Package 'sbde'

July 23, 2025

Version 1.0-1

Date 2024-02-16

Title Semiparametric Bayesian Density Estimation

Author Surya Tokdar <surya.tokdar@duke.edu>

Maintainer Surya Tokdar <surya.tokdar@duke.edu>

Depends R (>= 2.6)

Imports coda, extremefit

Description Offers Bayesian semiparametric density estimation
and tail-index estimation for heavy tailed data, by
using a parametric, tail-respecting transformation
of the data to the unit interval and then modeling
the transformed data with a purely nonparametric
logistic Gaussian process density prior. Based on
Tokdar et al. (2022) <doi:10.1080/01621459.2022.2104727>.

License GPL-2

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-02-16 23:40:02 UTC

Contents

coef.sbde *Coefficient Extraction from sbde Model Fit*

Description

Post process MCMC output from sbde to create summaries of parameter and quantile estimates.

Usage

```
## S3 method for class 'sbde'
coef(object, burn.perc = 0.5, nmc = 200,
    prob = c(.001,.01,.1,1:99,99.9,99.99,99.999)/100, ...)
```

Arguments

object	a fitted model of the class sbde.
burn.perc	a positive fraction indicating what fraction of the saved draws are to be discarded as burn-in
nmc	integer giving the number of samples, post burn-in, to be used in Monte Carlo averaging
prob	a numeric vector of probabiities at which quantiles are to be estimated.
...	not currently implemented

Value

Extracts posterior summary of model parameters, as well as estimated quantiles. A list is returned invisibly with the following fields.

psamp	a matrix with 3 columns and nmc rows storing the posterior draws of the parameters of base distribution used in transformation
parametric	a matrix with posterior median, 2.5th and 97.5th percentiles of the parameters of the base distribution.
prob	numeric vector of probabilities at which quantiles have been estimated. Could differ slightly from the input vector prob, by removing repetitions, as well as values that are not strictly between zero and one.
qsamp	a matrix with nmc columns giving the posterior draws of the quantile values at levels given by prob.
qest	a summary of qsamp given by the posterior median and 95 precent credible interval end points.
ss	a vector of integers giving the indices of the mcmc samples that were used in posterior summary calculations.

See Also

sbde, summary.sbde and predict.sbde for model fitting under sbde.

Examples

```
y <- abs(rt(n=1000, df=4))
fit <- sbde(y, blocking="all", fbase="gpd", verbose=FALSE)
coef(fit)
```

`predict.sbde` *Posterior predictive Summary for Semiparametric Density Estimation*

Description

Extract posterior predictive density estimate for sbde

Usage

```
## S3 method for class 'sbde'
predict(object, burn.perc = 0.5, nmc = 200, yRange = range(object$y), yLength = 401, ...)
```

Arguments

`object`	a fitted model of the class 'sbde'.
`burn.perc`	a positive fraction indicating what fraction of the saved draws are to be discarded as burn-in
`nmc`	integer giving the number of samples, post burn-in, to be used in Monte Carlo averaging
`yRange`	Range of values over which posterior predictive density is to be evaluated.
`yLength`	Number of grid points spanning yRange for posterior predictive density evaluation.
`...`	no additional parameters are used.

Value

Returns a list with three items:

`y`	vector giving the grid over which the posterior predictive density is evaluated.
`fsamp`	a matrix with `yLength` many rows and `nmc` many columns. Each column corresponds to a draw of the response density from the posterior predictive.
`fest`	summary of the posterior predictive density given by point-wise median, 2.5th and 97.5th percentiles.

See Also

sbde, coef.sbde and summary.sbde.

Examples

```
y <- abs(rt(n=1000, df=4))
fit <- sbde(y, blocking="all", fbase="gpd", verbose=FALSE)
pp <- predict(fit)
hist(y, 50, freq=FALSE)
with(pp, for(j in 1:3) lines(y, fest[,j], lty=1+(j!=2)))
```

---

sbde *Bayesian Semiparametric Density Estimation*

Description

Provides a semiparametric estimation of the density function of independent univariate data.

Usage

```
sbde(y, nsamp = 1e3, thin = 10, cens = rep(0,length(y)),
     wt = rep(1,length(y)), incr = list(knot=0.2, grid=0.01),
     par = c("Hill-kde", "pmean", "rand")[1], tail.warp = c(0,0),
     hyper = list(sig = c(.1,.1), lam = c(6,4), kap = c(1.5,1.5,1)),
     prox.range = c(.2,.95), acpt.target = 0.15, ref.size = 3,
     blocking = c("all", "gp", "loc+scale+tail"), temp = 1, expo = 2,
     blocks.mu, blocks.S, fix.nu = FALSE,
     fbase = c("t", "t+", "gpd", "gpd-rescaled", "unif"),
     spacing=list(knot="regular", grid="regular"),
     verbose = TRUE)

## S3 method for class 'sbde'
update(object, nadd, append = TRUE, ...)
```

Arguments

y	numeric vector of response data.
nsamp	number of posterior samples to be saved; defaults to 1000.
thin	thinning rate for the Markov chain sampler – one posterior sample is saved per thin iterations. Defaults to 10. The Markov chain sampler runs for a total of nsamp * thin many iterations.
cens	censoring status of response. Must be a vector of 0s and 1s of length same as length(y), with 1 indicating right censoring, and, 0 indicating no censoring. Defaults to all zeros.
wt	weights attached to the observation units, expected to be non-negative numbers, and defaults to a vector of ones.
incr	a list with two named elements, 'knot' and 'grid', giving the increment sizes for the knots in the predictive process approximation and the grid to be used for logistic Gaussian process likelihood evaluation. Defaults to 0.2 and 0.01 respectively

`par` either a numeric vector giving the parameter initialization or a character string indicating how the parameter should be initialized. If input numeric vector length is smaller than required parameter count, then supplied values are appended with zeros to create a full initialization. If input equals "pmean" then the mcmc is initialized at the prior center given by a vector of zeros, or if it equals "rand" then intialization is done by drawing randomly from the prior, or if it equals "Hill-kde" then the Hill estimate is used to estimate the shape parameter, the location and scale parameters are set based on data median and 95th percentile, and the initialization of the Gaussian process is done based on a kernel density estimate of the transformed data.

`tail.warp` a non-negative 2-vector giving the degrees of tail warping to be done at each tail. Larger values will allow more variation of the non-parametric density at the corresponding tail.

`hyper` hyperparameters of the prior distribution. Must be a list with one or both of the following two fields: `lam`: a two vector giving the parameters of the beta distribution on proximity = $\exp(-0.01*\lambda^2)$, and `kap`: a vector to be coerced into a `3 * nkap` matrix, with `nkap` being the number of components in the mixture of gamma prior on `kappa`, and each column of the matrix gives the shape, rate and mixing weight of a component.

`prox.range` for specifying the range of length-scale parameter of the Gaussian process prior.

`acpt.target` target acceptance rate of the adaptive Metropolis sampler; defaults to 0.15

`ref.size` adaptation rate of the adaptive Metropolis sampler. The proposal density is updated once every `ref.size` iterations. Could be a single number or a vector of length same as the number of blocks.

`blocking` type of blocking to be applied represented by a character vector with elements comprising of the strings: "gp", "loc", "scale", "tail" and their combinations separated by "+". Each of the basic string types will include the corresponding model parameters into the block. For example a valid input could be c("gp", "gp+loc+scale", "loc+scale+tail"), where the first block updates only the Gaussian process parameters, the second block jointly updates the GP parameters and the location and scale, and, the third block updates the location, scale and tail parameters. A combination of all four types can be represented as "all".

`temp` temperature of the log-likelihood function. The log-likelihood function is raised to the power of `temp`. Defaults to 1.

`expo` the exponent to be used in the covariance kernel of the Gaussian process priors. Defaults to 2, giving the standard squared-exponential covariance kernel.

`blocks.mu` initial block specific means in the form of a list. If left unspecified then will be automatically generated as a list of vectors of zeros of appropriate lengths matching the corresponding block sizes.

`blocks.S` initial block specific covariance matrices in the form of a list. If left unspecified then will be automatically generated as a list of identity matrices of appropriate dimensions matching the corresponding block sizes.

`fix.nu` either the logical FALSE indicating that nu should be learned, or a positive real number giving the fixed value of nu, which is then excluded from MCMC updates

fbase	either "t" (default) or "t+" (for half-t distributions on the positive real lines) or "gpd" (for generalized pareto distributions with location zero and parametrized by nu = 1 / shape) or "gpd-rescaled" (same as gpd, but scale parameter adjusted according to shape so that 90-th percentile matches that of gpd with shape=1/6 and scale=1) or "unif" to indicate what base distribution is to be used.
spacing	the type of spacing to be used for the predictive process knots and the likelihood evaluation grid. For either object, the default choice is "regular". Any other specification is taken to equal "irregular". A regular grid places points equally between 0 and 1 as given by the prespecified increment value. When the likelihood "grid" is chosen to be "irregular", the regular grid is appended with more points at both extremes by recursive bisection until 1/n or 1 - 1/n is reached. For predictive process knots, "irregular" applies only when `tail.warp` is different that `c(0,0)`, and more knots are appended at each extreme based on how much warping is done to it.
verbose	logical indicating whether MCMC progress should be printed, defaults to TRUE
object	a fitted model of the class 'qde'.
nadd	number of additional MCMC samples.
append	logical indicating whether new samples should be appended to old ones. If FALSE then old samples are discarded.
...	no additional arguments are allowed

Details

For positive valued data, it is recommended to use fbase as "gpd", which yields much faster computation than the choice of "t+". The difference is entirely due to difference in machine time needed to compute the CDF of the generalized Pareto versus that of the Student-t.

Value

`sbde(y, ...)` returns a 'sbde' class object to be used by `coef`, `summary` and `predict`.

Returned object is a list containing the following variables.

par	latest draw of the parameter vector
y	response vector
cens	censoring status vector
wt	vector of observation weights
hyper	completed list of hyper-parameters
dim	model dimension vector of the form c(n, length of tau grid, position of τ_0 on the grid, nknots, length of lambda grid, nkap, total number of MCMC iterations, thin, nsamp)
gridmats	details of covariance matrix factors etc, intended for internal use.
tau.g	the tau grid
muV	list of means for parameter blocks
SV	list of covariance matrices for parameter blocks

blocks	list of blocks
blocks.size	vector of block lengths
dmcmcpar	numeric vector containing details of adaptive MCMC runs, equals c(temp, decay rate of adaptation, vector of target acceptance rates for the blocks, vector of increment scales used in adaptation). Intended strictly for internal use.
imcmcpar	numeric vector containing details of adaptive MCMC runs, equals c(number of parameter blocks, ref.size, indicator on whether details are to be printed during MCMC progress, rate of details printing, a vector of counters needed for printing). Intended strictly for internal use.
parsamp	a long vector containing the parameter draws. Could be coerced into a matrix of dim npar * nsamp. Intended primarily for use by summary and coef.
acptsamp	a long vector containing rates of acceptance statistics for parameter blocks. Could be coerced into a matrix of dim nblocks * nsamp. Not very informative, because thinning times and adaptation times may not be exactly synced.
lpsamp	vector of log posterior values for the saved MCMC draws.
other.controls	a vector of two integers, with the first storing the choice of the fbase, and the second storing the choice of the gridtype.
prox	vector of proximity (exp(-0.01*lambda^2)) grid values
runtime	run time of the MCMC
base.bundle	a list of densty, distribution, quantile etc functions associated with the base distribution.

References

Tokdar, S.T., Jiang, S. and Cunningham, E.L. (2022). Heavy-tailed density estimation. *Journal of the American Statistical Association*, (just-accepted) <https://doi.org/10.1080/01621459.2022.2104727>.

See Also

summary.sbde, coef.sbde and predict.sbde.

Examples

```
y <- abs(rt(n=1000, df=4))
fit <- sbde(y, blocking="all", fbase="gpd", verbose=FALSE)
coef(fit)
```

---

summary.sbde *Summary Method for Semiparametric Density Estimation*

Description

Summarize model fit for sbde

Usage

```
## S3 method for class 'sbde'
summary(object, ntrace = 1000, burn.perc = 0.5, plot.dev = TRUE,
        more.details = FALSE, ...)
```

Arguments

object	a fitted model of the class 'sbde'.
ntrace	number of draws to be included in trace plots
burn.perc	fraction of MCMC draws to be discarded as burn-in.
plot.dev	logical indicator of whether to show trace plot of deviance
more.details	logical indicating whether other details from MCMC are to be plotted
...	a limited number of plotting controls that are passed onto the deviance plot

Value

Displays the trace of the deviance statistic. More details include trace plots of of the proximity parameter of each GP, a plot of Geweke p-values for (from geweke.diag) convergence of each model parameter and an image plot of parameter correlation.

The following quantities are returned invisibly.

deviance	vector deviance statistic of the samples parameter draws
pg	a matrix with nsamp number of columns. Each column gives the conditional posterior weights on the lambda grid values for the corresponding GP function.
prox	posterior draws of proximity parameter.
ll	a matrix of n*nsamp containing observation level log-likelihood contributions. Used to calculate *waic*, and could be used for other AIC calculations.
waic	Two versions of Watanabe AIC from Gelman, Hwang and Vehtari (2014).

References

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criterion for Bayesian models. *Stat Comput*, 24, 997-1016.

See Also

sbde and coef.sbde.

Examples

```
y <- abs(rt(n=1000, df=4))
fit <- sbde(y, blocking="all", fbase="gpd", verbose=FALSE)
sm <- summary(fit, more=TRUE)
print(sm$waic)
```

Index