Package: GBASS (via r-universe)

May 25, 2026

Title Generalized Bayesian Adaptive Smoothing Splines

Version 2.0.1

Description Bayesian nonlinear regression under a range of likelihood models using generalized Bayesian adaptive smoothing splines. Robust regression with Student's t likelihoods, quantile regression, and related latent-scale models are included as special cases.

License BSD_3_clause + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports Matrix, GIGrvg, BASS

Suggests knitr, rmarkdown, lhs, testthat (>= 2.1.0)

NeedsCompilation no

Author Kellin Rumsey [aut, cre]

Maintainer Kellin Rumsey <knrumsey@lanl.gov>

Config/pak/sysreqs pari-gp

Repository https://knrumsey-lanl.r-universe.dev

Date/Publication 2026-04-24 21:09:35 UTC

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

RemoteRef HEAD

RemoteSha 0b6d6528d1680cd9dd19a203ac685371f0d960a5

Contents

`GBASS-package` *GBASS: Generalized Bayesian Adaptive Smoothing Splines*

Description

Bayesian nonlinear regression under a range of likelihood models using generalized Bayesian adaptive smoothing splines. Robust regression with Student's t likelihoods, quantile regression, and related latent-scale models are included as special cases.

Author(s)

Maintainer: Kellin Rumsey <knrumsey@lanl.gov>

`build_prior` *Construct a prior specification for GBASS*

Description

Construct a prior specification for GBASS

Usage

```
build_prior(
  type = c("GIG", "GBP"),
  p,
  a,
  b,
  lower_bound = NULL,
  prop_sigma = NULL,
  adapt = NULL,
  adapt_delay = NULL,
```

```
  adapt_thin = NULL
)

build_GIG(
  p,
  a,
  b,
  lower_bound = NULL,
  prop_sigma = NULL,
  adapt = NULL,
  adapt_delay = NULL,
  adapt_thin = NULL
)

build_GBP(
  p,
  a,
  b,
  lower_bound = NULL,
  prop_sigma = NULL,
  adapt = NULL,
  adapt_delay = NULL,
  adapt_thin = NULL
)
```

Arguments

type	"GIG" or "GBP"
p, a, b	prior hyperparameters
lower_bound	optional lower bound
prop_sigma	optional proposal sd (log scale)
adapt	logical; use adaptive MH?
adapt_delay	delay before adapting
adapt_thin	thinning for adaptation updates

Value

A named list containing the prior specification.

---

gbass *Generalized Bayesian MARS*

Description

Fits a generalized Bayesian MARS (GBASS) model for nonlinear regression under flexible latent-scale likelihoods.

Usage

```
gbass(
  X,
  y,
  w_prior = list(type = "GIG", p = 0, a = 0, b = 0),
  v_prior = list(type = "GIG", p = -15, a = 0, b = 30),
  maxInt = 3,
  maxBasis = 1000,
  npart = NULL,
  nmcmc = 10000,
  nburn = 9000,
  thin = 1,
  moveProbs = rep(1/3, 3),
  a_tau = 1/2,
  b_tau = NULL,
  a_lambda = 1,
  b_lambda = 1,
  m_beta = 0,
  s_beta = 0,
  scale = 1,
  Iw0 = rep(1, maxInt),
  Zw0 = rep(1, ncol(X)),
  verbose = TRUE
)
```

Arguments

X	An $N \times p$ numeric matrix of predictors. A numeric vector is treated as a single-column matrix.
y	A numeric response vector of length N.
w_prior	A named list specifying the prior for the global variance component. See Details.
v_prior	A named list specifying the shared prior for the local variance components. See Details.
maxInt	Integer giving the maximum interaction degree in proposed basis functions.
maxBasis	Maximum number of basis functions.
npart	Minimum number of nonzero points required for a proposed basis function. Defaults to `min(20, 0.1 * N)`.
nmcmc	Total number of MCMC iterations.
nburn	Number of initial iterations discarded as burn-in.
thin	Thinning interval for retained draws.
moveProbs	A length-3 vector giving probabilities for birth, death, and mutation moves.
a_tau	Prior hyperparameter for `tau`.
b_tau	Prior hyperparameter for `tau`. Defaults to `N / 2`.
a_lambda	Prior hyperparameter for `lambda`.

b_lambda	Prior hyperparameter for lambda.
m_beta	Prior mean for beta.
s_beta	Prior standard deviation for beta.
scale	Fixed variance-scale parameter. Defaults to 1.
Iw0	Vector of positive nominal weights for interaction order in proposed basis functions. Must have length maxInt.
Zw0	Vector of positive nominal weights for variable selection in proposed basis functions. Must have length ncol(X).
verbose	Logical; should progress be printed?

Details

The priors for w and v_i must belong to the generalized inverse Gaussian ("GIG") or generalized beta prime ("GBP") families.

Each prior list may contain:

1. type: either "GIG" or "GBP".
2. p, a, b: prior hyperparameters.
3. lower_bound: optional lower bound for the parameter support. For backward compatibility, lb is also accepted.
4. prop_sigma: proposal standard deviation on the log scale for Metropolis updates, when applicable.
5. adapt, adapt_delay, adapt_thin: optional controls for adaptive Metropolis updates when applicable.

For w, prop_sigma and the adaptive Metropolis fields are mainly relevant when w_prior$type = "GBP". In the "GIG" case with nonzero beta, w is sampled using the modified half-normal formulation rather than a random-walk Metropolis step.

Retained draws are taken at iterations nburn + 1, nburn + 1 + thin, ... Thus nburn is interpreted as the number of initial iterations discarded as burn-in.

Value

An object of class "gbass" containing posterior draws and fitted model information.

Note

Current implementation notes:

1. Basis function parameters are stored as lists.
2. Knot locations use continuous uniform proposals.
3. Basis coefficients use a ridge-type prior.

Examples

```
ff1 <- function(x) 10.391*((x[1]-0.4)*(x[2]-0.6) + 0.36)

n <- 100
p <- 4
X <- matrix(runif(n * p), nrow = n)
y <- apply(X, 1, ff1)
mod <- gbass(X, y, nmcmc = 1000, nburn = 900, thin = 2)
```

gbass2bass *Convert GBASS object to BASS-like object*

Description

Converts an object of class gbass to an object with class bass, so that downstream BASS utilities such as Sobol decomposition can be used.

Usage

```
gbass2bass(gm)

gm2bm(gm)
```

Arguments

gm	An object of class gbass.

Value

An object with class c("bass", "gbass").

gsobol *Sobol decomposition for GBASS models*

Description

Computes Sobol sensitivity indices for a fitted gbass model by first converting it to a compatible BASS object with gbass2bass(), and then calling BASS::sobol().

Usage

```
gsobol(object, ...)
```

Arguments

object	A fitted object of class "gbass".
...	Additional arguments passed to BASS::sobol().

Details

This is a thin wrapper around `BASS::sobol()` for GBASS models. Users who want direct access to the converted `BASS` object can call `gbass2bass()` explicitly and then use `BASS::sobol()` themselves.

Value

An object of class `"bassSob"` returned by `BASS::sobol()`.

See Also

`gbass2bass`, `sobol`

Examples

```
ff1 <- function(x) 10.391*((x[1]-0.4)*(x[2]-0.6) + 0.36)

n <- 100
p <- 4
X <- matrix(runif(n * p), nrow = n)
y <- apply(X, 1, ff1)

mod <- gbass(X, y, nmcmc = 1000, nburn = 900)

# Direct wrapper
sob <- gsobol(mod)

# Equivalent manual conversion
bm <- gbass2bass(mod)
sob2 <- BASS::sobol(bm)
```

---

hbass — *HBASS - Bayesian MARS with Horseshoe or Strawderman-Berger likelihood*

Description

Wrapper around `gbass()` using generalized beta prime latent-scale priors.

Usage

```
hbass(
  X,
  y,
  w_prior = list(type = "GBP", p = 1, a = 1/2, b = 1/2),
  likelihood = "h",
  ...
)
```

Arguments

`X`	An $N \times p$ numeric matrix of predictor variables.
`y`	A numeric response vector of length N.
`w_prior`	A named list specifying the prior for the global variance component. See Details.
`likelihood`	Character string specifying the likelihood family. Use `"h"` for Horseshoe or `"sb"` for Strawderman-Berger.
`...`	Additional arguments passed to `gbass()`.

Details

The Horseshoe and Strawderman-Berger likelihoods can be expressed using generalized beta prime latent-scale distributions. This function provides a convenient wrapper for these likelihoods. For additional flexibility, use `gbass()` directly.

Value

An object of class `c("hbass", "gbass")` containing posterior draws and fitted model information.

`nw_est_mom` *Method-of-moments estimation for the Normal-Wald distribution*

Description

Estimates Normal-Wald parameters from sample moments or from a supplied vector of moments.

Usage

```
nw_est_mom(
  data = NULL,
  stats = NULL,
  mu = NA,
  delta = NA,
  beta = NA,
  alpha = NA,
  triangle = FALSE,
  ...
)
```

Arguments

`data`	Optional data vector. If supplied, empirical moments are computed from `data` and `stats` is ignored.
`stats`	Optional numeric vector of length 4 containing mean, variance, skewness, and kurtosis. Entries may be `NA` when the corresponding parameter is fixed.
`mu`	Location parameter, if fixed.

delta	Scale parameter, if fixed.
beta	Skewness parameter, if fixed.
alpha	Tail parameter, if fixed.
triangle	Logical; if TRUE, return only the asymmetry and steepness summary parameters.
...	Additional arguments passed to optim.

Details

This function computes method-of-moments estimates for the Normal-Wald distribution. If data is supplied, sample mean, variance, skewness, and kurtosis are computed automatically. Otherwise, the user may provide these moments directly through stats.

If some parameters are fixed, only the remaining parameters are estimated. When triangle = TRUE, the returned values are the transformed summary quantities steepness $= (1 + |\gamma|)^{-1/2}$ and a corresponding asymmetry measure.

Value

If triangle = FALSE, a named numeric vector with entries mu, delta, beta, and alpha. If triangle = TRUE, a named numeric vector with entries asymmetry and steepness.

Examples

```
n <- 500
y <- rgamma(n, 3, 1.5) + rlnorm(n, 1, 0.5)
z <- (y - mean(y)) / sd(y)
skew <- mean(z^3)
kurt <- mean(z^4)

nw_est_mom(stats = c(NA, NA, skew, kurt), mu = 0, delta = 1, triangle = TRUE)
nw_est_mom(stats = c(NA, var(y), skew, kurt), mu = 0, triangle = TRUE)
```

---

nw_gamma_prior — *Calibrate a prior for the Normal-Wald gamma parameter*

Description

Selects prior hyperparameters for gamma in the Normal-Wald model using probability statements about the steepness parameter $\xi = (1 + \gamma)^{-1/2}$.

Usage

```
nw_gamma_prior(
  q1 = 0.1,
  q2 = 0.9,
  p1 = 0.5,
  p2 = 0.05,
  par0 = NULL,
  lambda = 0
)
```

Arguments

q1	Lower reference value for the steepness parameter. Default is 0.1.
q2	Upper reference value for the steepness parameter. Default is 0.9.
p1	Target probability associated with q1. Default is 0.5.
p2	Target probability associated with q2. Default is 0.05.
par0	Optional starting values for the numerical optimization.
lambda	Optional ridge penalty used in the optimization. Default is 0.

Details

This function chooses hyperparameters for a prior on gamma by solving a simple calibration problem based on the steepness parameter $(1+\gamma)^{-1/2}$. The optimization is carried out with optim using the Nelder-Mead method.

Value

A named numeric vector with entries m_gamma and s_gamma.

Examples

```
nw_gamma_prior()
nw_gamma_prior(q1 = 0.2, q2 = 0.8, p1 = 0.4, p2 = 0.1)
```

---

nw_triangle — *Plot Normal-Wald shape parameters on the asymmetry-steepness triangle*

Description

Visualizes posterior draws of the Normal-Wald shape parameters in terms of asymmetry and steepness. The plotted coordinates are steepness $= (1+\gamma)^{-1/2}$ and asymmetry $= \beta(\beta^2+\gamma^2)^{-1/2} \times$ steepness.

Usage

```
nw_triangle(obj, add = FALSE, details = FALSE, ...)
```

Arguments

obj	A fitted object containing components beta and gamma, typically an object returned by nwbass().
add	Logical; if FALSE, initialize a new triangle plot. If TRUE, add points to the current plot.
details	Logical; if TRUE and add = FALSE, add a few reference distributions to the plot.
...	Additional graphical arguments passed to points.

Details

This plot provides a simple geometric summary of the shape of the Normal-Wald likelihood. The upper boundary corresponds to symmetric models, while points away from zero on the horizontal axis indicate asymmetry.

Value

Invisibly returns a data frame with columns `asymmetry` and `steepness`.

Examples

```
# mod is a fitted nwbass model

# Simple example
n <- 200
X <- lhs::maximinLHS(n, 2)
f <- 20 * apply(X, 1, function(x) sin(pi * x[1]) + x[2]^2)
eps <- rgamma(n, 3, 1.5) - 2
y <- f + eps

mod <- nwbass(X, y,
              m_beta=0, s_beta=10,
              m_gamma = nw_gamma_prior()[1], s_gamma = nw_gamma_prior()[2])
nw_triangle(mod)
```

`nwbass` *Generalized Bayesian MARS with a Normal-Wald likelihood*

Description

Fits a generalized BMARS model under a Normal-Wald likelihood. This provides flexible nonlinear regression with a unimodal, potentially skewed error model.

Usage

```
nwbass(
  X,
  y,
  w_prior = list(type = "GIG", p = -0.1, a = 0, b = 0.1),
  maxInt = 3,
  maxBasis = 1000,
  npart = NULL,
  nmcmc = 10000,
  nburn = 9000,
  thin = 1,
  moveProbs = rep(1/3, 3),
  a_tau = 1/2,
```

```
  b_tau = NULL,
  a_lambda = 1,
  b_lambda = 1,
  m_beta = 0,
  s_beta = 0,
  lag_beta = 1001,
  m_gamma = 1,
  s_gamma = 0,
  scale = 1,
  Iw0 = rep(1, maxInt),
  Zw0 = rep(1, ncol(X)),
  verbose = TRUE
)

nwbass2(...)
```

Arguments

X	An $N \times p$ numeric matrix of predictor variables. A numeric vector is treated as a single-column matrix.
y	A numeric response vector of length N.
w_prior	A named list specifying the prior for the global variance component. See Details.
maxInt	Integer giving the maximum degree of interaction in spline basis functions. Defaults to 3.
maxBasis	Maximum number of basis functions.
npart	Minimum number of nonzero points required for a proposed basis function. Defaults to min(20, 0.1 * N).
nmcmc	Total number of MCMC iterations.
nburn	Number of initial MCMC iterations discarded as burn-in.
thin	Thinning interval for retained draws.
moveProbs	A length-3 vector giving probabilities for birth, death, and mutation moves.
a_tau	Prior hyperparameter for tau.
b_tau	Prior hyperparameter for tau. Defaults to N / 2.
a_lambda	Prior hyperparameter for lambda.
b_lambda	Prior hyperparameter for lambda.
m_beta	Prior mean for beta.
s_beta	Prior standard deviation for beta.
lag_beta	Number of initial iterations for which beta is fixed at m_beta. This is often used to stabilize early sampling.
m_gamma	Prior mean for gamma.
s_gamma	Prior standard deviation for gamma.
scale	Fixed variance-scale parameter. Defaults to 1.

`Iw0`	Vector of positive nominal weights for interaction order in proposed basis functions. Must have length `maxInt`.
`Zw0`	Vector of positive nominal weights for variable selection in proposed basis functions. Must have length `ncol(X)`.
`verbose`	Logical; should progress be printed?
`...`	(for backwards compatability)

Details

The latent local-scale prior is fixed internally to the Normal-Wald form $v_i \sim \text{GIG}(-1/2, \gamma^2, 1)$. Unlike `gbass()`, `nwbass()` does not expose a user-specified `v_prior`.

The `w_prior` list should contain:

1. `type`: either `"GIG"` or `"GBP"`.
2. `p`, `a`, `b`: hyperparameters for the prior.
3. `lower_bound`: optional lower bound for `w`. For backward compatibility, `lb` is also accepted.
4. `prop_sigma`: proposal standard deviation on the log scale for Metropolis updates of `w`. This is only used when `w` is updated by Metropolis-Hastings, such as the `"GBP"` case.
5. `adapt`, `adapt_delay`, `adapt_thin`: optional controls for adaptive Metropolis updates of `w` when applicable.

Retained draws are taken at iterations `nburn + 1, nburn + 1 + thin, ...`. Thus `nburn` is interpreted as the number of initial iterations discarded as burn-in.

Value

An object of class `c("nwbass", "gbass")` containing posterior draws and fitted model information.

Examples

```
n <- 200
X <- lhs::maximinLHS(n, 2)
f <- 20 * apply(X, 1, function(x) sin(pi * x[1]) + x[2]^2)
eps <- rgamma(n, 3, 1.5) - 2
y <- f + eps

mod <- nwbass(X, y,
              m_beta=0, s_beta=10,
              m_gamma = nw_gamma_prior()[1], s_gamma = nw_gamma_prior()[2])
```

plot.gbass	*Plot method for gbass objects*

Description

Plot method for gbass objects

Usage

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

Arguments

`x`	A fitted object of class `"gbass"`.
`...`	Additional graphical arguments.

Value

No return value, called for its side effects.

predict.gbass	*Predict method for GBASS objects*

Description

Returns posterior draws of either: (i) the linear predictor / mean surface, or (ii) the full posterior predictive distribution.

Usage

```
## S3 method for class 'gbass'
predict(
  object,
  newdata = NULL,
  mcmc.use = NULL,
  predictive = TRUE,
  bias_correct = FALSE,
  samples = 1,
  ...
)
```

Arguments

`object`	an object of class "gbass" (including subclasses like "tbass", "qbass", "nwbass")
`newdata`	a matrix of predictor variables. Defaults to training inputs.
`mcmc.use`	optional vector indicating which posterior draws to use.
`predictive`	logical. If TRUE, return posterior predictive draws. If FALSE, return draws of the linear predictor.
`bias_correct`	logical. Ignored unless predictive = FALSE. If TRUE, return the posterior mean response rather than just the linear predictor.
`samples`	Integer giving the number of predictive samples to generate per retained MCMC draw when `predictive = TRUE`. Ignored when `predictive = FALSE`. Default is `1`.
`...`	Additional graphical arguments (not used)

Details

If predictive = FALSE and bias_correct = FALSE, this returns draws of B(x)a.

If predictive = FALSE and bias_correct = TRUE, this returns draws of B(x)a + E(error | posterior draw), i.e. the mean response under the fitted GBASS error model.

If predictive = TRUE, this returns posterior predictive draws by simulating a fresh latent local variance v_new and Gaussian draw for each posterior sample.

For qbass objects, bias_correct = TRUE is usually not what you want, because qbass is typically being used for quantile regression rather than mean regression.

Currently, posterior predictive draws are implemented for GIG-based models. If the fitted object uses a GBP prior for v, predictive = TRUE will stop.

Value

a matrix with rows corresponding to posterior draws and columns corresponding to rows of newdata.

`print.gbass` *Print a gbass object*

Description

Print a gbass object

Usage

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

Arguments

`x`	An object of class `"gbass"`.
`...`	Unused.

Value

The input object, invisibly.

---

qbass *QBASS - Bayesian MARS with an asymmetric Laplace likelihood*

Description

Wrapper around gbass() for quantile regression.

Usage

```
qbass(
  X,
  y,
  q = 0.5,
  w_prior = list(type = "GIG", p = -0.1, a = 0, b = 0.1),
  ...
)
```

Arguments

X	An $N \times p$ numeric matrix of predictor variables.
y	A numeric response vector of length N.
q	Quantile of interest. Default is 0.5 for median regression.
w_prior	Prior for the global variance factor.
...	Additional arguments passed to gbass().

Details

Performs quantile regression for quantile q using the asymmetric Laplace representation. For many quantiles, fitting separate models in parallel may be convenient.

Value

An object of class c("qbass", "gbass") containing posterior draws and fitted model information.

rgig2 *Generalized Inverse Gaussian Generator*

Description

This function generates samples from the GIG(p, a, b) distribution with density function f(x) = cx^(p-1)exp(-1/2*(a*x + b/x))

Usage

```
rgig2(p, a, b)
```

Arguments

p	a real valued parameter
a	a non-negative parameter. If a=0, then p must be negative.
b	a non-negative parameter. If b=0, then p must be positive.

Details

A uniformly bounded rejection sample based on Hörmann et. al. (2014). Special cases include Gamma (b=0, p>0), Inverse Gamma (a=0, p<0) and Inverse Gaussian (p=-1/2).

Value

A numeric vector of random draws from the generalized inverse Gaussian distribution.

References

Hörmann, Wolfgang, and Josef Leydold. "Generating generalized inverse Gaussian random variates." Statistics and Computing 24.4 (2014): 547-557.

Examples

```
x <- rep(NA, 1000)
for(i in 1:1000) x[i] <- rgig2(2, 3, 0.5)
hist(x)
```

`rmhn` *Random generation from the modified half-normal distribution*

Description

Generates random samples from the modified half-normal distribution with density proportional to $x^{\alpha-1}\exp(-\beta x^2 + \gamma x)$ for $x > 0$.

Usage

```
rmhn(n = 1, alpha, beta, gamma)
```

Arguments

`n`	Number of random draws.
`alpha`	Positive shape parameter.
`beta`	Positive rate parameter multiplying x^2.
`gamma`	Real-valued linear parameter multiplying x.

Details

This distribution arises in latent-scale updates for the generalized Bayesian adaptive smoothing spline model. The function is intended mainly for internal model fitting, but it may also be useful on its own.

Value

A numeric vector of length `n` containing random draws from the modified half-normal distribution.

Examples

```
x <- rmhn(1000, alpha = 3, beta = 1, gamma = 0.5)
hist(x, breaks = 30, freq = FALSE)
```

`tbass` *TBASS - Bayesian MARS with a Student's t likelihood*

Description

Wrapper around `gbass()` for a Student's t error model.

Usage

```
tbass(X, y, df = 5, ...)
```

Arguments

`X`	An $N \times p$ numeric matrix of predictor variables.
`y`	A numeric response vector of length N.
`df`	Degrees of freedom. Default is `5`.
`...`	Additional arguments passed to `gbass()`.

Details

Uses an inverse-gamma latent-scale representation through a GIG prior on the local variance components. If `df > 2`, the scale is set to `(df - 2) / df` so that the marginal variance matches `w`.

Value

An object of class `c("tbass", "gbass")` containing posterior draws and fitted model information.

Index