Package 'vbayesGP'

August 4, 2023

Type Package

Title Gaussian Variational Approximation to Gaussian Process Regression

Version 0.1.0

Date 2023-07-30

Author Seongil Jo [aut, cre],
Woojoo Lee [aut]

Maintainer Seongil Jo <bstatsjo@gmail.com>

Description Implements Gaussian variational approximation to
Bayesian semiparametric regression with Gaussian process prior based on
the Radial basis function (RBF) kernel. Consider the normal prior,
the independent normal priors, or the horseshoe prior on
the positive real number for the lengthscale parameters of the RBF kernel.

License GPL-2

Imports Rcpp (>= 1.0.8), fields, ggplot2, MASS, ks, dplyr, magrittr

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.2.3

Encoding UTF-8

LazyLoad yes

R topics documented:

extractELBO *Extract ELBO from VGPR model fits*

Description

Compute the expected lower bound (ELBO) using the posterior samples for class "gpr"

Usage

```
extractELBO(object, nsamples = 1000)
```

Arguments

object	an object of class gpr.
nsamples	(positive integer), number of posterior samples to draw and save, defaults to 1000.

Author(s)

Seongil Jo

See Also

gvagpr

extractPostSamps *Extract Posterior Samples from VGPR model fits*

Description

Generate the posterior samples for class "gpr"

Usage

```
extractPostSamps(object, nsamples = 1000)
```

Arguments

object	an object of class gpr.
nsamples	(positive integer), number of posterior samples to draw and save, defaults to 1000.

Value

a data frame including posterior samples for β, σ^2, λ_f, and γ. If `object$id` is not `NULL`, the data frame also includes $b_i, i = 1, \ldots, N$.

Author(s)

Seongil Jo

See Also

gvagpr

`extractPPIPs` *Extract pseudo posterior inclusion probabilities (PPIPs) from VGPR model fits*

Description

Extract pseudo posterior inclusion probabilities (PPIPs) from Variational Gaussian Process Regression (VGPR) model fit

Usage

```
extractPPIPs(object, nsamples = 1000)
```

Arguments

`object`	an object of class `gpr`.
`nsamples`	(positive integer), number of posterior samples to draw and save, defaults to 1000.

Value

a data frame including the variable-specific PIPs for VGPR fit with horseshoe prior.

Author(s)

Seongil Jo

See Also

gvagpr

`fitted.gpr` *Extract GPR Model Fitted Values*

Description

fitted is a generic function which extracts fitted values of nonparametric part from an object of class "gpr"

Usage

```
## S3 method for class 'gpr'
fitted(object, nsamples = 1000)
```

Arguments

object	an object of class gpr.
nsamples	(positive integer) number of posterior samples. Default value is 1000.

Value

fmean	posterior mean of nonparametric part.
fcov	posterior variance of nonparametric part.

an object of class "gprfit", which has the associated method:

* plot (i.e., plot.gprfit)

Author(s)

Seongil Jo

See Also

gvagpr

---

gvagpr *Gaussian Variational Approximation to Gaussian Process Regression*

Description

Fits the Bayesian kernel machine regression using Gaussian variational approximation algorithm.

Usage

```
gvagpr(
  y,
  X,
  Z,
  id = NULL,
  random.slope = NULL,
  priors = list(),
  covstr = c("diagonal", "fullrank"),
  control = list(),
  minibatch = FALSE,
  verbose = TRUE,
  seed = 1
)
```

Arguments

y	a vector of response of length n.
X	an n-by-p matrix of covariates for parametric term. Should not contain an intercept.
Z	an n-by-M matrix of predictor variables to be included in nonparametric part.

id	optional vector (of length n) of grouping factors for fitting a model with random effects (including both a random intercept and a random slope). If NULL then no random effects will be included.
random.slope	a column index of the matrix (X) including covariates for random slope. If NULL and id is given, the model considers the random intercept only.
priors	a list giving the prior information. The list includes the following parameters (with default values in parentheses): asig (0.001) and bsig (0.001) giving the hyper parameters for σ^2, alam (0.1) and blam (0.01) giving the hyper parameters for λ_f, lam0 (1) and tau0 (1) giving the hyper parameters of the horseshoe prior.
covstr	Either "diagonal" (the default) or "fullrank", indicating which covariance structure of variational distribution is used. The "diagonal" option uses a fully factorized Gaussian for the approximation whereas the fullrank option uses a Gaussian with a full-rank covariance matrix for the approximation.
control	a named list of parameters to control the algorithm's behavior. The list includes the following parameters (with default values in parentheses): max_iter (100000) giving the maximum number of iterations, rho (0.95) giving the decaying constant, eps (1e-6) giving the small positive constant added to ensure the denominator of the step size is positive and the initial step size is nonzero, nws (2500) giving rolling window size for calculating the moving average of the lower bounds, nsp (100) giving the maximum patience parameter.
minibatch	TRUE or FALSE: If TRUE, nbatch (the number of batch) should be given in control argument and max_iter denotes the number of epoch. Default value is $n/150$.
verbose	TRUE or FALSE: flag indicating whether to print intermediate diagnostic information during the model fitting.
seed	The seed for random number generation. The default is generated from 1 to the maximum integer supported by **R** on the machine.

Details

Jo, and Lee (2023+) proposed the Bayesian semiparametric regression model with Gaussian process prior based on the Radial basis function (RBF) kernel:

$$y_i = x_i^\top \beta + f(z_i) + \epsilon_i, \;\; \epsilon_i \overset{iid}{\sim} N(0, \sigma^2),$$

$$f = (f(z_1), \ldots, f(z_D))^\top \sim GP(0, \sigma^2 \lambda_f K_D), \; z_i = (z_{i1}, \ldots, z_{iM})^\top,$$

where K_D denotes the RBF kernel given as

1) Equal lengthscale parameter:

$$K_D = \left(\exp\left(-\gamma \sum_{m=1}^{M} \|z_i - z_j\|^2 \right) \right)_{i,j=1}^{D}$$

2) Varying lengthscale parameters:

$$K_D = \left(\exp\left(-\sum_{m=1}^{M} \gamma_m \|z_i - z_j\|^2 \right) \right)_{i,j=1}^{D}$$

For the parameters, the following priors are used:

$$\pi(\beta) \propto 1,$$

$$\pi(\sigma^{-2}) = Gamma(a_\sigma, b_\sigma),$$
$$\pi(\lambda_f) = Gamma(a_\lambda, b_\lambda),$$

1) Normal prior:

$$\pi(\gamma) = N_+(0, \tau_0^2)$$

2) Independent Normal priors:

$$\pi(\gamma_m) = N_+(0, \tau_0^2),\ m = 1, \ldots, M$$

3) Horseshoe prior:

$$\pi(\gamma_m \mid \lambda_m, \tau_\gamma) = N_+(0, \lambda_m^2 \tau_\gamma^2),\ m = 1, \ldots, M$$
$$\pi(\lambda_m) = C_+(0, \lambda_0),\ m = 1, \ldots, M$$
$$\pi(\tau_\gamma) = C_+(0, \tau_0),$$

where $a_\sigma, b_\sigma, a_\lambda, b_\lambda, \lambda_0$ and τ_0 are positive constants specified by users.

For more details, see Jo and Lee (2023+).

Value

an object of class "gpr", which has the associated methods:

* extractELBO

* fitted (i.e., fitted.gpr)

* summary (i.e., summary.gpr)

* predict (i.e., predict.gpr)

* plot (i.e., plot.gpr)

Author(s)

Seongil Jo and Woojoo Lee

References

Jo, S., and Lee, W. (2023+), "Gaussian variational inference for Bayesian kernel machine regression with Horseshoe prior for estimating high-dimensional exposures", *preprint*.

Titsias, M. K. and L\'azaro-Gredilla, M. (2014), "Doubly stochastic variational Bayes for non-conjugate inference", *Proceedings of the 31st ICML*.

Bobb, J. F., Valeri, L., Claus, H. B., Christiani, D. C., Wright, R. O., Mazumdar, M., Godleski, J. J., and Coull, B. A. (2015). "Bayesian Kernel Machine Regression for Estimating the Health Effects of Multi-Pollutant Mixtures", *Biostatistics*, 16, 493-508.

Chen, H., Zheng, L., Kontai, R. A., and Raskutti, G. (2022), "Gaussian process parameter estimation using mini-batch stochastic gradient descent: convergence guarantees and empirical benefits", *Journal of Machine Learning Research*, 23, 1-59.

See Also

extractELBO, fitted.gpr, predict.gpr, plot.gpr, summary.gpr

Examples

```
## Not run:
sdat <- bkmr::SimData()
y <- sdat$y
X <- sdat$X
Z <- sdat$Z

fout <- vbayesGP::gvagpr(y, X, Z, priors = list(lengthscale = 'normal'), covstr = 'diagonal')
plot(fout)
summary(fout)
vbayesGP::extractELBO(fout)  # ELBO
## End(Not run)
```

`plot.gpr` *Plot Diagnostics for a gpr Object*

Description

Provides a plot of the smoothed evidence lower bound (ELBO) against iterations for checking the convergence.

Usage

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

Arguments

`x`	gpr object, result of **gvagpr**.
`...`	unused argument.

Author(s)

Seongil Jo

See Also

gvagpr

plot.gprfitBivar *Plot univariate predictor-response function on a new grid of points*

Description

Provides a plot of univariate predictor-response function on a new grid of points

Usage

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

Arguments

x	gpr object, result of **predictorResponseBivar**.
...	unused argument.

Author(s)

Seongil Jo

See Also

gvagpr predictorResponseBivar

plot.gprfitUnivar *Plot univariate predictor-response function on a new grid of points*

Description

Provides a plot of univariate predictor-response function on a new grid of points

Usage

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

Arguments

x	gpr object, result of **predictorResponseUnivar**.
...	unused argument.

Author(s)

Seongil Jo

See Also

gvagpr predictorResponseUnivar

predict.gpr *Extract GPR Model Predicted Values*

Description

predicted is a generic function which extracts predicted values for nonparametric part from an object of class "gpr"

Usage

```
## S3 method for class 'gpr'
predict(object, Z_new, nsamples = 1000, ...)
```

Arguments

object	an object of class gpr.
Z_new	a matrix of new predictor values at which to predict new f, where each row represents a new observation.
nsamples	(positive integer) number of posterior samples. Default value is 1000.
...	additional arguments affecting the predictions produced.

Value

fmean	posterior mean of nonparametric part.
fcov	posterior variance of nonparametric part.

an object of class "gprfit", which has the associated method:

* plot (i.e., plot.gprfit)

Author(s)

Seongil Jo

See Also

gvagpr

predictorResponseBivar *Predict the exposure-response function at a new grid of points*

Description

Predict the exposure-response function at a new grid of points

Usage

```
predictorResponseBivar(
  fit,
  z.pairs = NULL,
  ngrid = 50,
  q.fixed = 0.5,
  nsamples = 1000,
  min.plot.dist = 0.5,
  center = TRUE,
  z.names = colnames(Z),
  verbose = TRUE,
  ...
)
```

Arguments

fit	an object of class gpr
z.pairs	data frame showing which pairs of predictors to plot
ngrid	number of grid points in each dimension
q.fixed	a second quantile at which to compare the estimated f function
nsamples	(positive integer) number of posterior samples. Default value is 1000
min.plot.dist	specifies a minimum distance that a new grid point needs to be from an observed data point in order to compute the prediction; points further than this will not be computed
center	flag for whether to scale the exposure-response function to have mean zero
z.names	a vector of names of predictors Z. Default values are colnames(Z)
verbose	TRUE or FALSE: flag of whether to print intermediate output to the screen
...	additional arguments affecting the predictions produced

Value

a long data frame with the name of the first predictor, the name of the second predictor, the value of the first predictor, the value of the second predictor, the posterior mean estimate, and the posterior standard deviation of the estimated exposure response function

References

Bobb J (2023). _bkmr: Bayesian Kernel Machine Regression_. R package version 0.2.2, <https://github.com/jenfb/bkmr>

Examples

```
## Not run:
## First generate dataset
set.seed(111)
dat <- bkmr::SimData(n = 50, M = 4)
y <- dat$y
Z <- dat$Z
X <- dat$X

set.seed(111)
priors = list(lengthscale = 'horseshoe')
```

```
fout <- vbayesGP::gvagpr(y = y, Z = Z, X = X, priors = priors, covstr = 'diagonal')

## Obtain predicted value on new grid of points for each pair of predictors
## Using only a 10-by-10 point grid to make example run quickly
pred.resp.bivar <- vbayesGP::PredictorResponseBivar(fit = fout, min.plot.dist = 1, ngrid = 10)
## End(Not run)
```

`predictorResponseBivarPair` *Predict bivariate predictor-response function on a new grid of points*

Description

Predict bivariate predictor-response function on a new grid of points

Usage

```
predictorResponseBivarPair(
  fit,
  whichz1 = 1,
  whichz2 = 2,
  whichz3 = NULL,
  prob = 0.5,
  q.fixed = 0.5,
  nsamples = 1000,
  ngrid = 50,
  min.plot.dist = 0.5,
  center = TRUE,
  ...
)
```

Arguments

`fit`	an object of class gpr
`whichz1`	vector identifying the first predictor that (column of Z) should be plotted
`whichz2`	vector identifying the second predictor that (column of Z) should be plotted
`whichz3`	vector identifying the third predictor that will be set to a pre-specified fixed quantile (determined by prob)
`prob`	pre-specified quantile to set the third predictor (determined by whichz3); defaults to 0.5 (50th percentile)
`q.fixed`	a second quantile at which to compare the estimated f function
`nsamples`	(positive integer) number of posterior samples. Default value is 1000
`ngrid`	number of grid points in each dimension
`min.plot.dist`	specifies a minimum distance that a new grid point needs to be from an observed data point in order to compute the prediction; points further than this will not be computed
`center`	flag for whether to scale the exposure-response function to have mean zero
`...`	additional arguments affecting the predictions produced

Value

a data frame with value of the first predictor, the value of the second predictor, the posterior mean estimate, and the posterior standard deviation

Examples

```
## Not run:
## First generate dataset
set.seed(111)
dat <- bkmr::SimData(n = 50, M = 4)
y <- dat$y
Z <- dat$Z
X <- dat$X

set.seed(111)
priors = list(lengthscale = 'horseshoe')
fout <- vbayesGP::gvagpr(y = y, Z = Z, X = X, priors = priors, covstr = 'diagonal')

## Obtain predicted value on new grid of points
## Using only a 10-by-10 point grid to make example run quickly
pred.resp.bivar12 <- PredictorResponseBivarPair(fit = fout, min.plot.dist = 1, ngrid = 10)
## End(Not run)
```

---

predictorResponseUnivar

Predict univariate predictor-response function on a new grid of points

---

Description

Predict univariate predictor-response function on a new grid of points

Usage

```
predictorResponseUnivar(
  fit,
  which.z = 1:ncol(Z),
  ngrid = 50,
  q.fixed = 0.5,
  nsamples = 1000,
  min.plot.dist = Inf,
  center = TRUE,
  z.names = colnames(Z),
  ...
)
```

Arguments

`fit`	an object of class `gpr`
`which.z`	vector identifying which predictors (columns of `Z`) should be plotted
`ngrid`	number of grid points to cover the range of each predictor (column in `Z`)
`q.fixed`	a second quantile at which to compare the estimated `f` function

nsamples	(positive integer) number of posterior samples. Default value is 1000
min.plot.dist	specifies a minimum distance that a new grid point needs to be from an observed data point in order to compute the prediction; points further than this will not be computed
center	flag for whether to scale the exposure-response function to have mean zero
z.names	a vector of names of predictors Z. Default values are colnames(Z).
...	additional arguments affecting the predictions produced.

Value

a long data frame with the predictor name, predictor value, posterior mean estimate, and posterior standard deviation

References

Bobb, J. F., Valeri, L., Claus, H. B., Christiani, D. C., Wright, R. O., Mazumdar, M., Godleski, J. J., and Coull, B. A. (2015). "Bayesian Kernel Machine Regression for Estimating the Health Effects of Multi-Pollutant Mixtures", *Biostatistics*, 16, 493-508.

Bobb J (2023). _bkmr: Bayesian Kernel Machine Regression_. R package version 0.2.2, <https://github.com/jenfb/bkmr>

Examples

```
## Not run:
## First generate dataset
set.seed(111)
dat <- bkmr::SimData(n = 50, M = 4)
y <- dat$y
Z <- dat$Z
X <- dat$X

set.seed(111)
priors <- list(lengthscale = 'horseshoe')
fout <- vbayesGP::gvagpr(y = y, Z = Z, X = X, priors = priors, covstr = 'diagonal')
pred.resp.univar <- vbayesGP::PredictorResponseUnivar(fout)
## End(Not run)
```

---

print.gpr *Print basic summary of gpr model fit*

Description

print method for class "gpr"

Usage

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

Arguments

x	an object of class gpr.
...	unused argument.

Author(s)

Seongil Jo

See Also

gvagpr

`summary.gpr` *Summarizing gpr model fits*

Description

summary method for class "gpr"

Usage

```
## S3 method for class 'gpr'
summary(object, q = c(0.025, 0.975), digits = 5, nsamples = 1000, ...)
```

Arguments

`object`	an object of class `gpr`.
`q`	quantiles of posterior distribution (credible interval) to show.
`digits`	the number of digits to show when printing.
`nsamples`	(positive integer), number of posterior samples to draw and save, defaults to 1000.
`...`	unused argument.

Author(s)

Seongil Jo

See Also

gvagpr

Index