version proposed by Gelman et al. (2014).

Value

An object of class `DDPstar`. It is a list containing the following objects:

`call`	The matched call.
`data`	The original supplied data argument.
`data_model`	A list with the data used in the fit: response variable and design matrix.
`missing.ind`	A logical value indicating whether for each pair of observations (responses and covariates) missing values occur.
`mcmc`	A list of control values for the Markov chain Monte Carlo (MCMC)
`fit`	Named list with the following information: (1) `formula`: the value of the argument `formula` used in the call. (2) `mm`: information needed to construct the model matrix associated with the DDPstar model. (3) `beta`: array of dimension `nsave`×`L`×`K` with the sampled regression coefficients. (4) `sd`: matrix of dimension `nsave`×`L` with the sampled variances. (5) `probs`: matrix of dimension `nsave`×`L` with the sampled components' weights. (6) `tau_sm`: matrix of dimension `nsave`×`L`×`R` with the sampled variances for smooth/random terms. (7)`S`: matrix of dimension `n`×`nsave` with the sampled allocations. (8) `alpha`: vector of length `nsave` with the sampled precision parameter of the Dirichlet Process. Here, `nsave` is the number of Gibbs sampler iterations saved, `L` is the maximum number of mixture components, `K` is the dimension of the coefficient vector (including both parametric and nonlinear/smooth/random effects), `R` is the number of nonlinear/smooth/random components, and `n` is the number of observations.
`lpml`	If computed, a list with two components: the log pseudo marginal likelihood (LPML) and the conditional predictive ordinates (CPO).
`WAIC`	If computed, a list with two components: widely applicable information criterion (WAIC) and associated complexity penalty (pW).
`DIC`	If computed, a list with two components: deviance information criterion (DIC) and associated complexity penalty (pD).

Note

The input argument `formula` is similar to that used for the `glm` function, except that flexible specifications can be added by means of function `f`. For instance, specification `y ~ x1 + f(x2)` would assume a linear effect of `x1` (if `x1` is continuous) and the effect of `x2` (assumed continuous)) would be modeled using Bayesian P-splines. Categorical variables (factors) can be also incorporated, as well as random effects (see `rae`) and interaction terms. For example, to include the factor-by-curve interaction between `x4` (continuous covariate) and `x3` (categorical covariate) we need to specify `y ~ x3 + f(x4, by = x3)`.

References

Celeux, G., Forbes, F., Robert C. P., and Titerrington, D. M. (2006). Deviance information criteria for missing data models. Bayesian Analysis, **1**, 651-674.

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. Statistical Science, **11**(2), 89-121.

Eilers, P.H.C. and Marx, B.D. (2003). Multidimensional calibration with temperature interaction using two- dimensional penalized signal regression. Chemometrics and Intelligence Laboratory Systems, **66**, 159-174.

De Iorio, M., Muller, P., Rosner, G. L., and MacEachern, S. N. (2004). An anova model for dependent random measures. Journal of the American Statistical Association, **99**(465), 205-215

De Iorio, M., Johnson, W. O., Muller, P., and Rosner, G. L. (2009). Bayesian nonparametric nonproportional hazards survival modeling. Biometrics, **65**, 762–775.

Geisser, S. and Eddy, W.F. (1979) A Predictive Approach to Model Selection, Journal of the American Statistical Association, **74**, 153-160.

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., and Rubin, D.B. (2014). Bayesian Data Analysis, 3rd ed. CRC Press: Boca Raton, FL.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. Statistics and Computing, **24**, 997-1010.

Lang, S. and Brezger, A. (2004). Bayesian P-splines. Journal of Computational and Graphical Statistics, **13**(1), 183-212.

Lee, D.-J., Durban, M., and Eilers, P. (2013). Efficient two-dimensional smoothing with P- spline ANOVA mixed models and nested bases. Computational Statistics & Data Analysis, **61**, 22-37.

Rodriguez-Alvarez, M. X, Inacio, V. and Klein N. (2025). Density regression via Dirichlet process mixtures of normal structured additive regression models. Accepted for publication in Statistics and Computing (DOI: 10.1007/s11222-025-10567-0).

Speigelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). Bayesian measures of model comparison and fit. Journal of the Royal Statistical Society, Ser. B, **64**, 583-639.

Watanabe, S. (2010). Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. Journal of Machine Learning Research, **11**, 3571-3594.

See Also

`f`, `rae`, `predict.DDPstar`

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

formula.dde <- GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005)
mcmc <- mcmccontrol(nburn = 20000, nsave = 15000, nskip = 1)
prior <- priorcontrol(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20)

set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = formula.dde,
  data = dde, mcmc = mcmc, prior = prior,
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE, compute.DIC = TRUE)
summary(fit_dde)
```

---

f *Defining smooth terms in DDPstar formulae*

Description

Auxiliary function used to define smooth terms using Bayesian P-splines within DDPstar model formulae. The function does not evaluate the smooth - it exists purely to help set up a model using P-spline based smoothers.

Usage

```
f(..., by = NULL, nseg = 5, bdeg = 3, pord = 2, atau = 1, btau = 0.005,
  prior.2D = c("anisotropic", "isotropic"))
```

Arguments

...	a list of up to two covariates to construct the smooth term.
by	a numeric or factor variable of the same dimension as each covariate. Following package mgcv, in the numeric case the elements multiply the smooth, evaluated at the corresponding covariate values (a 'varying coefficient model'). Here, the smooth function does not exclude the intercept, so the by variable should not be added as an additional main effect. In the factor case, a replicate of the smooth is produced for each factor level. In this case, these smooths will exclude the intercept, so the factor needs to be added as a main effect as well (see details).
nseg	the number of segments for the (marginal) B-spline bases used to represent the smooth term. Numerical vector of length equal to the number of covariates. Atomic values are also valid, being recycled. The default value is 5.
bdeg	the order of the polynomial for the (marginal) B-spline bases for this term. Numerical vector of length equal to the number of covariates. Atomic values are also valid, being recycled. The default value is 3 (cubic B-splines).

pord — penalty order. Numerical vector of length equal to the number of covariates. Atomic values are also valid, being recycled. The default value is 2 (second-order penalty).

atau — a numeric value. Hyperparameter; shape parameter of the gamma prior distribution for the precisions (inverse variances) of the smooth term. The default is 1.

btau — a numeric value. Hyperparameter; rate parameter of the gamma prior distribution for the precisions (inverse variances) of the smooth term. The default is 0.005.

prior.2D — character string. Indicates whether the same precision parameter should be used for both variables in two-dimensional smooth functions (isotropy), or if each variable should have a different precision parameter (anisotropy). In the latter case, the Bayesian extension of the PS-ANOVA model by Lee et al. (2013) is used. The default is "anisotropic".

Details

The functions `f()` is designed to represent either one dimensional smooth functions for main effects of continuous explanatory variables or two dimensional smooth functions representing two way interactions of continuous variables using Bayesian P-splines. By default, the values of the arguments `nseg`, `pord` and `bdeg` are repeated to the length of the explanatory covariates. The two dimensional smooth terms are constructed using the tensor-product of marginal (one-dimensional) B-spline bases (Eilers and Marx, 2003). In this case, the Bayesian extension of the PS-ANOVA model by Lee et al. (2013) is considered when `prior.2D = "anisotropic"`. It is also possible to include factor-by-curve interactions between a continuous covariate (e.g., `x1`) and a categorical covariate (e.g., `x2`) by means of argument `by`: `y ~ x2 + f(x1, by = x2)`.

Value

The function is interpreted in the formula of a `DDPstar` model and creates the right framework for fitting the smoother. List containing (among others) the following elements:

vars — names of the covariates involved in the smooth term.

nseg — the number of segments for the (marginal) B-spline basis for each covariate (numerical vector of length equal to the number of covariates).

pord — the penalty order (numerical vector of length equal to the number of covariates).

bdeg — the order of the polynomial for the (marginal) B-Spline bases for this term (numerical vector of length equal to the number of covariates)

atau — shape parameter of the gamma prior distribution

btau — rate parameter of the gamma prior distribution

References

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. Statistical Science, **11**(2), 89-121.

Eilers, P.H.C. and Marx, B.D. (2003). Multidimensional calibration with temperature interaction using two- dimensional penalized signal regression. Chemometrics and Intelligence Laboratory Systems, **66**, 159-174.

Lang, S. and Brezger, A. (2004). Bayesian P-splines. Journal of Computational and Graphical Statistics, **13**(1), 183-212.

Lee, D.-J., Durban, M., and Eilers, P. (2013). Efficient two-dimensional smoothing with P- spline ANOVA mixed models and nested bases. Computational Statistics & Data Analysis, **61**, 22-37.

See Also

rae, DDPstar

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

formula.dde <- GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005)

set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = formula.dde,
  data = dde, mcmc = list(nburn = 20000, nsave = 15000, nskip = 1),
  prior = list(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20),
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE, compute.DIC = TRUE)
summary(fit_dde)
```

---

mcmccontrol *Markov chain Monte Carlo (MCMC) parameters*

Description

This function is used to set various parameters controlling the Markov chain Monte Carlo (MCMC) parameters.

Usage

```
mcmccontrol(nsave = 8000, nburn = 2000, nskip = 1)
```

Arguments

nsave	An integer giving the total number of scans to be saved (does not include the burn-in and thinning iterations).
nburn	An integer giving the number of burn-in scans.
nskip	An integer giving the thinning interval,

Details

The value returned by this function is used as a control argument of the DDPstar function.

Value

A list with components for each of the possible arguments.

See Also

DDPstar

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

mcmc <- mcmccontrol(nburn = 20000, nsave = 15000, nskip = 1)

set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005),
  data = dde, mcmc = mcmc, prior = list(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20),
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE, compute.DIC = TRUE)
summary(fit_dde)
```

---

predict.DDPstar *Predictions from fitted DDPstar models*

Description

Takes a fitted DDPstar object produced by DDPstar() and produces predictions for different functionals given a new set of values for the model covariates.

Usage

```
## S3 method for class 'DDPstar'
predict(object, what = NULL, newdata, reg.select = NULL, den.grid = NULL,
  quant.probs = NULL, q.value = NULL,
  parallel = c("no", "multicore", "snow"), ncpus = 1, cl = NULL, ...)
```

Arguments

object	An object of class DDPstar as produced by DDPstar().
what	Character vector with the functionals to be obtained: "regfun" (regression function); "varfun" (variance function); "quantfun" (condional quantiles); "denfun" (conditional densities); ; "probfun" (conditional probabilities, i.e., probabilities of the form $P(Y \leq y^* \mid X = x)$).

newdata	Data frame containing the values of the covariates at which predictions will be computed
reg.select	Numeric value containing the position of the covariate effect in the DDPstar formula for which predictions are required. If NULL (and "regfun" is in input argument what), the regression function (coditional mean) is computed.
den.grid	Numeric vector with the response variable values at which the conditional densities should be computed. Applicable and required if "denfun" is in argument what. If required and NULL, a sequence of 100 values between the minumum and the maximum of the observed responses is considered.
quant.probs	Numeric vector with the quantiles at which to obtain/predict the conditional quantiles. Applicable and required if "quantfun" is in argument what. If required and NULL, the deciles are computed.
q.value	Numeric vector with the response variable values at which to obtain/predict the conditional probabilities. Applicable and required if "probfun" is in argument what.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.
...	Further arguments passed to or from other methods. Not yet implemented.

Value

A list with the following components (npred is the number of columns in newdata and nsave is the number of Gibbs sampler iterations saved)

regfun	Present if "regfun" is in input argument what. A matrix of dimension npred×nsave with the MCMC realisations of the regression function (conditional mean) or the desired component (if reg.select is used).
reg.select	Argument reg.select used in the call. Present if "regfun" is in input argument what.
varfun	Present if "varfun" is in input argument what. A matrix of dimension npred×nsave with the MCMC realisations of the conditional variance.
quantfun	Present if "quantfun" is in input argument what. An array of dimension npred×nsave×nq with the MCMC realisations of the conditional quantiles. Here nq is the length of argument quant.probs.
quant.probs	Argument quant.probs used in the call. Present if "quantfun" is in input argument what.
denfun	Present if "denfun" is in input argument what. An array of dimension nden×npred×nsave with the MCMC realisations of the conditional densities. Here nden is the length of argument den.grid.

den.grid	Argument den.grid used in the call. Present if "denfun" is in input argument what.
probfun	Present if "probfun" is in input argument what. An array of dimension ny×npred×nsave with the MCMC realisations of the conditional probabilities. Here ny is the length of argument q.value
q.value	Argument q.value used in the call. Present if "probfun" is in input argument what.

See Also

DDPstar

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005),
 data = dde, mcmc = list(nburn = 20000, nsave = 15000, nskip = 1),
  prior = list(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20),
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE)

# Data frame for predictions (regression, variance, quartiles and probabilities functions)
df_pred <- data.frame(DDE = seq(min(dde$DDE), max(dde$DDE), length = 50))

# Data frame for conditional densities
# Compute the densities for 4 different DDE values
df_pred_den <- data.frame(DDE = quantile(dde$DDE, c(0.1, 0.6, 0.9, 0.99)))
sequenceGAD <- seq(min(dde$GAD), max(dde$GAD), length = 100)

# Regression and variance function
  reg_var_DDE <- predict(fit_dde, newdata = df_pred,
    what = c("regfun", "varfun"),
    parallel = "multicore",
    ncpus = 2)
    # Regression function
    reg_m  <- apply(reg_var_DDE$regfun, 1, median)
    reg_ql <- apply(reg_var_DDE$regfun, 1, quantile, 0.025)
    reg_qh <- apply(reg_var_DDE$regfun, 1, quantile, 0.975)

    plot(dde$DDE, dde$GAD, xlab = "DDE (mg/L)",
      ylab = "Gestational age at delivery (in weeks)",
      main = "Regression function")
    lines(df_pred$DDE, reg_m, lwd = 2)
    lines(df_pred$DDE, reg_ql, lty = 2)
    lines(df_pred$DDE, reg_qh, lty = 2)

    # Variance function
    var_m  <- apply(reg_var_DDE$varfun, 1, median)
    var_ql <- apply(reg_var_DDE$varfun, 1, quantile, 0.025)
```

```
  var_qh <- apply(reg_var_DDE$varfun, 1, quantile, 0.975)

  plot(df_pred$DDE, var_m, type = "l", lwd = 2,
    xlab = "DDE (mg/L)",
    ylab = "",
    main = "Variance function")
  lines(df_pred$DDE, var_ql, lty = 2)
  lines(df_pred$DDE, var_qh, lty = 2)

# Quartiles
  p <- c(0.25, 0.5, 0.75)
  quart_DDE <- predict(fit_dde, newdata = df_pred,
    what = "quantfun", quant.probs = p,
    parallel = "multicore", ncpus = 2)

  quart_m <- apply(quart_DDE$quantfun , c(1,3), median)
  quart_ql <- apply(quart_DDE$quantfun , c(1,3), quantile, 0.025)
  quart_qh <- apply(quart_DDE$quantfun , c(1,3), quantile, 0.975)

  plot(dde$DDE, dde$GAD, xlab = "DDE (mg/L)",
      ylab = "Gestational age at delivery (in weeks)",
      main = "Conditional quartiles")

  # Q1
  lines(df_pred$DDE, quart_m[,1], lwd = 2, col = 2)
  lines(df_pred$DDE, quart_ql[,1], lty = 2, col = 2)
  lines(df_pred$DDE, quart_qh[,1], lty = 2, col = 2)

  # Median
  lines(df_pred$DDE, quart_m[,2], lwd = 2)
  lines(df_pred$DDE, quart_ql[,2], lty = 2)
  lines(df_pred$DDE, quart_qh[,2], lty = 2)

  # Q3
  lines(df_pred$DDE, quart_m[,3], lwd = 2, col = 3)
  lines(df_pred$DDE, quart_ql[,3], lty = 2, col = 3)
  lines(df_pred$DDE, quart_qh[,3], lty = 2, col = 3)

# Conditional densities
  den_DDE <- predict(fit_dde, newdata = df_pred_den,
    what = "denfun", den.grid = sequenceGAD,
    parallel = "multicore", ncpus = 2)

  den_m  <- apply(den_DDE$denfun, c(1,2), median)
  den_ql <- apply(den_DDE$denfun, c(1,2), quantile, 0.025)
  den_qh <- apply(den_DDE$denfun, c(1,2), quantile, 0.975)

  op <- par(mfrow = c(2,2))
  plot(sequenceGAD, den_m[,1], type = "l", lwd = 2,
    xlab = "DDE (mg/L)",
    ylab = "",
    main = "DDE = 12.57 (10th percentile)")
  lines(sequenceGAD, den_ql[,1], lty = 2)
```

```
  lines(sequenceGAD, den_qh[,1], lty = 2)

  plot(sequenceGAD, den_m[,2], type = "l", lwd = 2,
  xlab = "DDE (mg/L)",
    ylab = "",
    main = "DDE = 28.44 (60th percentile)")
  lines(sequenceGAD, den_ql[,2], lty = 2)
  lines(sequenceGAD, den_qh[,2], lty = 2)

  plot(sequenceGAD, den_m[,3], type = "l", lwd = 2,
    xlab = "DDE (mg/L)",
    ylab = "",
    main = "DDE = 53.72 (90th percentile)")
  lines(sequenceGAD, den_ql[,3], lty = 2)
  lines(sequenceGAD, den_qh[,3], lty = 2)
  plot(sequenceGAD, den_m[,4], type = "l", lwd = 2,
    xlab = "DDE (mg/L)",
    ylab = "",
    main = "DDE = 105.47 (99th percentile)")
  lines(sequenceGAD, den_ql[,4], lty = 2)
  lines(sequenceGAD, den_qh[,4], lty = 2)
  par(op)

# Conditional probabilities
  probs_DDE <- predict(fit_dde, newdata = df_pred,
    what = "probfun", q.value = c(37),
    parallel = "multicore", ncpus = 2)

  prob_m  <- apply(probs_DDE$probfun, c(1,2), median)
  prob_ql <- apply(probs_DDE$probfun, c(1,2), quantile, 0.025)
  prob_qh <- apply(probs_DDE$probfun, c(1,2), quantile, 0.975)

  plot(df_pred$DDE, prob_m, type = "l", lwd = 2, ylim = c(0, 1),
      xlab = "DDE (mg/L)",
      ylab = "P(GAD < 37 | DDE)",
      main = "Conditional Probabilities (premature delivery)")
  lines(df_pred$DDE, prob_ql, lty = 2)
  lines(df_pred$DDE, prob_qh, lty = 2)
```

`predictive.checks.DDPstar` *Posterior predictive checks.*

Description

Implements posterior predictive checks for objects of class `DDPstar` as produced by function `DDPstar`.

Usage

```
predictive.checks.DDPstar(object, statistics = c("min", "max", "kurtosis", "skewness"),
  devnew = TRUE, ndensity = 512, trim = 0.025)
```

Arguments

object	An object of class DDPstar as produced by function DDPstar.
statistics	Character vector. Statistics to be used for the posterior predictive checking. By default, "min", "max", "kurtosis" and "skewness".
devnew	A logical value. If TRUE, each plot is depicted in a new graphic device.
ndensity	An integer giving the number of equally spaced points at which the density of the response variable and of the simulated datasets from the posterior predictive distribution (see more on Details) is to be estimated (for more details see the help of the function density in the stats package)
trim	the fraction (0 to 0.5) of (simulated) test statistics to be trimmed.

Details

Compares a selected test statistic computed based on the observed response variable against the same test statistics computed based on simulated data from the posterior predictive distribution of the response variable obtained using a DDPstar model. The following graphics are depicted: (1) histograms of the desired statistics computed from simulated datasets (nsave of them) from the posterior predictive distribution of the response variable. In these plots, the estimated statistics from the observed response variable are also depicted. (2) Kernel density estimates computed computed from simulated datasets (nsave of them) from the posterior predictive distribution of the response variable. In these plots, the kernel density estimate of the observed response variable is also depicted. For a detailed discussion about predictive checks, see Gabry et al. (2019).

Value

As a result, the function provides a list with the following components:

yrep	numeric matrix. Each column of the matrix (there are nsave of them) corresponds to a dataset generated from the posterior predictive distribution of the of the response variable.
y	numeric vector containing the observed response variable.

References

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. (2019). Visualization in Bayesian workflow. Journal of the Royal Statistical Society, Series A, 182, 1-14.

See Also

DDPstar

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005),
  data = dde, mcmc = list(nburn = 20000, nsave = 15000, nskip = 1),
  prior = list(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20),
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE)
op <- par(mfrow = c(2,3))
predictive.checks.DDPstar(fit_dde)
par(op)
```

---

print.DDPstar *Print method for DDPstar objects*

Description

Default print method for objects fitted with function `DDPstar()`.

Usage

```
## S3 method for class 'DDPstar'
print(x, ...)
```

Arguments

`x`	An object of class `DDPstar` as produced by `DDPstar()`.
`...`	Further arguments passed to or from other methods. Not yet implemented.

Details

A short summary is printed.

Value

No return value, called for side effects

See Also

`DDPstar`

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005),
  data = dde, mcmc = list(nburn = 20000, nsave = 15000, nskip = 1),
  prior = list(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20),
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE, compute.DIC = TRUE)
fit_dde
```

`priorcontrol` *Prior information for the DDPstar model*

Description

This function is used to set various parameters controlling the prior information to be used in the `DDPstar` function.

Usage

```
priorcontrol(m0 = NA, S0 = NA, nu = NA, Psi = NA,
  atau = 1, btau = 0.005, a = 2, b = NA, alpha.fixed = FALSE,
  alpha = 1, aalpha = 2, balpha = 2, L = 10)
```

Arguments

`m0`	A numeric vector. Hyperparameter; mean vector of the (multivariate) normal prior distribution for the parametric coefficients. `NA` signals autoinitialization, with defaults: a vector, of length Q (where Q is the number of parametric coefficients), of zeros, if the data are standardised and the least squares estimates of the regression coefficients if the data are not standardised.
`S0`	A numeric matrix. Hyperparameter; covariance matrix of the (multivariate) normal prior distribution for the parametric coefficients. `NA` signals autoinitialization, with defaults: $10I_{Q\times Q}$ if the data are standardised (where Q is the number of parametric coefficients) and $\hat{\mathbf{\Sigma}}$ if the data are not standardised, where $\hat{\mathbf{\Sigma}}$ is the estimated covariance matrix of the parametric coefficients obtained by fitting a linear model to the data.
`nu`	A numeric value. Hyperparameter; degrees of freedom of the Wishart prior distribution for the precision matrix associated with the parametric coefficients. `NA` signals autoinitialization, with default $Q + 2$ (where Q is the number of parametric coefficients)

`Psi` A numeric matrix. Hyperparameter; scale matrix of the Wishart prior distribution for the precision matrix associated with the parametric coefficients. `NA` signals autoinitialization, with defaults: $I_{Q\times Q}$ if the data are standardised (where Q is the number of parametric coefficients) and to $30\hat{\boldsymbol{\Sigma}}$ if the data are not standardised, where $\hat{\boldsymbol{\Sigma}}$ is the estimated covariance matrix of the parametric coefficients obtained by fitting a linear model to the data.

`atau` A numeric value. Hyperparameter; shape parameter of the gamma prior distribution for the precisions (inverse variances) of the smooth/nonlinear/random terms. The default is 1.

`btau` A numeric value. Hyperparameter; rate parameter of the gamma prior distribution for the precisions (inverse variances) of the smooth/nonlinear/random terms. The default is 0.005.

`a` A numeric value. Hyperparameter; shape parameter of the gamma prior distribution for the precisions (inverse variances) of each component. The default is 2.

`b` A numeric value. Hyperparameter; shape parameter of the gamma prior distribution for the precisions (inverse variances) of each component. `NA` signals autoinitialization, with defaults: 0.5 if the data are standardised and $\frac{\hat{\sigma}^2}{2}$ if the data are not standardised

`alpha.fixed` A logical value. If `TRUE`, the precision parameter of the Dirichlet Process is considered fixed. If `FALSE`, a Gamma prior distribution is placed on it.

`alpha` A numeric value. Applicable when `alpha.fixed = TRUE`. Precision parameter of the Dirichlet Process. The default is 1.

`aalpha` A numeric value. Applicable when `alpha.fixed = FALSE`. Hyperparameter; shape parameter for the Gamma prior distribution of the precision parameter of the Dirichlet process prior. The default is 2.

`balpha` A numeric value. Applicable when `alpha.fixed = FALSE`. Hyperparameter; rate parameter for the Gamma prior distribution of the precision parameter of the Dirichlet process prior. The default is 2.

`L` A numeric value. Upper bound on the number of mixture components. Setting L = 1 corresponds to a normal model. The default is 10.

Value

A list with components for each of the possible arguments.

See Also

DDPstar

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

prior <- priorcontrol(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20)
```

```
set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005),
  data = dde, mcmc = list(nburn = 20000, nsave = 15000, nskip = 1), prior = prior,
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE, compute.DIC = TRUE)
summary(fit_dde)
```

---

quantileResiduals *Quantile residuals.*

Description

Produces quantile residuals for objects of class DDPstar as produced by function DDPstar.

Usage

```
## S3 method for class 'DDPstar'
quantileResiduals(object, parallel = c("no", "multicore", "snow"),
  ncpus = 1, cl = NULL, ...)
```

Arguments

object	An object of class DDPstar as produced by function DDPstar.
parallel	A characters string with the type of parallel operation: either "no" (default), "multicore" (not available on Windows) or "snow".
ncpus	An integer with the number of processes to be used in parallel operation. Defaults to 1.
cl	An object inheriting from class cluster (from the parallel package), specifying an optional parallel or snow cluster if parallel = "snow". If not supplied, a cluster on the local machine is created for the duration of the call.
...	further arguments passed to or from other methods. Not yet implemented

Details

Quantile residuals (Dunn and Smyth, 1996) are based on the well-known fact that for a continuous random variable, say Y, with CDF given by F, one has that $F(Y) \sim U(0,1)$. As a consequence, quantile residuals defined by $\hat{r}_j = \Phi^{-1}(\hat{F}(y_j))$, $j = 1, \ldots, n$, should follow, approximately, a standard normal distribution if a correct model has been specified. A quantile-quantile (QQ) plot can then be used to determine deviations of the quantile residuals from the standard normal distribution.

Value

As a result, the function provides a list with the following components:

x	Theoretical quantiles
y	Sample quantiles: posterior mean
ymin	Sample quantiles: posterior 0.025 quantile
ymax	Sample quantiles: posterior 0.975 quantile

References

Dunn, P. K. and Smyth, G. K. (1996). Randomized Quantile Residuals. Journal of Computational and Graphical Statistics, 5(3), 236-244.

See Also

DDPstar

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005),
  data = dde, mcmc = list(nburn = 20000, nsave = 15000, nskip = 1),
  prior = list(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20),
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE)

qres <- quantileResiduals.DDPstar(fit_dde)

plot(qres$x, qres$y, xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
  cex.main = 2, cex.lab = 1.5, cex.axis = 1.5)
lines(qres$x, qres$ymin, lty = 2, lwd = 2)
lines(qres$x, qres$ymax, lty = 2, lwd = 2)
abline(a = 0, b = 1, col ="red", lwd = 2)
```

---

rae *Defining random effects in DDPstar formulae*

Description

Auxiliary function used to define random effects terms in a DDPstar model formula.

Usage

```
rae(x, atau = 1, btau = 0.005)
```

Arguments

x	the x-variable (factor) that defines the random effects term.
atau	A numeric value. Hyperparameter; shape parameter of the gamma prior distribution for the precision (inverse variance) of the random effect term. The default is 1.
btau	A numeric value. Hyperparameter; rate parameter of the gamma prior distribution for the precision (inverse variance) of the random effect term. The default is 0.005.

Details

The functions is designed to represent random effects in DDPstar formulae.

Value

The function is interpreted in the formula of a DDPstar model and creates the right framework for fitting the random effect. List containing the following elements:

vars	name of the covariate involved.
atau	shape parameter of the gamma prior distribution
btau	rate parameter of the gamma prior distribution

References

Rodriguez-Alvarez, M. X, Inacio, V. and Klein N. (2025). Density regression via Dirichlet process mixtures of normal structured additive regression models. Accepted for publication in Statistics and Computing (DOI: 10.1007/s11222-025-10567-0).

See Also

f, DDPstar

Examples

```
# For an example including random effects, we refer to Rodriguez-Alvarez, Inacio et al. (2025)
# and associated codes (found in https://bitbucket.org/mxrodriguez/ddpstar)
```

summary.DDPstar *Summary method for DDPstar objects*

Description

Default summary method for objects fitted with function DDPstar.

Usage

```
## S3 method for class 'DDPstar'
summary(object, ...)
```

Arguments

object	An object of class DDPstar as produced by DDPstar().
...	Further arguments passed to or from other methods. Not yet implemented.

Details

The information printed includes: (1) the call to the function and (2) the sample size. If required, the function aso provides the log pseudo marginal likelihood (LPML), the widely applicable information criterion (WAIC) and/or the deviance information criterion (DIC).

Value

No return value, called for side effects

See Also

DDPstar

Examples

```
library(DDPstar)
data(dde)
dde$GAD <- dde$GAD/7 # GAD in weeks

set.seed(10) # For reproducibility
fit_dde <- DDPstar(formula = GAD ~ f(DDE, bdeg = 3, nseg = 20, pord = 2, atau = 1, btau = 0.005),
  data = dde, mcmc = list(nburn = 20000, nsave = 15000, nskip = 1),
  prior = list(a = 2, b = 0.5, aalpha = 2, balpha = 2, L = 20),
  standardise = TRUE, compute.lpml = TRUE, compute.WAIC = TRUE, compute.DIC = TRUE)
summary(fit_dde)
```

Index

∗ **datasets**
dde, 3
∗ **package**
DDPstar-package, 2

dde, 3
DDPstar, 4, *10, 11*, *13*, *16, 17*, *19*, *21–23*
DDPstar-package, 2

f, *7*, 8, *22*
formula, *4*

glm, *7*

mcmccontrol, *5*, 10

predict.DDPstar, *7*, 11
predictive.checks.DDPstar, 15
print.DDPstar, 17
priorcontrol, *5*, 18

quantileResiduals, 20

rae, *7*, *10*, 21

summary.DDPstar, 23