Package ‘bsamGP’

July 22, 2025

Type Package

Title Bayesian Spectral Analysis Models using Gaussian Process Priors

Version 1.2.7

Date 2025-07-17

Author Seongil Jo [aut],
Taeryon Choi [aut],
Beomjo Park [aut, cre],
Peter J. Lenk [ctb]

Maintainer Beomjo Park <beomjop@gmail.com>

Imports MASS, ggplot2, gridExtra

Description Contains functions to perform Bayesian inference using a spectral analysis of Gaussian process priors. Gaussian processes are represented with a Fourier series based on cosine basis functions. Currently the package includes parametric linear models, partial linear additive models with/without shape restrictions, generalized linear additive models with/without shape restrictions, and density estimation model. To maximize computational efficiency, the actual Markov chain Monte Carlo sampling for each model is done using codes written in FORTRAN 90. This software has been developed using funding supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. NRF-2016R1D1A1B03932178 and no. NRF-2017R1D1A3B03035235).

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-07-18 08:40:12 UTC

Contents

`blq` *Bayesian Quantile Regression*

Description

This function fits a Bayesian quantile regression model.

Usage

```
blq(formula, data = NULL, p, mcmc = list(), prior = list(), marginal.likelihood = TRUE)
```

Arguments

- `formula` — an object of class "`formula`"
- `data` — an optional data frame.
- `p` — quantile of interest (default=0.5).
- `mcmc` — a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): `nblow (1000)` giving the number of MCMC in transition period, `nskip (1)` giving the thinning interval, `smcmc (1000)` giving the number of MCMC for analysis.
- `prior` — a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): `beta_m0` and `beta_v0` giving the hyperparameters of the multivariate normal distribution for parametric part including intercept, `sigma2_m0` and `sigma2_v0` giving the prior mean and variance of the inverse gamma prior for the scale parameter of response.
- `marginal.likelihood` — a logical variable indicating whether the log marginal likelihood is calculated. The methods of Gelfand and Dey (1994) is used.

Details

This generic function fits a Bayesian quantile regression model.

Let y_i and w_i be the response and the vector of parametric predictors, respectively. Further, let $x_{i,k}$ be the covariate related to the response, linearly. The model is as follows.

$$y_i = w_i^T \beta + \epsilon_i,\ i = 1, \ldots, n,$$

where the error terms $\{\epsilon_i\}$ are a random sample from an asymmetric Laplace distribution, $ALD_p(0, \sigma^2)$, which has the following probability density function:

$$ALD_p(\epsilon; \mu, \sigma^2) = \frac{p(1-p)}{\sigma^2} \exp\Big(- \frac{(x-\mu)[p - I(x \le \mu)]}{\sigma^2} \Big),$$

where $0 < p < 1$ is the skew parameter, $\sigma^2 > 0$ is the scale parameter, $-\infty < \mu < \infty$ is the location parameter, and $I(\cdot)$ is the indication function.

The conjugate priors are assumed for β and σ:

$$\beta|\sigma \sim N(m_{0,\beta}, \sigma^2 V_{0,\beta}), \quad \sigma^2 \sim IG\Big(\frac{r_{0,\sigma}}{2}, \frac{s_{0,\sigma}}{2}\Big)$$

Value

An object of class `blm` representing the Bayesian parametric linear model fit. Generic functions such as `print` and `fitted` have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws`, the posterior samples of the fitted values are stored in the list `fit.draws`, and the MCMC samples for the log marginal likelihood are saved in the list `loglik.draws`. The output list also includes the following objects:

post.est	posterior estimates for all parameters in the model.
lmarg	log marginal likelihood using Gelfand-Dey method.
rsquarey	correlation between y and $\hat{y}$.
call	the matched call.
mcmctime	running time of Markov chain from system.time().

References

Gelfand, A. E. and Dey, K. K. (1994) Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B - Statistical Methodology*, **56**, 501-514.

Kozumi, H. and Kobayashi, G. (2011) Gibbs sampling methods for Bayesian quantile regression. *Journal of Statistical Computation and Simulation*, **81**(11), 1565-1578.

See Also

blr, gblr

Examples

```
#####################
# Simulated example #
#####################

# Simulate data
set.seed(1)

n <- 100
w <- runif(n)
y <- 3 + 2*w + rald(n, scale = 0.8, p = 0.5)

# Fit median regression
fout <- blq(y ~ w, p = 0.5)

# Summary
print(fout); summary(fout)

# fitted values
fit <- fitted(fout)

# Plots
plot(fout)
```

---

blr *Bayesian Linear Regression*

Description

This function fits a Bayesian linear regression model using scale invariant prior.

Usage

```
blr(formula, data = NULL, mcmc = list(), prior = list(), marginal.likelihood = TRUE)
```

Arguments

formula	an object of class "formula"
data	an optional data frame.
mcmc	a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): nblow (1000) giving the number of MCMC in transition period, nskip (1) giving the thinning interval, smcmc (1000) giving the number of MCMC for analysis.
prior	a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): beta_m0 and beta_v0 giving the hyperparameters of the multivariate normal distribution for parametric part including intercept, sigma2_m0 and sigma2_v0 giving the prior mean and variance of the inverse gamma prior for the scale parameter of response.
marginal.likelihood	a logical variable indicating whether the log marginal likelihood is calculated.

Details

This generic function fits a Bayesian linear regression model using scale invariant prior.

Let y_i and w_i be the response and the vector of parametric predictors, respectively. The model for regression function is as follows.

$$y_i = w_i^T \beta + \epsilon_i,\ i = 1, \ldots, n,$$

where the error terms $\{\epsilon_i\}$ are a random sample from a normal distribution, $N(0, \sigma^2)$.

The conjugate priors are assumed for β and σ:

$$\beta|\sigma \sim N(m_{0,\beta}, \sigma^2 V_{0,\beta}), \quad \sigma^2 \sim IG\Big(\frac{r_{0,\sigma}}{2}, \frac{s_{0,\sigma}}{2}\Big)$$

Value

An object of class blm representing the Bayesian spectral analysis model fit. Generic functions such as print and fitted have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list mcmc.draws and the posterior samples of the fitted values are stored in the list fit.draws. The output list also includes the following objects:

post.est	posterior estimates for all parameters in the model.
lmarg	log marginal likelihood.
rsquarey	correlation between y and $\hat{y}$.
call	the matched call.
mcmctime	running time of Markov chain from system.time().

See Also

blq, gblr

Examples

```
#####################
# Simulated example #
#####################

# Simulate data
set.seed(1)

n <- 100
w <- runif(n)
y <- 3 + 2*w + rnorm(n, sd = 0.8)

# Fit the model with default priors and mcmc parameters
fout <- blr(y ~ w)

# Summary
print(fout); summary(fout)

# Fitted values
fit <- fitted(fout)

# Plots
plot(fout)
```

---

bsad *Bayesian Semiparametric Density Estimation*

Description

This function fits a semiparametric model, which consists of parametric and nonparametric components, for estimating density using a logistic Gaussian process.

Usage

```
bsad(x, xmin, xmax, nint, MaxNCos, mcmc = list(), prior = list(),
smoother = c('geometric', 'algebraic'),
parametric = c('none', 'normal', 'gamma', 'laplace'), marginal.likelihood = TRUE,
verbose = FALSE)
```

Arguments

x	a vector giving the data from which the density estimate is to be computed.
xmin	minimum value of x.
xmax	maximum value of x.

`nint` number of grid points for plots (need to be odd). The default is 201.

`MaxNCos` maximum number of Fourier coefficients.

`mcmc` a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): `kappaloop (5)` giving the number of MCMC loops within each choice of kappa, `nblow (10000)` giving the number of MCMC in transition period, `nskip (10)` giving the thinning interval, `smcmc (1000)` giving the number of MCMC for analysis, and `ndisp (1000)` giving the number of saved draws to be displayed on screen (the function reports on the screen when every `ndisp` iterations have been carried out).

`prior` a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): `gmax` giving maximum value for gamma (default = 5), `PriorProbs` giving prior probability of parametric and semiparametric models, `beta_m0` and `beta_v0` giving the hyperparameters for prior distribution of the parametric coefficients, `r0` and `s0` giving the hyperparameters of σ^2 for the logits, `u0` and `v0` giving the hyperparameters of τ^2 for Fourier coefficients, `PriorKappa` and `KappaGrid` giving prior on the number of cosine terms.

`smoother` types of smoothing priors for Fourier coefficients. See Details.

`parametric` specifying a distribution of the parametric part to be test.

`marginal.likelihood` a logical variable indicating whether the log marginal likelihood is calculated.

`verbose` a logical variable. If `TRUE`, the iteration number and the Metropolis acceptance rate are printed to the screen.

Details

This generic function fits a semiparametric model, which consists of parametric and nonparametric, for density estimation (Lenk, 2003):

$$f(x|\beta, Z) = \frac{\exp[h(x)^\top \beta + Z(x)]}{\int_{\mathcal{X}} \exp[h(y)^\top \beta + Z(y)] dG(y)}$$

where Z is a zero mean, second-order Gaussian process with bounded, continuous covariance function. i.e.,

$$E[Z(x), Z(y)] = \sigma(x, y), \quad \int_{\mathcal{X}} Z dG = 0 \ \ (a.s.)$$

Using the Karhunen-Loeve Expansion, Z is represented as infinite series with random coefficients

$$Z(x) = \sum_{j=1}^{\infty} \theta_j \varphi_j(x),$$

where $\{\varphi_j\}$ is the cosine basis, $\varphi_j(x) = \sqrt{2}\cos[j\pi G(x)]$.

For the random Fourier coefficients of the expansion, two smoother priors are assumed (optional),

$$\theta_j|\tau, \gamma \sim N(0, \tau^2 \exp[-j\gamma]), \ j \geq 1 \ (geometric\ smoother)$$

$$\theta_j|\tau, \gamma \sim N(0, \tau^2 \exp[-ln(j+1)\gamma]), \ j \geq 1 \ (algebraic\ smoother)$$

The coefficient β have the popular normal prior,

$$\beta|m_{0,\beta}, V_{0,\beta} \sim N(m_{0,\beta}, V_{0,\beta})$$

To complete the model specification, independent hyper priors are assumed,

$$\tau^2|r_0, s_0 \sim IGa(r_0/2, s_0/2)$$

$$\gamma|w_0 \sim Exp(w_0)$$

Note that the posterior algorithm is based on computing a discrete version of the likelihood over a fine mesh on $\mathcal{X}$.

Value

An object of class `bsad` representing the Bayesian spectral analysis density estimation model fit. Generic functions such as `print`, `fitted` and `plot` have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws`, the posterior samples of the fitted values are stored in the list `fit.draws`, and the MCMC samples for the log marginal likelihood are saved in the list `loglik.draws`. The output list also includes the following objects:

`post.est`	posterior estimates for all parameters in the model.
`lmarg`	log marginal likelihood.
`ProbProbs`	posterior probability of models.
`call`	the matched call.
`mcmctime`	running time of Markov chain from `system.time()`.

References

Jo, S., Choi, T., Park, B. and Lenk, P. (2019). bsamGP: An R Package for Bayesian Spectral Analysis Models Using Gaussian Process Priors. *Journal of Statistical Software*, **90**, 310-320.

Lenk, P. (2003) Bayesian semiparametric density estimation and model verification using a logistic Gaussian process. *Journal of Computational and Graphical Statistics*, **12**, 548-565.

Examples

```
## Not run:
############################
# Old Faithful geyser data #
############################
data(faithful)
attach(faithful)

# mcmc parameters
mcmc <- list(nblow = 10000,
           smcmc = 1000,
           nskip = 10,
           ndisp = 1000,
```

```
            kappaloop = 5)

# fits BSAD model
fout <- bsad(x = eruptions, xmin = 0, xmax = 8, nint = 501, mcmc = mcmc,
             smoother = 'geometric', parametric = 'gamma')

# Summary
print(fout); summary(fout)

# fitted values
fit <- fitted(fout)

# predictive density plot
plot(fit, ask = TRUE)

detach(faithful)

## End(Not run)
```

---

bsaq *Bayesian Shape-Restricted Spectral Analysis Quantile Regression*

Description

This function fits a Bayesian semiparametric quantile regression model to estimate shape-restricted functions using a spectral analysis of Gaussian process priors.

Usage

```
bsaq(formula, xmin, xmax, p, nbasis, nint, mcmc = list(), prior = list(),
shape = c('Free', 'Increasing', 'Decreasing', 'IncreasingConvex', 'DecreasingConcave',
'IncreasingConcave', 'DecreasingConvex', 'IncreasingS', 'DecreasingS',
'IncreasingRotatedS','DecreasingRotatedS','InvertedU','Ushape',
'IncMultExtreme','DecMultExtreme'), nExtreme = NULL,
marginal.likelihood = TRUE, spm.adequacy = FALSE, verbose = FALSE)
```

Arguments

formula	an object of class "formula"
xmin	a vector or scalar giving user-specific minimum values of x. The default values are minimum values of x.
xmax	a vector or scalar giving user-specific maximum values of x. The default values are maximum values of x.
p	quantile of interest (default=0.5).
nbasis	number of cosine basis functions.
nint	number of grid points where the unknown function is evaluated for plotting. The default is 200.

`mcmc`	a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): `nblow0 (1000)` giving the number of initialization period for adaptive metropolis, `maxmodmet (5)` giving the maximum number of times to modify metropolis, `nblow (10000)` giving the number of MCMC in transition period, `nskip (10)` giving the thinning interval, `smcmc (1000)` giving the number of MCMC for analysis, and `ndisp (1000)` giving the number of saved draws to be displayed on screen (the function reports on the screen when every `ndisp` iterations have been carried out).
`prior`	a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): `iflagprior` choosing a smoothing prior for spectral coefficients (iflagprior=0 assigns T-Smoother prior (default), iflagprior=1 chooses Lasso-Smoother prior), `theta0_m0` and `theta0_s0` giving the hyperparameters for prior distribution of the spectral coefficients (`theta0_m0` and `theta0_s0` are used when the functions have shape-restriction), `tau2_m0`, `tau2_s0` and `w0` giving the prior mean and standard deviation of smoothing prior (When iflagprior=1, tau2_m0 is only used as the hyperparameter), `beta_m0` and `beta_v0` giving the hyperparameters of the multivariate normal distribution for parametric part including intercept, `sigma2_m0` and `sigma2_v0` giving the prior mean and variance of the inverse gamma prior for the scale parameter of response, `alpha_m0` and `alpha_s0` giving the prior mean and standard deviation of the truncated normal prior distribution for the constant of integration, `iflagpsi` determining the prior of slope for logisitic function in `S` or `U` shaped (iflagpsi=1 (default), slope ψ is sampled and iflagpsi=0, ψ is fixed), `psifixed` giving initial value (iflagpsi=1) or fixed value (iflagpsi=0) of slope, `omega_m0` and `omega_s0` giving the prior mean and standard deviation of the truncated normal prior distribution for the inflection point of `S` or `U` shaped function.
`shape`	a vector giving types of shape restriction.
`nExtreme`	a vector of extreme points for 'IncMultExtreme', 'DecMultExtreme' shape restrictions.
`marginal.likelihood`	a logical variable indicating whether the log marginal likelihood is calculated. The methods of Gelfand and Dey (1994) and Newton and Raftery (1994) are used.
`spm.adequacy`	a logical variable indicating whether the log marginal likelihood of linear model is calculated. The marginal likelihood gives the values of the linear regression model excluding the nonlinear parts.
`verbose`	a logical variable. If `TRUE`, the iteration number and the Metropolis acceptance rate are printed to the screen.

Details

This generic function fits a Bayesian spectral analysis quantile regression model for estimating shape-restricted functions using Gaussian process priors. For enforcing shape-restrictions, the model assumed that the derivatives of the functions are squares of Gaussian processes.

Let y_i and w_i be the response and the vector of parametric predictors, respectively. Further, let $x_{i,k}$ be the covariate related to the response through an unknown shape-restricted function. The model for estimating shape-restricted functions is as follows.

$$y_i = w_i^T \beta + \sum_{k=1}^{K} f_k(x_{i,k}) + \epsilon_i,\ i = 1, \ldots, n,$$

where f_k is an unknown shape-restricted function of the scalar $x_{i,k} \in [0, 1]$ and the error terms $\{\epsilon_i\}$ are a random sample from an asymmetric Laplace distribution, $ALD_p(0, \sigma^2)$, which has the following probability density function:

$$ALD_p(\epsilon; \mu, \sigma^2) = \frac{p(1-p)}{\sigma^2} \exp\Big(- \frac{(x-\mu)[p - I(x \leq \mu)]}{\sigma^2} \Big),$$

where $0 < p < 1$ is the skew parameter, $\sigma^2 > 0$ is the scale parameter, $-\infty < \mu < \infty$ is the location parameter, and $I(\cdot)$ is the indication function.

The prior of function without shape restriction is:

$$f(x) = Z(x),$$

where Z is a second-order Gaussian process with mean function equal to zero and covariance function $\nu(s, t) = E[Z(s)Z(t)]$ for $s, t \in [0, 1]$. The Gaussian process is expressed with the spectral representation based on cosine basis functions:

$$Z(x) = \sum_{j=0}^{\infty} \theta_j \varphi_j(x)$$

$$\varphi_0(x) = 1 \ \text{and}\ \varphi_j(x) = \sqrt{2} \cos(\pi j x),\ j \geq 1,\ 0 \leq x \leq 1$$

The shape-restricted functions are modeled by assuming the qth derivatives of f are squares of Gaussian processes:

$$f^{(q)}(x) = \delta Z^2(x) h(x),\ \ \delta \in \{1, -1\},\ \ q \in \{1, 2\},$$

where h is the squish function. For monotonic, monotonic convex, and concave functions, $h(x) = 1$, while for S and U shaped functions, h is defined by

$$h(x) = \frac{1 - \exp[\psi(x - \omega)]}{1 + \exp[\psi(x - \omega)]},\ \ \psi > 0,\ \ 0 < \omega < 1$$

For the spectral coefficients of functions without shape constraints, the scale-invariant prior is used (The intercept is included in β):

$$\theta_j | \sigma, \tau, \gamma \sim N(0, \sigma^2 \tau^2 \exp[-j\gamma]),\ j \geq 1$$

The priors for the spectral coefficients of shape restricted functions are:

$$\theta_0 | \sigma \sim N(m_{\theta_0}, \sigma v_{\theta_0}^2), \quad \theta_j | \sigma, \tau, \gamma \sim N(m_{\theta_j}, \sigma \tau^2 \exp[-j\gamma]),\ j \geq 1$$

To complete the model specification, the conjugate priors are assumed for β and σ:

$$\beta | \sigma \sim N(m_{0,\beta}, \sigma^2 V_{0,\beta}), \quad \sigma^2 \sim IG\Big(\frac{r_{0,\sigma}}{2}, \frac{s_{0,\sigma}}{2}\Big)$$

Value

An object of class `bsam` representing the Bayesian spectral analysis model fit. Generic functions such as `print`, `fitted` and `plot` have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws`, the posterior samples of the fitted values are stored in the list `fit.draws`, and the MCMC samples for the log marginal likelihood are saved in the list `loglik.draws`. The output list also includes the following objects:

`post.est`	posterior estimates for all parameters in the model.
`lmarg.lm`	log marginal likelihood for linear quantile regression model.
`lmarg.gd`	log marginal likelihood using Gelfand-Dey method.
`lmarg.nr`	log marginal likelihood using Netwon-Raftery method, which is biased.
`rsquarey`	correlation between y and $\hat{y}$.
`call`	the matched call.
`mcmctime`	running time of Markov chain from `system.time()`.

References

Jo, S., Choi, T., Park, B. and Lenk, P. (2019). bsamGP: An R Package for Bayesian Spectral Analysis Models Using Gaussian Process Priors. *Journal of Statistical Software*, **90**, 310-320.

Lenk, P. and Choi, T. (2017) Bayesian Analysis of Shape-Restricted Functions using Gaussian Process Priors. *Statistica Sinica*, **27**: 43-69.

Gelfand, A. E. and Dey, K. K. (1994) Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B - Statistical Methodology*, **56**, 501-514.

Kozumi, H. and Kobayashi, G. (2011) Gibbs sampling methods for Bayesian quantile regression. *Journal of Statistical Computation and Simulation*, **81**(11), 1565-1578.

Newton, M. A. and Raftery, A. E. (1994) Approximate Bayesian inference with the weighted likelihood bootstrap (with discussion). *Journal of the Royal Statistical Society. Series B - Statistical Methodology*, **56**, 3-48.

See Also

`bsar`, `gbsar`

Examples

```
## Not run:
######################
# Increasing-concave #
######################

# Simulate data
set.seed(1)

n <- 200
```

```
x <- runif(n)
y <- log(1 + 10*x) + rald(n, scale = 0.5, p = 0.5)

# Number of cosine basis functions
nbasis <- 50

# Fit the model with default priors and mcmc parameters
fout1 <- bsaq(y ~ fs(x), p = 0.25, nbasis = nbasis,
              shape = 'IncreasingConcave')
fout2 <- bsaq(y ~ fs(x), p = 0.5, nbasis = nbasis,
              shape = 'IncreasingConcave')
fout3 <- bsaq(y ~ fs(x), p = 0.75, nbasis = nbasis,
              shape = 'IncreasingConcave')

# fitted values
fit1 <- fitted(fout1)
fit2 <- fitted(fout2)
fit3 <- fitted(fout3)

# plots
plot(x, y, lwd = 2, xlab = 'x', ylab = 'y')
lines(fit1$xgrid, fit1$wbeta$mean[1] + fit1$fxgrid$mean, lwd=2, col=2)
lines(fit2$xgrid, fit2$wbeta$mean[1] + fit2$fxgrid$mean, lwd=2, col=3)
lines(fit3$xgrid, fit3$wbeta$mean[1] + fit3$fxgrid$mean, lwd=2, col=4)
legend('topleft', legend = c('1st Quartile', '2nd Quartile', '3rd Quartile'),
       lwd = 2, col = 2:4, lty = 1)


## End(Not run)
```

`bsaqdpm` *Bayesian Shape-Restricted Spectral Analysis Quantile Regression with Dirichlet Process Mixture Errors*

Description

This function fits a Bayesian semiparametric quantile regression model to estimate shape-restricted functions using a spectral analysis of Gaussian process priors. The model assumes that the errors follow a Dirichlet process mixture model.

Usage

```
bsaqdpm(formula, xmin, xmax, p, nbasis, nint,
mcmc = list(), prior = list(), egrid, ngrid = 500,
shape = c('Free', 'Increasing', 'Decreasing', 'IncreasingConvex', 'DecreasingConcave',
'IncreasingConcave', 'DecreasingConvex', 'IncreasingS', 'DecreasingS',
'IncreasingRotatedS', 'DecreasingRotatedS', 'InvertedU', 'Ushape'),
verbose = FALSE)
```

Arguments

formula	an object of class "formula"
xmin	a vector or scalar giving user-specific minimum values of x. The default values are minimum values of x.
xmax	a vector or scalar giving user-specific maximum values of x. The default values are maximum values of x.
p	quantile of interest (default=0.5).
nbasis	number of cosine basis functions.
nint	number of grid points where the unknown function is evaluated for plotting. The default is 200.
mcmc	a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): nblow0 (1000) giving the number of initialization period for adaptive metropolis, maxmodmet (5) giving the maximum number of times to modify metropolis, nblow (10000) giving the number of MCMC in transition period, nskip (10) giving the thinning interval, smcmc (1000) giving the number of MCMC for analysis, and ndisp (1000) giving the number of saved draws to be displayed on screen (the function reports on the screen when every ndisp iterations have been carried out).
prior	a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): iflagprior choosing a smoothing prior for spectral coefficients (iflagprior=0 assigns T-Smoother prior (default), iflagprior=1 chooses Lasso-Smoother prior), theta0_m0 and theta0_s0 giving the hyperparameters for prior distribution of the spectral coefficients (theta0_m0 and theta0_s0 are used when the functions have shape-restriction), tau2_m0, tau2_s0 and w0 giving the prior mean and standard deviation of smoothing prior (When iflagprior=1, tau2_m0 is only used as the hyperparameter), beta_m0 and beta_v0 giving the hyperparameters of the multivariate normal distribution for parametric part including intercept, sigma2_m0 and sigma2_v0 giving the prior mean and variance of the inverse gamma prior for the scale parameter of response, alpha_m0 and alpha_s0 giving the prior mean and standard deviation of the truncated normal prior distribution for the constant of integration, iflagpsi determining the prior of slope for logisitic function in S or U shaped (iflagpsi=1 (default), slope ψ is sampled and iflagpsi=0, ψ is fixed), psifixed giving initial value (iflagpsi=1) or fixed value (iflagpsi=0) of slope, omega_m0 and omega_s0 giving the prior mean and standard deviation of the truncated normal prior distribution for the inflection point of S or U shaped function.
egrid	a vector giving grid points where the residual density estimate is evaluated. The default range is from -10 to 10.
ngrid	a vector giving number of grid points where the residual density estimate is evaluated. The default value is 500.
shape	a vector giving types of shape restriction.
verbose	a logical variable. If TRUE, the iteration number and the Metropolis acceptance rate are printed to the screen.

Details

This generic function fits a Bayesian spectral analysis quantile regression model for estimating shape-restricted functions using Gaussian process priors. For enforcing shape-restrictions, the model assumes that the derivatives of the functions are squares of Gaussian processes. The model also assumes that the errors follow a Dirichlet process mixture model.

Let y_i and w_i be the response and the vector of parametric predictors, respectively. Further, let $x_{i,k}$ be the covariate related to the response through an unknown shape-restricted function. The model for estimating shape-restricted functions is as follows.

$$y_i = w_i^T \beta + \sum_{k=1}^{K} f_k(x_{i,k}) + \epsilon_i,\ i = 1, \ldots, n,$$

where f_k is an unknown shape-restricted function of the scalar $x_{i,k} \in [0, 1]$ and the error terms $\{\epsilon_i\}$ are a random sample from a Dirichlet process mixture of an asymmetric Laplace distribution, $ALD_p(0, \sigma^2)$, which has the following probability density function:

$$\epsilon_i \sim f(\epsilon) = \int ALD_p(\epsilon; 0, \sigma^2) dG(\sigma^2),$$

$$G \sim DP(M, G0),\ \ G0 = Ga\left(\sigma^{-2}; \frac{r_{0,\sigma}}{2}, \frac{s_{0,\sigma}}{2}\right).$$

The prior of function without shape restriction is:

$$f(x) = Z(x),$$

where Z is a second-order Gaussian process with mean function equal to zero and covariance function $\nu(s, t) = E[Z(s)Z(t)]$ for $s, t \in [0, 1]$. The Gaussian process is expressed with the spectral representation based on cosine basis functions:

$$Z(x) = \sum_{j=0}^{\infty} \theta_j \varphi_j(x)$$

$$\varphi_0(x) = 1 \text{ and } \varphi_j(x) = \sqrt{2}\cos(\pi j x),\ j \geq 1,\ 0 \leq x \leq 1$$

The shape-restricted functions are modeled by assuming the qth derivatives of f are squares of Gaussian processes:

$$f^{(q)}(x) = \delta Z^2(x) h(x),\ \ \delta \in \{1, -1\},\ \ q \in \{1, 2\},$$

where h is the squish function. For monotonic, monotonic convex, and concave functions, $h(x) = 1$, while for `S` and `U` shaped functions, h is defined by

$$h(x) = \frac{1 - \exp[\psi(x - \omega)]}{1 + \exp[\psi(x - \omega)]},\ \ \psi > 0,\ \ 0 < \omega < 1$$

For the spectral coefficients of functions without shape constraints, the scale-invariant prior is used (The intercept is included in β):

$$\theta_j | \tau, \gamma \sim N(0, \tau^2 \exp[-j\gamma]),\ j \geq 1$$

The priors for the spectral coefficients of shape restricted functions are:

$$\theta_0 \sim N(m_{\theta_0}, v_{\theta_0}^2), \quad \theta_j|\tau, \gamma \sim N(m_{\theta_j}, \tau^2 \exp[-j\gamma]),\ j \geq 1$$

To complete the model specification, the popular normal prior is assumed for β:

$$\beta| \sim N(m_{0,\beta}, V_{0,\beta})$$

Value

An object of class `bsam` representing the Bayesian spectral analysis model fit. Generic functions such as `print`, `fitted` and `plot` have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws`, the posterior samples of the fitted values are stored in the list `fit.draws`, and the MCMC samples for the log marginal likelihood are saved in the list `loglik.draws`. The output list also includes the following objects:

`post.est`	posterior estimates for all parameters in the model.
`lpml`	log pseudo marginal likelihood using Mukhopadhyay and Gelfand method.
`rsquarey`	correlation between y and $\hat{y}$.
`imodmet`	the number of times to modify Metropolis.
`pmet`	proportion of θ accepted after burn-in.
`call`	the matched call.
`mcmctime`	running time of Markov chain from `system.time()`.

References

Jo, S., Choi, T., Park, B. and Lenk, P. (2019). bsamGP: An R Package for Bayesian Spectral Analysis Models Using Gaussian Process Priors. *Journal of Statistical Software*, **90**, 310-320.

Kozumi, H. and Kobayashi, G. (2011) Gibbs sampling methods for Bayesian quantile regression. *Journal of Statistical Computation and Simulation*, **81**(11), 1565-1578.

Lenk, P. and Choi, T. (2017) Bayesian Analysis of Shape-Restricted Functions using Gaussian Process Priors. *Statistica Sinica*, **27**, 43-69.

MacEachern, S. N. and Müller, P. (1998) Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, **7**, 223-238.

Mukhopadhyay, S. and Gelfand, A. E. (1997) Dirichlet process mixed generalized linear models. *Journal of the American Statistical Association*, **92**, 633-639.

Neal, R. M. (2000) Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, **9**, 249-265.

See Also

`bsaq`, `bsardpm`

Examples

```
## Not run:
######################
# Increasing-concave #
######################

# Simulate data
set.seed(1)

n <- 500
x <- runif(n)
e <- c(rald(n/2, scale = 0.5, p = 0.5),
       rald(n/2, scale = 3, p = 0.5))
y <- log(1 + 10*x) + e

# Number of cosine basis functions
nbasis <- 50

# Fit the model with default priors and mcmc parameters
fout1 <- bsaqdpm(y ~ fs(x), p = 0.25, nbasis = nbasis,
                 shape = 'IncreasingConcave')
fout2 <- bsaqdpm(y ~ fs(x), p = 0.5, nbasis = nbasis,
                 shape = 'IncreasingConcave')
fout3 <- bsaqdpm(y ~ fs(x), p = 0.75, nbasis = nbasis,
                 shape = 'IncreasingConcave')

# fitted values
fit1 <- fitted(fout1)
fit2 <- fitted(fout2)
fit3 <- fitted(fout3)

# plots
plot(x, y, lwd = 2, xlab = 'x', ylab = 'y')
lines(fit1$xgrid, fit1$wbeta$mean[1] + fit1$fxgrid$mean, lwd=2, col=2)
lines(fit2$xgrid, fit2$wbeta$mean[1] + fit2$fxgrid$mean, lwd=2, col=3)
lines(fit3$xgrid, fit3$wbeta$mean[1] + fit3$fxgrid$mean, lwd=2, col=4)
legend('topleft',legend=c('1st Quartile','2nd Quartile','3rd Quartile'),
       lwd=2, col=2:4, lty=1)


## End(Not run)
```

---

bsar *Bayesian Shape-Restricted Spectral Analysis Regression*

Description

This function fits a Bayesian semiparametric regression model to estimate shape-restricted functions using a spectral analysis of Gaussian process priors.

Usage

```
bsar(formula, xmin, xmax, nbasis, nint, mcmc = list(), prior = list(),
shape = c('Free', 'Increasing', 'Decreasing', 'IncreasingConvex', 'DecreasingConcave',
'IncreasingConcave', 'DecreasingConvex', 'IncreasingS', 'DecreasingS',
'IncreasingRotatedS','DecreasingRotatedS','InvertedU','Ushape',
'IncMultExtreme','DecMultExtreme'), nExtreme = NULL,
marginal.likelihood = TRUE, spm.adequacy = FALSE, verbose = FALSE)
```

Arguments

`formula`	an object of class "`formula`"
`xmin`	a vector or scalar giving user-specific minimum values of x. The default values are minimum values of x.
`xmax`	a vector or scalar giving user-specific maximum values of x. The default values are maximum values of x.
`nbasis`	number of cosine basis functions.
`nint`	number of grid points where the unknown function is evaluated for plotting. The default is 200.
`mcmc`	a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): `nblow0 (1000)` giving the number of initialization period for adaptive metropolis, `maxmodmet (5)` giving the maximum number of times to modify metropolis, `nblow (10000)` giving the number of MCMC in transition period, `nskip (10)` giving the thinning interval, `smcmc (1000)` giving the number of MCMC for analysis, and `ndisp (1000)` giving the number of saved draws to be displayed on screen (the function reports on the screen when every `ndisp` iterations have been carried out).
`prior`	a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): `iflagprior` choosing a smoothing prior for spectral coefficients (iflagprior=0 assigns T-Smoother prior (default), iflagprior=1 chooses Lasso-Smoother prior), `theta0_m0` and `theta0_s0` giving the hyperparameters for prior distribution of the spectral coefficients (`theta0_m0` and `theta0_s0` are used when the functions have shape-restriction), `tau2_m0`, `tau2_s0` and `w0` giving the prior mean and standard deviation of smoothing prior (When iflagprior=1, tau2_m0 is only used as the hyperparameter), `beta_m0` and `beta_v0` giving the hyperparameters of the multivariate normal distribution for parametric part including intercept, `sigma2_m0` and `sigma2_v0` giving the prior mean and variance of the inverse gamma prior for the scale parameter of response, `alpha_m0` and `alpha_s0` giving the prior mean and standard deviation of the truncated normal prior distribution for the constant of integration, `iflagpsi` determining the prior of slope for logisitic function in `S` or `U` shaped (iflagpsi=1 (default), slope ψ is sampled and iflagpsi=0, ψ is fixed), `psifixed` giving initial value (iflagpsi=1) or fixed value (iflagpsi=0) of slope, `omega_m0` and `omega_s0` giving the prior mean and standard deviation of the truncated normal prior distribution for the inflection point of `S` or `U` shaped function.
`shape`	a vector giving types of shape restriction.

`nExtreme` a vector of extreme points for 'IncMultExtreme', 'DecMultExtreme' shape restrictions.

`marginal.likelihood`
a logical variable indicating whether the log marginal likelihood is calculated. The methods of Gelfand and Dey (1994) and Newton and Raftery (1994) are used.

`spm.adequacy` a logical variable indicating whether the log marginal likelihood of linear model is calculated. The marginal likelihood gives the values of the linear regression model excluding the nonlinear parts.

`verbose` a logical variable. If `TRUE`, the iteration number and the Metropolis acceptance rate are printed to the screen.

Details

This generic function fits a Bayesian spectral analysis regression model (Lenk and Choi, 2015) for estimating shape-restricted functions using Gaussian process priors. For enforcing shape-restrictions, they assumed that the derivatives of the functions are squares of Gaussian processes.

Let y_i and w_i be the response and the vector of parametric predictors, respectively. Further, let $x_{i,k}$ be the covariate related to the response through an unknown shape-restricted function. The model for estimating shape-restricted functions is as follows.

$$y_i = w_i^T\beta + \sum_{k=1}^{K} f_k(x_{i,k}) + \epsilon_i,\ i = 1, \ldots, n,$$

where f_k is an unknown shape-restricted function of the scalar $x_{i,k} \in [0,1]$ and the error terms $\{\epsilon_i\}$ are a random sample from a normal distribution, $N(0, \sigma^2)$.

The prior of function without shape restriction is:

$$f(x) = Z(x),$$

where Z is a second-order Gaussian process with mean function equal to zero and covariance function $\nu(s,t) = E[Z(s)Z(t)]$ for $s,t \in [0,1]$. The Gaussian process is expressed with the spectral representation based on cosine basis functions:

$$Z(x) = \sum_{j=0}^{\infty} \theta_j \varphi_j(x)$$

$$\varphi_0(x) = 1 \text{ and } \varphi_j(x) = \sqrt{2}\cos(\pi j x),\ j \geq 1,\ 0 \leq x \leq 1$$

The shape-restricted functions are modeled by assuming the qth derivatives of f are squares of Gaussian processes:

$$f^{(q)}(x) = \delta Z^2(x) h(x),\ \ \delta \in \{1, -1\},\ \ q \in \{1, 2\},$$

where h is the squish function. For monotonic, monotonic convex, and concave functions, $h(x) = 1$, while for `S` and `U` shaped functions, h is defined by

$$h(x) = \frac{1 - \exp[\psi(x-\omega)]}{1 + \exp[\psi(x-\omega)]},\ \ \psi > 0,\ \ 0 < \omega < 1$$

For the spectral coefficients of functions without shape constraints, the scale-invariant prior is used (The intercept is included in β):

$$\theta_j|\sigma,\tau,\gamma \sim N(0, \sigma^2\tau^2 \exp[-j\gamma]),\ j \geq 1$$

The priors for the spectral coefficients of shape restricted functions are:

$$\theta_0|\sigma \sim N(m_{\theta_0}, \sigma v_{\theta_0}^2), \quad \theta_j|\sigma,\tau,\gamma \sim N(m_{\theta_j}, \sigma\tau^2 \exp[-j\gamma]),\ j \geq 1$$

To complete the model specification, the conjugate priors are assumed for β and σ:

$$\beta|\sigma \sim N(m_{0,\beta}, \sigma^2 V_{0,\beta}), \quad \sigma^2 \sim IG\left(\frac{r_{0,\sigma}}{2}, \frac{s_{0,\sigma}}{2}\right)$$

Value

An object of class `bsam` representing the Bayesian spectral analysis model fit. Generic functions such as `print`, `fitted` and `plot` have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws`, the posterior samples of the fitted values are stored in the list `fit.draws`, and the MCMC samples for the log marginal likelihood are saved in the list `loglik.draws`. The output list also includes the following objects:

- `post.est` posterior estimates for all parameters in the model.
- `lmarg.lm` log marginal likelihood for linear regression model.
- `lmarg.gd` log marginal likelihood using Gelfand-Dey method.
- `lmarg.nr` log marginal likelihood using Netwon-Raftery method, which is biased.
- `rsquarey` correlation between y and $\hat{y}$.
- `call` the matched call.
- `mcmctime` running time of Markov chain from `system.time()`.

References

Jo, S., Choi, T., Park, B. and Lenk, P. (2019). bsamGP: An R Package for Bayesian Spectral Analysis Models Using Gaussian Process Priors. *Journal of Statistical Software*, **90**, 310-320.

Lenk, P. and Choi, T. (2017) Bayesian Analysis of Shape-Restricted Functions using Gaussian Process Priors. *Statistica Sinica*, **27**, 43-69.

Gelfand, A. E. and Dey, K. K. (1994) Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B - Statistical Methodology*, **56**, 501-514.

Newton, M. A. and Raftery, A. E. (1994) Approximate Bayesian inference with the weighted likelihood bootstrap (with discussion). *Journal of the Royal Statistical Society. Series B - Statistical Methodology*, **56**, 3-48.

See Also

`bsardpm`

Examples

```
## Not run:

###########################################
# Increasing Convex to Concave (S-shape) #
###########################################

# simulate data
f <- function(x) 5*exp(-10*(x - 1)^4) + 5*x^2

set.seed(1)

n <- 100
x <- runif(n)
y <- f(x) + rnorm(n, sd = 1)

# Number of cosine basis functions
nbasis <- 50

# Fit the model with default priors and mcmc parameters
fout <- bsar(y ~ fs(x), nbasis = nbasis, shape = 'IncreasingConvex',
             spm.adequacy = TRUE)

# Summary
print(fout); summary(fout)

# Trace plots
plot(fout)

# fitted values
fit <- fitted(fout)

# Plot
plot(fit, ask = TRUE)

##############################################
# Additive Model                             #
# Monotone-Increasing and Increasing-Convex #
##############################################

# Simulate data
f1 <- function(x) 2*pi*x + sin(2*pi*x)
f2 <- function(x) exp(6*x - 3)

n <- 200
x1 <- runif(n)
x2 <- runif(n)
x <- cbind(x1, x2)

y <- 5 + f1(x1) + f2(x2) + rnorm(n, sd = 0.5)
```

```
# Number of cosine basis functions
nbasis <- 50

# MCMC parameters
mcmc <- list(nblow0 = 1000, nblow = 10000, nskip = 10,
             smcmc = 5000, ndisp = 1000, maxmodmet = 10)

# Prior information
xmin <- apply(x, 2, min)
xmax <- apply(x, 2, max)
xrange <- xmax - xmin
prior <- list(iflagprior = 0, theta0_m0 = 0, theta0_s0 = 100,
              tau2_m0 = 1, tau2_v0 = 100, w0 = 2,
              beta_m0 = numeric(1), beta_v0 = diag(100,1),
              sigma2_m0 = 1, sigma2_v0 = 1000,
              alpha_m0 = 3, alpha_s0 = 50, iflagpsi = 1,
              psifixed = 1000, omega_m0 = (xmin + xmax)/2,
              omega_s0 = (xrange)/8)

# Fit the model with user specific priors and mcmc parameters
fout <- bsar(y ~ fs(x1) + fs(x2), nbasis = nbasis, mcmc = mcmc, prior = prior,
             shape = c('Increasing', 'IncreasingS'))

# Summary
print(fout); summary(fout)

## End(Not run)
```

---

bsarBig *Bayesian Spectral Analysis Regression for Big data*

Description

This function fits a Bayesian spectral analysis regression model for Big data.

Usage

```
bsarBig(formula, nbasis, nint, mcmc = list(), prior = list(), verbose = FALSE)
```

Arguments

formula	an object of class "formula"
nbasis	number of cosine basis functions.
nint	number of grid points where the unknown function is evaluated for plotting. The default is 500.
mcmc	a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): nblow (10000) giving the number of MCMC in transition period, nskip (10) giving the thinning interval, smcmc (1000) giving the number of MCMC for analysis, and ndisp (1000) giving the number of

	saved draws to be displayed on screen (the function reports on the screen when every `ndisp` iterations have been carried out).
`prior`	a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): `sigma2_m0` and `sigma2_v0` giving the prior mean and variance of the inverse gamma prior for the scale parameter of response, `tau2_m0`, `tau2_s0` and `w0` giving the prior mean and standard deviation of smoothing prior.
`verbose`	a logical variable. If `TRUE`, the iteration number and the Metropolis acceptance rate are printed to the screen.

Value

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws` and the posterior samples of the fitted values are stored in the list `fit.draws`. The output list also includes the following objects:

`post.est`	posterior estimates for all parameters in the model.
`call`	the matched call.
`mcmctime`	running time of Markov chain from `system.time()`.

See Also

`bsar`

Examples

```
# Ttrue function
ftrue <- function(x){
  ft <- 7*exp(-3*x) + 2*exp(-70*(x-.6)^2) - 2 + 5*x
  return(ft)
}

# Generate data
set.seed(1)

nobs <- 100000 # Number of observations
sigmat <- .5 # True sigma
nxgrid <- 500 # number of grid points: approximate likelihood & plots

xdata <- runif(nobs) # Generate x values
fobst <- ftrue(xdata) # True f at observations
ydata <- fobst + sigmat*rnorm(nobs)

# Compute grid on 0 to 1
xdelta <- 1/nxgrid
xgrid <- seq(xdelta/2, 1-xdelta/2, xdelta)
xgrid <- matrix(xgrid,nxgrid)
fxgridt <- ftrue(xgrid) # True f on xgrid

# Fit data
```

```
fout <- bsarBig(ydata ~ xdata, nbasis = 50, nint = nxgrid, verbose = TRUE)

# Plots
smcmc <- fout$mcmc$smcmc
t <- 1:smcmc
par(mfrow=c(2,2))
matplot(t, fout$mcmc.draws$theta, type = "l", main = "Theta", xlab = "Iteration", ylab = "Draw")
plot(t, fout$mcmc.draws$sigma, type = "l", main = "Sigma", xlab = "Iteration", ylab = "Draw")
matplot(t, fout$mcmc.draws$tau, type = "l", main = "Tau", xlab = "Iteration", ylab = "Draw")
matplot(t, fout$mcmc.draws$gamma, type = "l", main = "Gamma", xlab = "Iteration", ylab = "Draw")

dev.new()
matplot(fout$fit.draws$xgrid, cbind(fxgridt, fout$post.est$fhatm, fout$post.est$fhatq),
        type = "l", main = "Regression Function", xlab = "X", ylab = "Y")

# Compute RMISE for regression function
sse <- (fout$post.est$fhatm - fxgridt)^2
rmise <- intgrat(sse, 1/nxgrid)
rmise <- sqrt(rmise)
rmise
```

`bsardpm` *Bayesian Shape-Restricted Spectral Analysis Regression with Dirichlet Process Mixture Errors*

Description

This function fits a Bayesian semiparametric regression model to estimate shape-restricted functions using a spectral analysis of Gaussian process priors. The model assumes that the errors follow a Dirichlet process mixture model.

Usage

```
bsardpm(formula, xmin, xmax, nbasis, nint,
mcmc = list(), prior = list(), egrid, ngrid, location = TRUE,
shape = c('Free', 'Increasing', 'Decreasing', 'IncreasingConvex', 'DecreasingConcave',
'IncreasingConcave', 'DecreasingConvex', 'IncreasingS', 'DecreasingS',
'IncreasingRotatedS','DecreasingRotatedS','InvertedU','Ushape'),
verbose = FALSE)
```

Arguments

`formula`	an object of class "`formula`"
`xmin`	a vector or scalar giving user-specific minimum values of x. The default values are minimum values of x.
`xmax`	a vector or scalar giving user-specific maximum values of x. The default values are maximum values of x.
`nbasis`	number of cosine basis functions.

nint	number of grid points where the unknown function is evaluated for plotting. The default is 200.
mcmc	a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): nblow0 (1000) giving the number of initialization period for adaptive metropolis, maxmodmet (5) giving the maximum number of times to modify metropolis, nblow (10000) giving the number of MCMC in transition period, nskip (10) giving the thinning interval, smcmc (1000) giving the number of MCMC for analysis, and ndisp (1000) giving the number of saved draws to be displayed on screen (the function reports on the screen when every ndisp iterations have been carried out).
prior	a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): iflagprior choosing a smoothing prior for spectral coefficients (iflagprior=0 assigns T-Smoother prior (default), iflagprior=1 chooses Lasso-Smoother prior), theta0_m0 and theta0_s0 giving the hyperparameters for prior distribution of the spectral coefficients (theta0_m0 and theta0_s0 are used when the functions have shape-restriction), tau2_m0, tau2_s0 and w0 giving the prior mean and standard deviation of smoothing prior (When iflagprior=1, tau2_m0 is only used as the hyperparameter), beta_m0 and beta_v0 giving the hyperparameters of the multivariate normal distribution for parametric part including intercept, sigma2_m0 and sigma2_v0 giving the prior mean and variance of the inverse gamma prior for the scale parameter of response, alpha_m0 and alpha_s0 giving the prior mean and standard deviation of the truncated normal prior distribution for the constant of integration, iflagpsi determining the prior of slope for logisitic function in S or U shaped (iflagpsi=1 (default), slope ψ is sampled and iflagpsi=0, ψ is fixed), psifixed giving initial value (iflagpsi=1) or fixed value (iflagpsi=0) of slope, omega_m0 and omega_s0 giving the prior mean and standard deviation of the truncated normal prior distribution for the inflection point of S or U shaped function.
egrid	a vector giving grid points where the residual density estimate is evaluated. The default range is from -10 to 10.
ngrid	a vector giving number of grid points where the residual density estimate is evaluated. The default value is 500.
location	a logical value. If it is true, error density is modelled using location-scale mixture.
shape	a vector giving types of shape restriction.
verbose	a logical variable. If TRUE, the iteration number and the Metropolis acceptance rate are printed to the screen.

Details

This generic function fits a Bayesian spectral analysis regression model for estimating shape-restricted functions using Gaussian process priors. For enforcing shape-restrictions, the model assumes that the derivatives of the functions are squares of Gaussian processes. The model also assumes that the errors follow a Dirichlet process mixture model.

Let y_i and w_i be the response and the vector of parametric predictors, respectively. Further, let $x_{i,k}$ be the covariate related to the response through an unknown shape-restricted function. The model for estimating shape-restricted functions is as follows.

$$y_i = w_i^T\beta + \sum_{k=1}^{K} f_k(x_{i,k}) + \epsilon_i,\ i = 1, \ldots, n,$$

where f_k is an unknown shape-restricted function of the scalar $x_{i,k} \in [0, 1]$ and the error terms $\{\epsilon_i\}$ are a random sample from a Dirichlet process mixture model,

1. scale mixture :

$$\epsilon_i \sim f(\epsilon) = \int N(\epsilon; 0, \sigma^2) dG(\sigma^2),$$

$$G \sim DP(M, G0),\ \ G0 = Ga\left(\sigma^{-2}; \frac{r_{0,\sigma}}{2}, \frac{s_{0,\sigma}}{2}\right).$$

2. location-scale mixture :

$$\epsilon_i \sim f(\epsilon) = \int N(\epsilon; \mu, \sigma^2) dG(\mu, \sigma^2),$$

$$G \sim DP(M, G0),\ \ G0 = N\left(\mu; \mu_0, \kappa\sigma^2\right) Ga\left(\sigma^{-2}; \frac{r_{0,\sigma}}{2}, \frac{s_{0,\sigma}}{2}\right).$$

The prior of function without shape restriction is:

$$f(x) = Z(x),$$

where Z is a second-order Gaussian process with mean function equal to zero and covariance function $\nu(s, t) = E[Z(s)Z(t)]$ for $s, t \in [0, 1]$. The Gaussian process is expressed with the spectral representation based on cosine basis functions:

$$Z(x) = \sum_{j=0}^{\infty} \theta_j \varphi_j(x)$$

$$\varphi_0(x) = 1 \ \text{ and } \ \varphi_j(x) = \sqrt{2}\cos(\pi j x),\ j \geq 1,\ 0 \leq x \leq 1$$

The shape-restricted functions are modeled by assuming the qth derivatives of f are squares of Gaussian processes:

$$f^{(q)}(x) = \delta Z^2(x) h(x),\ \ \delta \in \{1, -1\},\ \ q \in \{1, 2\},$$

where h is the squish function. For monotonic, monotonic convex, and concave functions, $h(x) = 1$, while for S and U shaped functions, h is defined by

$$h(x) = \frac{1 - \exp[\psi(x - \omega)]}{1 + \exp[\psi(x - \omega)]},\ \ \psi > 0,\ \ 0 < \omega < 1$$

For the spectral coefficients of functions without shape constraints, the scale-invariant prior is used (The intercept is included in β):

$$\theta_j | \tau, \gamma \sim N(0, \tau^2 \exp[-j\gamma]),\ j \geq 1$$

The priors for the spectral coefficients of shape restricted functions are:

$$\theta_0 \sim N(m_{\theta_0}, v_{\theta_0}^2), \quad \theta_j|\tau,\gamma \sim N(m_{\theta_j}, \tau^2 \exp[-j\gamma]),\ j \geq 1$$

To complete the model specification, the popular normal prior is assumed for β:

$$\beta| \sim N(m_{0,\beta}, V_{0,\beta})$$

Value

An object of class `bsam` representing the Bayesian spectral analysis model fit. Generic functions such as `print`, `fitted` and `plot` have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws`, the posterior samples of the fitted values are stored in the list `fit.draws`, and the MCMC samples for the log marginal likelihood are saved in the list `loglik.draws`. The output list also includes the following objects:

`post.est`	posterior estimates for all parameters in the model.
`lpml`	log pseudo marginal likelihood using Mukhopadhyay and Gelfand method.
`imodmet`	the number of times to modify Metropolis.
`pmet`	proportion of θ accepted after burn-in.
`call`	the matched call.
`mcmctime`	running time of Markov chain from `system.time()`.

References

Jo, S., Choi, T., Park, B. and Lenk, P. (2019). bsamGP: An R Package for Bayesian Spectral Analysis Models Using Gaussian Process Priors. *Journal of Statistical Software*, **90**, 310-320.

Lenk, P. and Choi, T. (2017) Bayesian Analysis of Shape-Restricted Functions using Gaussian Process Priors. *Statistica Sinica*, **27**, 43-69.

MacEachern, S. N. and Müller, P. (1998) Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, **7**, 223-238.

Mukhopadhyay, S. and Gelfand, A. E. (1997) Dirichlet process mixed generalized linear models. *Journal of the American Statistical Association*, **92**, 633-639.

Neal, R. M. (2000) Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, **9**, 249-265.

See Also

`bsar`, `bsaqdpm`

Examples

```
## Not run:
#####################
# Increasing-convex #
#####################

# Simulate data
set.seed(1)

n <- 200
x <- runif(n)
e <- c(rnorm(n/2, sd = 0.5), rnorm(n/2, sd = 3))
y <- exp(6*x - 3) + e

# Number of cosine basis functions
nbasis <- 50

# Fit the model with default priors and mcmc parameters
fout <- bsardpm(y ~ fs(x), nbasis = nbasis, shape = 'IncreasingConvex')

# Summary
print(fout); summary(fout)

# fitted values
fit <- fitted(fout)

# Plot
plot(fit, ask = TRUE)


## End(Not run)
```

---

cadmium — *Cadmium dose-response meta data*

Description

This dataset includes minimal information of NCC-2012 meta data.

Usage

```
data("cadmium")
```

Format

A data frame with 190 observations on the following 5 variables.

`gender` a numeric vector with 1 : Female, 0 : Male, 0.5 : Unknown or both

`ethnicity` a integer vector with 1 : Asian, 2 : Caucasian

`Ucd_GM` a numeric vector of Geometric means of urinary cadmium

`b2_GM` a numeric vector of Geometric means of Beta2-Microglobulin

`isOld` a logical vector whether the observation is older than 50

References

Lee, Minjea, Choi, Taeryon; Kim, Jeongseon; Woo, Hae Dong (2013) Bayesian Analysis of Dose-Effect Relationship of Cadmium for Benchmark Dose Evaluation. *Korean Journal of Applied Statistics*, **26**(3), 453–470.

Examples

```
## Not run:
data(cadmium)

## End(Not run)
```

`Elec.demand` *Electricity demand data*

Description

The Elec.demand data consists of 288 quarterly observations in Ontario from 1971 to 1994.

Usage

```
data(Elec.demand)
```

Format

A data frame with 288 observations on the following 7 variables.

quarter date (yyyy-mm) from 1971 to 1994

enerm electricity demand.

gdp gross domestic product.

pelec price of electricity.

pgas price of natural gas.

hddqm the number of heating degree days relative to a reference temperature.

cddqm the number of cooling degree days relative to a reference temperature.

Source

Yatchew, A. (2003). *Semiparametric Regression for the Applied Econometrician*. Cambridge University Press.

References

Engle, R. F., Granger, C. W. J., Rice, J. and Weiss, A. (1986). Semiparametric estimates of the relation between weather and electricity sales. *Journal of the American Statistical Association*, **81**, 310-320.

Lenk, P. and Choi, T. (2017). Bayesian analysis of shape-restricted functions using Gaussian process priors. *Statistica Sinica*, **27**, 43-69.

Examples

```
## Not run:
data(Elec.demand)
plot(Elec.demand)

## End(Not run)
```

---

fitted.blm *Compute fitted values for a blm object*

Description

Computes pointwise posterior means and 95% credible intervals of the fitted Bayesian linear models.

Usage

```
## S3 method for class 'blm'
fitted(object, alpha = 0.05, HPD = TRUE, ...)
```

Arguments

`object`	a `bsam` object
`alpha`	a numeric scalar in the interval (0,1) giving the $100(1-\alpha)\%$ credible intervals.
`HPD`	a logical variable indicating whether the $100(1-\alpha)\%$ Highest Posterior Density (HPD) intervals are calculated. If `HPD`=FALSE, the $100(1-\alpha)\%$ equal-tail credible intervals are calculated. The default is TRUE.
`...`	not used

Details

None.

Value

A list containing posterior means and 95% credible intervals.

The output list includes the following objects:

`wbeta`	posterior estimates for regression function.
`yhat`	posterior estimates for generalised regression function.

References

Chen, M., Shao, Q. and Ibrahim, J. (2000) *Monte Carlo Methods in Bayesian computation*. Springer-Verlag New York, Inc.

See Also

blq, blr, gblr

Examples

```
## See examples for blq and blr
```

---

fitted.bsad *Compute fitted values for a bsad object*

Description

Computes pointwise posterior means and $100(1-\alpha)\%$ credible intervals of the fitted Bayesian spectral analysis density estimation model.

Usage

```
## S3 method for class 'bsad'
fitted(object, alpha = 0.05, HPD = TRUE, ...)
```

Arguments

object	a bsad object
alpha	a numeric scalar in the interval (0,1) giving the $100(1-\alpha)\%$ credible intervals.
HPD	a logical variable indicating whether the $100(1-\alpha)\%$ Highest Posterior Density (HPD) intervals are calculated. If HPD=FALSE, the $100(1-\alpha)\%$ equal-tail credible intervals are calculated. The default is TRUE.
...	not used

Details

None.

Value

A list object of class `fitted.bsad` containing posterior means and $100(1-\alpha)\%$ credible intervals. Generic function `plot` displays the results of the fit.

The output list includes the following objects:

fpar	posterior estimates for parametric model.
fsemi	posterior estimates for semiparametric model.
fsemiMaxKappa	posterior estimates for semiparametric model with maximum number of basis.

See Also

bsad

Examples

```
## See examples for bsad
```

---

fitted.bsam *Compute fitted values for a bsam object*

Description

Computes pointwise posterior means and $100(1-\alpha)\%$ credible intervals of the fitted Bayesian spectral analysis models.

Usage

```
## S3 method for class 'bsam'
fitted(object, alpha = 0.05, HPD = TRUE, ...)
```

Arguments

object	a bsam object
alpha	a numeric scalar in the interval (0,1) giving the $100(1-\alpha)\%$ credible intervals.
HPD	a logical variable indicating whether the $100(1-\alpha)\%$ Highest Posterior Density (HPD) intervals are calculated. If HPD=FALSE, the $100(1-\alpha)\%$ equal-tail credible intervals are calculated. The default is TRUE.
...	not used

Details

None.

Value

A list object of class fitted.bsam containing posterior means and $100(1-\alpha)\%$ credible intervals. Generic function plot displays the results of the fit.

The output list includes the following objects:

fxobs	posterior estimates for unknown functions over observation.
fxgrid	posterior estimates for unknown functions over grid points.
wbeta	posterior estimates for parametric part.
yhat	posterior estimates for fitted values of response. For gbsar, it gives posterior estimates for expectation of response.

See Also

bsaq, bsaqdpm, bsar, bsardpm

Examples

```
## See examples for bsaq, bsaqdpm, bsar, and bsardpm
```

---

fitted.bsamdpm — *Compute fitted values for a bsamdpm object*

Description

Computes pointwise posterior means and $100(1-\alpha)\%$ credible intervals of the fitted Bayesian spectral analysis models with Dirichlet process mixture error.

Usage

```
## S3 method for class 'bsamdpm'
fitted(object, alpha = 0.05, HPD = TRUE, ...)
```

Arguments

`object`	a bsamdpm object
`alpha`	a numeric scalar in the interval (0,1) giving the $100(1-\alpha)\%$ credible intervals.
`HPD`	a logical variable indicating whether the $100(1-\alpha)\%$ Highest Posterior Density (HPD) intervals are calculated. If `HPD`=FALSE, the $100(1-\alpha)\%$ equal-tail credible intervals are calculated. The default is TRUE.
`...`	not used

Details

None.

Value

A list object of class `fitted.bsamdpm` containing posterior means and 95% credible intervals. Generic function `plot` displays the results of the fit.

The output list includes the following objects:

`edens`	posterior estimate for unknown error distribution over grid points.
`fxobs`	posterior estimates for unknown functions over observation.
`fxgrid`	posterior estimates for unknown functions over grid points.
`wbeta`	posterior estimates for parametric part.
`yhat`	posterior estimates for fitted values of response.

See Also

bsaqdpm, bsardpm

Examples

```
## See examples for bsaqdpm and bsardpm
```

---

fs *Specify a Fourier Basis Fit in a BSAM Formula*

Description

A symbolic wrapper to indicate a nonparametric term in a formula argument to bsaq, bsaqdpm, bsar, bsardpm, and gbsar.

Usage

```
fs(x)
```

Arguments

x	a vector of the univariate covariate for nonparametric component

Examples

```
## Not run:

# fit x using a Fourier basis
y ~ w + fs(x)

# fit x1 and x2 using a Fourier basis
y ~ fs(x1) + fs(x2)


## End(Not run)
```

---

gblr *Generalized Bayesian Linear Models*

Description

This function fits a Bayesian generalized linear regression model.

Usage

```
gblr(formula, data = NULL, family, link, mcmc = list(), prior = list(),
marginal.likelihood = TRUE, algorithm = c('AM', 'KS'), verbose = FALSE)
```

Arguments

`formula` an object of class "`formula`"

`data` an optional data frame.

`family` a description of the error distribution to be used in the model: The family contains bernoulli ("bernoulli"), poisson ("poisson"), negative-binomial ("negative.binomial"), poisson-gamma mixture ("poisson.gamma").

`link` a description of the link function to be used in the model.

`mcmc` a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): `nblow (10000)` giving the number of MCMC in transition period, `nskip (10)` giving the thinning interval, `smcmc (1000)` giving the number of MCMC for analysis, and `ndisp (1000)` giving the number of saved draws to be displayed on screen (the function reports on the screen when every `ndisp` iterations have been carried out).

`prior` a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): `beta_m0` and `beta_v0` giving the hyperparameters of the multivariate normal distribution for parametric part including intercept, `kappa_m0` and `kappa_v0` giving the prior mean and variance of the gammal prior distribution for dispersion parameter (negative-binomial).

`marginal.likelihood` a logical variable indicating whether the log marginal likelihood is calculated. The methods of Gelfand and Dey (1994) is used.

`algorithm` a description of the algorithm to be used in the fitting of the logistic model: The algorithm contains the Gibbs sampler based on the Kolmogorov-Smirnov distribution (`KS`) and an adaptive Metropolis algorithm (`AM`).

`verbose` a logical variable. If `TRUE`, the iteration number and the Metropolis acceptance rate are printed to the screen.

Details

This generic function fits a Bayesian generalized linear regression models.

Let y_i and w_i be the response and the vector of parametric predictors, respectively. The model is as follows.

$$y_i|\mu_i \sim F(\mu_i),$$
$$g(\mu_i) = w_i^T\beta,\ i = 1, \dots, n,$$

where $g(\cdot)$ is a link function and $F(\cdot)$ is a distribution of an exponential family.

For unknown coefficients, the following prior is assumed for β:

$$\beta \sim N(m_{0,\beta}, V_{0,\beta})$$

The prior for the dispersion parameter of negative-binomial regression is

$$\kappa \sim Ga(r_0, s_0)$$

Value

An object of class `blm` representing the generalized Bayesian linear model fit. Generic functions such as `print`, `fitted` and `plot` have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws`, the posterior samples of the fitted values are stored in the list `fit.draws`, and the MCMC samples for the log marginal likelihood are saved in the list `loglik.draws`. The output list also includes the following objects:

`post.est`	posterior estimates for all parameters in the model.
`lmarg`	log marginal likelihood using Gelfand-Dey method.
`family`	the family object used.
`link`	the link object used.
`methods`	the method object used in the logit model.
`call`	the matched call.
`mcmctime`	running time of Markov chain from `system.time()`.

References

Albert, J. H. and Chib, S. (1993) Bayesian Analysis of Binary and Polychotomous Response Data. *Journal of the American Statistical Association*, **88**, 669-679.

Holmes, C. C. and Held, L. (2006) Bayesian Auxiliary Variables Models for Binary and Multinomial Regression. *Bayesian Analysis*, **1**, 145-168.

Gelfand, A. E. and Dey, K. K. (1994) Bayesian Model Choice: Asymptotics and Exact Calculations. *Journal of the Royal Statistical Society. Series B - Statistical Methodology*, **56**, 501-514.

Roberts, G. O. and Rosenthal, J. S. (2009) Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics*, **18**, 349-367.

See Also

`blr`, `blq`

Examples

```
############################
# Poisson Regression Model #
############################

# Simulate data
set.seed(1)

n <- 100
x <- runif(n)
y <- rpois(n, exp(0.5 + x*0.4))

# Fit the model with default priors and mcmc parameters
fout <- gblr(y ~ x, family = 'poisson', link = 'log')
```

```
# Summary
print(fout); summary(fout)

# Plot
plot(fout)

# fitted values
fitf <- fitted(fout)
```

---

gbsar   *Bayesian Shape-Restricted Spectral Analysis for Generalized Partial Linear Models*

Description

This function fits a Bayesian generalized partial linear regression model to estimate shape-restricted functions using a spectral analysis of Gaussian process priors.

Usage

```
gbsar(formula, xmin, xmax, family, link, nbasis, nint, mcmc = list(), prior = list(),
shape = c('Free','Increasing','Decreasing','IncreasingConvex','DecreasingConcave',
          'IncreasingConcave','DecreasingConvex','IncreasingS','DecreasingS',
          'IncreasingRotatedS','DecreasingRotatedS','InvertedU','Ushape'),
marginal.likelihood = TRUE, algorithm = c('AM', 'KS'), verbose = FALSE)
```

Arguments

`formula`	an object of class "`formula`"
`xmin`	a vector or scalar giving user-specific minimum values of x. The default values are minimum values of x.
`xmax`	a vector or scalar giving user-specific maximum values of x. The default values are maximum values of x.
`family`	a description of the error distribution to be used in the model: The family contains bernoulli ("bernoulli"), poisson ("poisson"), negative-binomial ("negative.binomial"), poisson-gamma mixture ("poisson.gamma").
`link`	a description of the link function to be used in the model.
`nbasis`	number of cosine basis functions.
`nint`	number of grid points where the unknown function is evaluated for plotting. The default is 200.
`mcmc`	a list giving the MCMC parameters. The list includes the following integers (with default values in parentheses): `nblow0 (1000)` giving the number of initialization period for adaptive metropolis, `maxmodmet (5)` giving the maximum number of times to modify metropolis, `nblow (10000)` giving the number of MCMC in transition period, `nskip (10)` giving the thinning interval, `smcmc (1000)` giving the number of MCMC for analysis, and `ndisp (1000)` giving

the number of saved draws to be displayed on screen (the function reports on the screen when every `ndisp` iterations have been carried out).

`prior` a list giving the prior information. The list includes the following parameters (default values specify the non-informative prior): `iflagprior` choosing a smoothing prior for spectral coefficients (iflagprior=0 assigns T-Smoother prior (default), iflagprior=1 chooses Lasso-Smoother prior), `theta_m0`, `theta0_m0` and `theta0_s0` giving the hyperparameters for prior distribution of the spectral coefficients (`theta0_m0` and `theta0_s0` are used when the functions have shape-restriction), `tau2_m0`, `tau2_s0` and `w0` giving the prior mean and standard deviation of smoothing prior (When iflagprior=1, tau2_m0 is only used as the hyperparameter), `beta_m0` and `beta_v0` giving the hyperparameters of the multivariate normal distribution for parametric part including intercept, `alpha_m0` and `alpha_s0` giving the prior mean and standard deviation of the truncated normal prior distribution for the constant of integration, `iflagpsi` determining the prior of slope for logisitic function in `S` or `U` shaped (iflagpsi=1 (default), slope ψ is sampled and iflagpsi=0, ψ is fixed), `psifixed` giving initial value (iflagpsi=1) or fixed value (iflagpsi=0) of slope, `omega_m0` and `omega_s0` giving the prior mean and standard deviation of the truncated normal prior distribution for the inflection point of `S` or `U` shaped function, `kappa_m0` and `kappa_v0` giving the prior mean and variance of the gammal prior distribution for dispersion parameter (negative-binomial).

`shape` a vector giving types of shape restriction.

`marginal.likelihood`
a logical variable indicating whether the log marginal likelihood is calculated. The methods of Gelfand and Dey (1994) and Newton and Raftery (1994) are used.

`algorithm` a description of the algorithm to be used in the fitting of the logistic model: The algorithm contains the Gibbs sampler based on the Kolmogorov-Smirnov distribution (`KS`) and an adaptive Metropolis algorithm (`AM`).

`verbose` a logical variable. If `TRUE`, the iteration number and the Metropolis acceptance rate are printed to the screen.

Details

This generic function fits a Bayesian generalized partial linear regression models for estimating shape-restricted functions using Gaussian process priors. For enforcing shape-restrictions, they assumed that the derivatives of the functions are squares of Gaussian processes.

Let y_i and w_i be the response and the vector of parametric predictors, respectively. Further, let $x_{i,k}$ be the covariate related to the response through an unknown shape-restricted function. The model for estimating shape-restricted functions is as follows.

$$y_i|\mu_i \sim F(\mu_i),$$

$$g(\mu_i) = w_i^T \beta + \sum_{k=1}^{K} f_k(x_{i,k}),\ i = 1, \dots, n,$$

where $g(\cdot)$ is a link function and f_k is an unknown nonlinear function of the scalar $x_{i,k} \in [0, 1]$.

The prior of function without shape restriction is:

$$f(x) = Z(x),$$

where Z is a second-order Gaussian process with mean function equal to zero and covariance function $\nu(s,t) = E[Z(s)Z(t)]$ for $s, t \in [0,1]$. The Gaussian process is expressed with the spectral representation based on cosine basis functions:

$$Z(x) = \sum_{j=0}^{\infty} \theta_j \varphi_j(x)$$

$$\varphi_0(x) = 1 \text{ and } \varphi_j(x) = \sqrt{2}\cos(\pi j x),\ j \geq 1,\ 0 \leq x \leq 1$$

The shape-restricted functions are modeled by assuming the qth derivatives of f are squares of Gaussian processes:

$$f^{(q)}(x) = \delta Z^2(x) h(x),\ \ \delta \in \{1, -1\},\ \ q \in \{1, 2\},$$

where h is the squish function. For monotonic, monotonic convex, and concave functions, $h(x) = 1$, while for `S` and `U` shaped functions, h is defined by

$$h(x) = \frac{1 - \exp[\psi(x - \omega)]}{1 + \exp[\psi(x - \omega)]},\ \ \psi > 0,\ \ 0 < \omega < 1$$

For the spectral coefficients of functions without shape constraints, the following prior is used (The intercept is included in β):

$$\theta_j | \tau, \gamma \sim N(0, \tau^2 \exp[-j\gamma]),\ j \geq 1$$

The priors for the spectral coefficients of shape restricted functions are:

$$\theta_0 | \sim N(m_{\theta_0}, v_{\theta_0}^2), \quad \theta_j | \tau, \gamma \sim N(m_{\theta_j}, \tau^2 \exp[-j\gamma]),\ j \geq 1$$

To complete the model specification, the following prior is assumed for β:

$$\beta | \sim N(m_{0,\beta}, V_{0,\beta})$$

Value

An object of class `bsam` representing the Bayesian spectral analysis model fit. Generic functions such as `print`, `fitted` and `plot` have methods to show the results of the fit.

The MCMC samples of the parameters in the model are stored in the list `mcmc.draws`, the posterior samples of the fitted values are stored in the list `fit.draws`, and the MCMC samples for the log marginal likelihood are saved in the list `loglik.draws`. The output list also includes the following objects:

`post.est`	posterior estimates for all parameters in the model.
`lmarg.gd`	log marginal likelihood using Gelfand-Dey method.
`lmarg.nr`	log marginal likelihood using Netwon-Raftery method, which is biased.
`family`	the family object used.
`link`	the link object used.
`call`	the matched call.
`mcmctime`	running time of Markov chain from `system.time()`.

References

Jo, S., Choi, T., Park, B. and Lenk, P. (2019). bsamGP: An R Package for Bayesian Spectral Analysis Models Using Gaussian Process Priors. *Journal of Statistical Software*, **90**, 310-320.

Lenk, P. and Choi, T. (2017) Bayesian Analysis of Shape-Restricted Functions using Gaussian Process Priors. *Statistica Sinica*, **27**, 43-69.

Roberts, G. O. and Rosenthal, J. S. (2009) Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics*, **18**, 349-367.

Holmes, C. C. and Held, L. (2006) Bayesian Auxiliary Variables Models for Binary and Multinomial Regression. *Bayesian Analysis*, **1**, 145-168.

Gelfand, A. E. and Dey, K. K. (1994) Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B - Statistical Methodology*, **56**, 501-514.

Newton, M. A. and Raftery, A. E. (1994) Approximate Bayesian inference with the weighted likelihood bootstrap (with discussion). *Journal of the Royal Statistical Society. Series B - Statistical Methodology*, **56**, 3-48.

Albert, J. H. and Chib, S. (1993) Bayesian Analysis of Binary and Polychotomous Response Data. *Journal of the American Statistical Association*, **88**, 669-679.

See Also

bsaq, bsar

Examples

```
## Not run:
###########################
# Probit Regression Model #
###########################

# Simulate data
set.seed(1)

f <- function(x) 1.5 * sin(pi * x)

n <- 1000
b <- c(1,-1)
rho <- 0.7
u  <- runif(n, min = -1, max = 1)
x  <- runif(n, min = -1, max = 1)
w1 <- runif(n, min = -1, max = 1)
w2 <- round(f(rho * x + (1 - rho) * u))
w  <- cbind(w1, w2)

y  <- w %*% b + f(x) + rnorm(n)
y <- (y > 0)

# Number of cosine basis functions
nbasis <- 50

# Fit the model with default priors and mcmc parameters
```

```
fout <- gbsar(y ~ w1 + w2 + fs(x), family = "bernoulli", link = "probit",
              nbasis = nbasis, shape = 'Free')

# Summary
print(fout); summary(fout)

# fitted values
fit <- fitted(fout)

# Plot
plot(fit, ask = TRUE)

######################################
# Logistic Additive Regression Model #
######################################

# Wage-Union data
data(wage.union); attach(wage.union)

race[race==1 | race==2]=0
race[race==3]=1

y <- union
w <- cbind(race,sex,south)
x <- cbind(wage,education,age)

# mcmc parameters
mcmc <- list(nblow0 = 10000,
             nblow = 10000,
             nskip = 10,
             smcmc = 1000,
             ndisp = 1000,
             maxmodmet = 10)

foutGBSAR <- gbsar(y ~ race + sex + south + fs(wage) + fs(education) + fs(age),
                   family = 'bernoulli', link = 'logit', nbasis = 50, mcmc = mcmc,
                   shape = c('Free','Decreasing','Increasing'))

# fitted values
fitGBSAR <- fitted(foutGBSAR)

# Plot
plot(fitGBSAR, ask = TRUE)


## End(Not run)
```

intgrat *Numerical integration using a simple Trapezoidal rule*

Description

Trapezoidal rule is a technique for approximating the definite integral.

Usage

```
intgrat(f, delta)
```

Arguments

`f`	Function values to be integrated.
`delta`	Spacing size.

Value

`intgrat` returns the value of the intergral.

`intsim` *Numerical integration using Simpson's rule*

Description

Simpson's rule is a method for numerical integration.

Usage

```
intsim(f, delta)
```

Arguments

`f`	Function values to be integrated.
`delta`	Spacing size.

Value

`intsim` returns the value of the intergral.

London.Mortality *Daily Moratlity in London*

Description

The London.Mortality data consists of daily death occurrences from Jan. 1st, 1993 to Dec. 31st, 2006 and corresponding weather observations including temperature and humidity in London.

Usage

```
data(London.Mortality)
```

Format

A data frame with 5113 observations on the following 7 variables.

date date in YYYY-MM-DD.

tmean Mean temperature.

tmin Minimum dry-bulb temperature.

tmax Maximum dry-bulb temperature.

dewp Dew point.

rh Relative humidity.

death the number of death occurences.

Source

Office for National Statistics

British Atmospheric Data Centre

https://github.com/gasparrini/2015_gasparrini_Lancet_Rcodedata

References

Armstrong BG, Chalabi Z, Fenn B, Hajat S, Kovats S, Milojevic A, Wilkinson P (2011). Association of mortality with high temperatures in a temperate climate: England and Wales. *Journal of Epidemiology & Community Health*, **65**(4), 340–345.

Gasparrini A, Armstrong B, Kovats S, Wilkinson P (2012). The effect of high temperatures on cause-specific mortality in England and Wales. *Occupational and Environmental Medicine*, **69**(1), 56–61.

Gasparrini A, Guo Y, Hashizume M, Lavigne E, Zanobetti A, Schwartz J, Tobias A, Tong S, Rocklöv J, Forsberg B, et al.(2015). Mortality risk attributable to high and low ambient temperature: a multicountry observational study. *The Lancet*, **386**(9991), 369-375.

Examples

```
## Not run:
data(London.Mortality)

## End(Not run)
```

---

plasma *A Data Set for Plasma Levels of Retinol and Beta-Carotene*

Description

This data set contains 314 observations on 14 variables.

Usage

```
data(plasma)
```

Format

`age` Age (years).

`sex` Sex (1=Male, 2=Female).

`smoke` Smoking status (1=Never, 2=Former, 3=Current Smoker).

`vmi` BMI values (weight/(height^2)).

`vitas` Vitamin use (1=Yes,fairly often, 2=Yes, not often, 3=No).

`calories` Number of calories consumed per day.

`fat` Grams of fat consumed per day.

`fiber` Grams of fiber consumed per day.

`alcohol` Number of alcoholic drinks consumed per week.

`cholesterol` Cholesterol consumed (mg per day).

`beta diet` Dietary beta-carotene consumed (mcg per day).

`reedit` Dietary retinol consumed (mcg per day).

`betaplasma` Plasma beta-carotene (ng/ml).

`retplasma` Plasma Retinol (ng/ml).

Source

https://lib.stat.cmu.edu/datasets/Plasma_Retinol

References

Nierenberg, D. W., Stukel, T. A., Baron, J. A., Dain, B. J., and Greenberg, E. R. (1989). Determinants of plasma levels of beta-carotene and retinol. *American Journal of Epidemiology*, **130**, 511-521.

Meyer, M. C., Hackstadt, A. J., and Hoeting, J. A. (2011). Bayesian estimation and inference for generalized partial linear models using shape-restricted splines. *Journal of Nonparametric Statistics*, **23**(4), 867-884.

Examples

```
## Not run:
data(plasma)

## End(Not run)
```

`plot.blm` *Plot a blm object*

Description

Plots the posterior samples for Bayesian linear models

Usage

```
## S3 method for class 'blm'
plot(x, ...)
```

Arguments

`x`	a `blm` object
`...`	other options to pass to the plotting functions

Value

Returns a plot.

See Also

`blq`, `blr`

Examples

```
## See examples for blq and blr
```

plot.bsad *Plot a bsad object*

Description

Plots the posterior samples for Bayesian semiparametric density estimation using a logistic Gaussian process.

Usage

```
## S3 method for class 'bsad'
plot(x, ...)
```

Arguments

`x`	a `bsad` object
`...`	other options to pass to the plotting functions

Value

Returns a plot.

See Also

`bsad`

Examples

```
## See examples for bsad
```

plot.bsam *Plot a bsam object*

Description

Plots the posterior samples for Bayesian spectral analysis models.

Usage

```
## S3 method for class 'bsam'
plot(x, ...)
```

Arguments

`x`	a `bsam` object
`...`	other options to pass to the plotting functions

Value

Returns a plot.

See Also

bsaq, bsaqdpm, bsar, bsardpm

Examples

```
## See examples for bsaq, bsaqdpm, bsar, and bsardpm
```

---

plot.bsamdpm *Plot a bsamdpm object*

Description

Plots the posterior samples for Bayesian spectral analysis models with Dirichlet process mixture error.

Usage

```
## S3 method for class 'bsamdpm'
plot(x, ...)
```

Arguments

x	a bsamdpm object
...	other options to pass to the plotting functions

Value

Returns a plot.

See Also

bsaqdpm, bsardpm

Examples

```
## See examples for bsaqdpm and bsardpm
```

plot.fitted.bsad *Plot a fitted.bsad object*

Description

Plots the predictive density for Bayesian density estimation model using logistic Gaussian process

Usage

```
## S3 method for class 'fitted.bsad'
plot(x, ggplot2, legend.position, nbins, ...)
```

Arguments

`x`	a `fitted.bsad` object
`ggplot2`	a logical variable. If `TRUE` the `ggplot2` package is used.
`legend.position`	the position of legends (“none”, “left”, “right”, “bottom”, “top”). It is used when `ggplot2 = TRUE`.
`nbins`	Number of bins used. Default is 30.
`...`	other options to pass to the plotting functions

Value

Returns a plot.

See Also

`bsad`, `fitted.bsad`

Examples

```
## See example for bsad
```

plot.fitted.bsam *Plot a fitted.bsam object*

Description

Plots the data and the fit for Bayesian spectral analysis models.

Usage

```
## S3 method for class 'fitted.bsam'
plot(x, type, ask, ggplot2, legend.position, ...)
```

Arguments

`x`	a `fitted.bsam` object
`type`	the type of fitted plot. The default is on the scale of the response variable`type="response"`; the alternative `type="term"` is on the scale of the nonparametric predictor. Note that this affects only on glm type models. For example, binomial model with the default option gives the predicted probabilites.
`ask`	see. `par`
`ggplot2`	a logical variable. If `TRUE` the `ggplot2` package is used.
`legend.position`	the position of legends ("none", "left", "right", "bottom", "top"). It is used when `ggplot2 = TRUE`.
`...`	other options to pass to the plotting functions

Value

Returns a plot.

See Also

`bsaq`, `bsaqdpm`, `bsar`, `bsardpm`, `fitted.bsam`

Examples

```
## See examples for bsaq, bsaqdpm, bsar, and bsardpm
```

`plot.fitted.bsamdpm` *Plot a fitted.bsamdpm object*

Description

Plots the data and the fit for Bayesian spectral analysis models with Dirichlet process mixture error.

Usage

```
## S3 method for class 'fitted.bsamdpm'
plot(x, ask, ggplot2, legend.position, ...)
```

Arguments

`x`	a `fitted.bsamdpm` object
`ask`	see. `par`
`ggplot2`	a logical variable. If `TRUE` the `ggplot2` package is used.
`legend.position`	the position of legends ("none", "left", "right", "bottom", "top"). It is used when `ggplot2 = TRUE`.
`...`	other options to pass to the plotting functions

Value

Returns a plot.

See Also

bsaqdpm, bsardpm, fitted.bsamdpm

Examples

```
## See examples for bsaqdpm and bsardpm
```

---

predict.blm *Predict method for a blm object*

Description

Computes predicted values of Bayesian linear models.

Usage

```
## S3 method for class 'blm'
predict(object, newdata, alpha = 0.05, HPD = TRUE, ...)
```

Arguments

object	a bsam object
newdata	an optional data matrix or vector with which to predict. If omitted, the fitted values are returned.
alpha	a numeric scalar in the interval (0,1) giving the $100(1-\alpha)\%$ credible intervals.
HPD	a logical variable indicating whether the $100(1-\alpha)\%$ Highest Posterior Density (HPD) intervals are calculated. If HPD=FALSE, the $100(1-\alpha)\%$ equal-tail credible intervals are calculated. The default is TRUE.
...	not used

Details

None.

Value

A list containing posterior means and 95% credible intervals.

The output list includes the following objects:

wbeta	posterior estimates for regression function.
yhat	posterior estimates for generalised regression function.

References

Chen, M., Shao, Q. and Ibrahim, J. (2000) *Monte Carlo Methods in Bayesian computation*. Springer-Verlag New York, Inc.

See Also

blq, blr, gblr

Examples

```
## Not run:
#####################
# Simulated example #
#####################

# Simulate data
  set.seed(1)

  n <- 100
  w <- runif(n)
  y <- 3 + 2*w + rnorm(n, sd = 0.8)

  # Fit the model with default priors and mcmc parameters
  fout <- blr(y ~ w)

  # Predict
  new <- rnorm(n)
  predict(fout, newdata = new)

## End(Not run)
```

`predict.bsam` *Predict method for a bsam object*

Description

Computes the predicted values of Bayesian spectral analysis models.

Usage

```
## S3 method for class 'bsam'
predict(object, newp, newnp, alpha = 0.05, HPD = TRUE, type = "response", ...)
```

Arguments

`object`	a bsam object
`newp`	an optional data of parametric components with which to predict. If omitted, the fitted values are returned.

newnp	an optional data of nonparametric components with which to predict. If omitted, the fitted values are returned.
alpha	a numeric scalar in the interval (0,1) giving the $100(1-\alpha)\%$ credible intervals.
HPD	a logical variable indicating whether the $100(1-\alpha)\%$ Highest Posterior Density (HPD) intervals are calculated. If HPD=FALSE, the $100(1-\alpha)\%$ equal-tail credible intervals are calculated. The default is TRUE.
type	the type of prediction required. type = "response" gives the posterior predictive samples as default. The "mean" option returns expectation of the posterior estimates.
...	not used

Details

None.

Value

A list object of class predict.bsam containing posterior means and $100(1-\alpha)\%$ credible intervals.

The output list includes the following objects:

fxobs	posterior estimates for unknown functions over observation.
wbeta	posterior estimates for parametric part.
yhat	posterior estimates for fitted values of either response or expectation of response. For gbsar, it gives posterior estimates for expectation of response.
fxResid	posterior estimates for fitted parametric residuals. Not applicable for gbsar.

See Also

bsaq, bsar, gbsar

Examples

```
## Not run:

##########################################
# Increasing Convex to Concave (S-shape) #
##########################################

# simulate data
f <- function(x) 5*exp(-10*(x - 1)^4) + 5*x^2

set.seed(1)

n <- 100
x <- runif(n)
y <- f(x) + rnorm(n, sd = 1)

# Number of cosine basis functions
```

```
nbasis <- 50

# Fit the model with default priors and mcmc parameters
fout <- bsar(y ~ fs(x), nbasis = nbasis, shape = 'IncreasingConvex',
             spm.adequacy = TRUE)

# Prediction
xnew <- runif(n)
predict(fout, newnp = xnew)

## End(Not run)
```

`predict.bsamdpm` *Predict method for a bsamdpm object*

Description

Computes the predicted values of Bayesian spectral analysis models with Dirichlet process mixture errors.

Usage

```
## S3 method for class 'bsamdpm'
predict(object, newp, newnp, alpha = 0.05, HPD = TRUE, ...)
```

Arguments

`object`	a `bsamdpm` object
`newp`	an optional data of parametric components with which to predict. If omitted, the fitted values are returned.
`newnp`	an optional data of nonparametric components with which to predict. If omitted, the fitted values are returned.
`alpha`	a numeric scalar in the interval (0,1) giving the $100(1-\alpha)\%$ credible intervals.
`HPD`	a logical variable indicating whether the $100(1-\alpha)\%$ Highest Posterior Density (HPD) intervals are calculated. If `HPD`=FALSE, the $100(1-\alpha)\%$ equal-tail credible intervals are calculated. The default is TRUE.
`...`	not used

Details

None.

Value

A list object of class `predict.bsamdpm` containing posterior means and $100(1-\alpha)\%$ credible intervals.

The output list includes the following objects:

`fxobs`	posterior estimates for unknown functions over observation.
`wbeta`	posterior estimates for parametric part.
`yhat`	posterior estimates for fitted values of response.

See Also

`bsaqdpm`, `bsardpm`

Examples

```
## Not run:

#####################
# Increasing-convex #
#####################

# Simulate data
set.seed(1)

n <- 200
x <- runif(n)
e <- c(rnorm(n/2, sd = 0.5), rnorm(n/2, sd = 3))
y <- exp(6*x - 3) + e

# Number of cosine basis functions
nbasis <- 50

# Fit the model with default priors and mcmc parameters
fout <- bsardpm(y ~ fs(x), nbasis = nbasis, shape = 'IncreasingConvex')

# Prediction
xnew <- runif(n)
predict(fout, newnp = xnew)


## End(Not run)
```

`rald` *The asymmetric Laplace distribution*

Description

Density for and random values from a three-parameter asymmetric Laplace distribution.

Usage

```
rald(n, location=0, scale=1, p=0.5)
```

Arguments

`n`	Number of random values to be generated.
`location`	Location parameter.
`scale`	Scale parameter.
`p`	Skewness parameter.

Details

This generic function generates a random variable from an asymmetric Laplace distribution (ALD). The ALD has the following probability density function:

$$ALD_p(x; \mu, \sigma) = \frac{p(1-p)}{\sigma} \exp\Big(-\frac{(x-\mu)[p - I(x \leq \mu)]}{\sigma}\Big),$$

where $0 < p < 1$ is the skew parameter, $\sigma > 0$ is the scale parameter, $-\infty < \mu < \infty$ is the location parameter, and $I(\cdot)$ is the indication function. The range of x is $(-\infty, \infty)$.

Value

`rald` gives out a vector of random numbers generated by the asymmetric Laplace distribution.

References

Koenker, R. and Machado, J. (1999). Goodness of fit and related inference processes for quantile regression. *Journal of the American Statistical Association*, **94**(3), 1296-1309.

Yu, K. and Zhang, J. (2005). A Three-parameter asymmetric Laplace distribution and its extension. *Communications in Statistics - Theory and Methods*, **34**, 1867-1879.

`traffic` *Monthly traffic accidents data*

Description

This data set contains 108 observations on 6 variables.

Usage

```
data(traffic)
```

Format

`ln_number` logarithm of the number of monthly automobile accidents in the state of Michigan.

`month` months from January 1st, 1979 to Decembe 31st, 1987.

`ln_unemp` logarithm of unemployment rate

`spring` indicator for spring season.

`summer` indicator for summer season.

`fall` indicator for fall season.

References

Lenk (1999) Bayesian inference for semiparametric regression using a Fourier representation. *Journal of the Royal Statistical Society: Series B*, **61**(4), 863-879.

Examples

```
## Not run:
data(traffic)
pairs(traffic)

## End(Not run)
```

`wage.union` *Wage-Union data*

Description

This data set contains 534 observations on 11 variables.

Usage

```
data(wage.union)
```

Format

`education` number of years of education.

`south` indicator of living in southern region of U.S.A.

`sex` gender indicator: 0=male,1=female.

`experience` number of years of work experience.

`union` indicator of trade union membership: 0=non-member, 1=member.

`wage` wages in dollars per hour.

`age` age in years.

`race` 1=black, 2=Hispanic, 3=white.

`occupation` 1=management, 2=sales, 3=clerical, 4=service, 5=professional, 6=other.

`sector` 0=other, 1=manufacturing, 2=construction.

`married` indicator of being married: 0=unmarried, 1=married.

References

Berndt, E.R. (1991) *The Practice of Econometrics*. New York: Addison-Wesley.

Ruppert, D., Wand, M.P. and Carroll, R.J. (2003) *Semiparametric Regression*. Cambridge University Press.

Examples

```
## Not run:
data(wage.union)
pairs(wage.union)

## End(Not run)
```

Index