Package ‘vbayesGP’

August 21, 2024

Type Package

Title Gaussian Variational Approximation to Gaussian Process Regression

Version 0.5.1

Date 2024-08-21

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 lengthscale parameters of the RBF kernel.

License GPL-2

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

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 3.5.0)

RoxygenNote 7.3.2

Encoding UTF-8

LazyLoad yes

NeedsCompilation yes

Contents

`computeOverallRisk` *Calculate Overall Risk Summaries*

Description

Compare estimated `f` function when all predictors are at a particular quantile to when all are at a second fixed quantile

Usage

```
computeOverallRisk(
  object,
  qs = seq(0.25, 0.75, by = 0.05),
  q.fixed = 0.5,
  nsamples = 1000,
  ...
)
```

Arguments

`object`	an object of class `gpr` obtained from **gvagpr** function
`qs`	vector of quantiles at which to calculate the overall risk summary
`q.fixed`	a second quantile at which to compare the estimated `f` function
`nsamples`	(positive integer), number of posterior samples to draw and save, defaults to 1000
`...`	unused argument

Value

a data frame containing the (posterior mean) estimate and posterior standard deviation of the overall risk measures

References

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

See Also

gvagpr

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')
risks.overall <- vbayesGP::computeOverallRisk(fout, qs = seq(0.25, 0.75, by = 0.05), q.fixed = 0.5)
plot(risk.overall)
## End(Not run)
```

`computePPIPs` *Compute pseudo posterior inclusion probabilities (PPIPs) from VGPR model or VGPMIM fits*

Description

Compute pseudo posterior inclusion probabilities (PPIPs) using Sequential-2-Means from Variational Gaussian Process Regression (VGPR) model fit or Variational Gaussian Process Multiple Index Model (VGPMIM) fit

Usage

```
computePPIPs(object, nsamples = 1000, ntuning = 10)
```

Arguments

`object`	an object of class `gpr` or `gpmim`
`nsamples`	(positive integer), number of posterior samples to draw and save, defaults to 1000
`ntuning`	the number of chosen values of the tuning parameter for variable selection

Value

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

Author(s)

Seongil Jo

References

Li, H. and Pati, D. (2017). "Variable Selection Using Shrinkage Priors", Computational Statistics and Data Analysis, 107, 107-119.

See Also

gvagpr, gvagpmim

`computeSingVarInt` *Single Variable Interaction Summaries*

Description

Compare the single-predictor health risks when all of the other predictors in Z are fixed to their a specific quantile to when all of the other predictors in Z are fixed to their a second specific quantile.

Usage

```
computeSingVarInt(
  object,
  which.z,
  qs.diff = c(0.25, 0.75),
  qs.fixed = c(0.25, 0.75),
  nsamples = 1000,
  ...
)
```

Arguments

`object`	an object of class `gpr` obtained from **gvagpr** function
`which.z`	vector indicating which variables (columns of `Z`) for which the summary should be computed
`qs.diff`	vector indicating the two quantiles at which to compute the single-predictor risk summary
`qs.fixed`	vector indicating the two quantiles at which to fix all of the remaining exposures in `Z`
`nsamples`	(positive integer), number of posterior samples to draw and save, defaults to 1000
`...`	unused argument

Value

a data frame containing the (posterior mean) estimate and posterior standard deviation of the single-predictor risk measures

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')
risks.int <- computeSingVarInt(fout)
plot(risks.int)
## End(Not run)
```

`computeSingVarRisk` *Single Variable Risk Summaries*

Description

Compute summaries of the risks associated with a change in a single variable in `Z` from a single level (quantile) to a second level (quantile), for the other variables in `Z` fixed to a specific level (quantile)

Usage

```
computeSingVarRisk(
  object,
  which.z,
  qs.diff = c(0.25, 0.75),
  q.fixed = c(0.25, 0.5, 0.75),
  nsamples = 1000,
  ...
)
```

Arguments

`object`	an object of class `gpr` obtained from **gvagpr** function.
`which.z`	vector indicating which variables (columns of `Z`) for which the summary should be computed
`qs.diff`	vector indicating the two quantiles `q_1` and `q_2` at which to compute `f(z_{q2}) - f(z_{q1})`
`q.fixed`	vector of quantiles at which to fix the remaining predictors in `Z`
`nsamples`	(positive integer), number of posterior samples to draw and save, defaults to 1000
`...`	unused argument

Value

a data frame containing the (posterior mean) estimate and posterior standard deviation of the single-predictor risk measures

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)
fout <- vbayesGP::gvagpr(y = y, Z = Z, X = X, priors = priors, covstr = 'diagonal')
risks.singvar <- computeSingVarRisk(fout)
plot(risk.singvar)
## End(Not run)
```

`extractELBO` *Extract ELBO from gpr, ggpr, or gpmim model fits*

Description

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

Usage

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

Arguments

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

Author(s)

Seongil Jo

See Also

gvagpr, gvaggpr, gvagpmim

`extractPostSamps` *Extract Posterior Samples from gpr, ggpr, or gpmim model fits*

Description

Generate the posterior samples for class "gpr" or "gpmim"

Usage

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

Arguments

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

Value

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

Author(s)

Seongil Jo

See Also

gvagpr, gvaggpr, gvagpmim

`fitted.gpmim` *Extract GPMIM Fitted Values*

Description

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

Usage

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

Arguments

`object`	an object of class `gpmim`.
`nsamples`	(positive integer) number of posterior samples. Default value is 1000.
`...`	unused argument.

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

gvagpmim

`fitted.gpr` *Extract GPR and GGPR 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.

`...` unused argument.

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, gvaggpr

gvaggpr *Gaussian Variational Approximation to Generalized Gaussian Process Regression*

Description

Fits the Bayesian kernel machine regression for generalized linear model using Gaussian variational approximation algorithm.

Usage

```
gvaggpr(
  y,
  X,
  Z,
  id = NULL,
  random.slope = NULL,
  family = binomial,
  priors = list(),
  covstr = c("diagonal", "full"),
  control = list(),
  verbose = TRUE,
  seed = sample.int(.Machine$integer.max, 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.
`family`	a description of the error distribution and link function to be used in the model. Currently implemented for binomial family. (See `family` of `base` for details of family functions.)
`priors`	a list giving the prior information. The list includes the following parameters (with default values in parentheses): `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 "full", indicating which covariance structure of variational distribution for global parameters is used. The "diagonal" option uses a fully factorized Gaussian for the approximation whereas the "full" 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.

`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 generalized linear model with Gaussian process prior based on the Radial basis function (RBF) kernel:

$$y_i \sim Bern(p_i), \log\left(p_i/(1+p_i)\right) = x_i^\top \beta + f(z_i),$$

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

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:

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

$$\pi(\lambda_f) = Gamma(a_\lambda, b_\lambda),$$

1) Normal prior:

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

2) Independent Normal priors:

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

3) Horseshoe prior:

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

$$\pi(\lambda_m) = C_+(0, \lambda_0),\ m = 1, \dots, M$$

$$\pi(\tau_\gamma) = C_+(0, \tau_0),$$

where $a_\lambda, b_\lambda, \lambda_0$ 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 exponential families", *preprint*.

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::gvaggpr(y, X, Z, priors = list(lengthscale = 'normal'), family = binomial)
plot(fout)
summary(fout)
vbayesGP::extractELBO(fout)  # ELBO
## End(Not run)
```

gvagpmim *Gaussian Variational Approximation to Gaussian Process Multiple Index Model*

Description

Fits the Bayesian kernel machine multiple index model using Gaussian variational approximation algorithm.

Usage

```
gvagpmim(
  y,
  X,
  Z,
  m.index = NULL,
  id = NULL,
  random.slope = NULL,
  priors = list(),
  covstr = c("diagonal", "fullrank", "sparseprec"),
  control = list(),
  minibatch = FALSE,
  verbose = TRUE,
  seed = sample.int(.Machine$integer.max, 1)
)
```

Arguments

y	a vector of response of length n.
X	a matrix of covariates for parametric term. Should not contain an intercept.
Z	a matrix or a list of predictor variables to be included in nonparametric part.
m.index	a list of column indices grouping the predictor variables in nonparametric part. If Z is a matrix, m.index must be given.
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), "fullrank", or "sparseprec", 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 mixed model, the "sparseprec" option utilizes a Gaussian with sparse precision matrix whereas the "fullrank" option uses a Gaussian with a block-diagonal covariance matrix.

`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 $n/100$. Note. this option is not applicable for the "sparseprec" option of `covstr` and the random effects models.

`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:

$$y_i = x_i^\top \beta + f^M(z_{i1}^\top \alpha_1, \dots, z_{iM}^\top \alpha_M) + \epsilon_i, \;\; \epsilon_i \overset{iid}{\sim} N(0, \sigma^2),$$
$$f_i^M = f^M(z_{i1}^\top \alpha_1, \dots, z_{iM}^\top \alpha_M),$$
$$f^M = (f_1^M, \dots, f_D^M)^\top \sim GP(0, \sigma^2 \lambda_f K_D),$$

where K_D denotes the RBF kernel given as

Varying lengthscale parameters:

$$K_D = \left(\exp\left(-\sum_{m=1}^{M} ((z_{im} - z_{jm})^\top D_m^{-1} \gamma_m)^2 \right) \right)_{i,j=1}^{D}, \;\; \gamma_m = (\gamma_{m1}, \dots, \gamma_{mL_m})^\top,$$

where D_m is a non-singular matrix for constraints.

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

$$y_{ir} = x_{ir}^\top \beta + f^M(z_{ir1}^\top \alpha_1, \dots, z_{irM}^\top \alpha_M) + u_{ir}^\top b_i + \epsilon_{ir}, \;\; \epsilon_{ir} \overset{iid}{\sim} N(0, \sigma^2),$$
$$b_i = (b_{i1}, \dots, b_{iq})^\top \overset{iid}{\sim} N(0, \Sigma_b).$$

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) Independent Normal priors:

$$\pi(\gamma_{ml}) = N(0, \tau_0^2), \; m = 1, \dots, M, \; l = 1, \dots, L_m - 1$$
$$\pi(\gamma_{mL_m}) = N_+(0, \tau_0^2), \; m = 1, \dots, M$$

2) Horseshoe prior:

$$\pi(\gamma_{ml} \mid \lambda_{ml}, \tau_m, \tau_\gamma) = N(0, \lambda_{ml}^2 \tau_m^2 \tau_\gamma^2),\ m = 1, \ldots, M,\ l = 1, \ldots, L_m - 1$$

$$\pi(\gamma_{mL_m} \mid \lambda_{mL_m}, \tau_m, \tau_\gamma) = N_+(0, \lambda_{mL_m}^2 \tau_m^2 \tau_\gamma^2),\ m = 1, \ldots, M$$

$$\pi(\lambda_{ml}) = C_+(0, \lambda_0),\ m = 1, \ldots, M,\ l = 1, \ldots, L_m,$$

$$\pi(\tau_m) = C_+(0, \tau_0),\ m = 1, \ldots, M,$$

$$\pi(\tau_\gamma) = C_+(0, \tau_{\gamma,0}),$$

where $a_\sigma, b_\sigma, a_\lambda, b_\lambda, \lambda_0, \tau_0$ and $\tau_{\gamma,0}$ are positive constants specified by users.

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

Value

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

* extractELBO

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

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

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

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

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.

McGee, G., Wilson, A., Webster, T. F., Coull, B. A. (2021), "Bayesian multiple index models for environmental mixtures", *Biometrics*, 1-13.

See Also

extractELBO, fitted.gpmim, predict.gpmim, plot.gpmim, summary.gpmim

Examples

```
## Not run:
sdat <- bkmr::SimData(M = 13)
y <- sdat$y
X <- sdat$X
Z <- sdat$Z
m.index <- list(1:5, 6:13)

fout <- vbayesGP::gvagpmim(y, X, Z, m.index, priors = list(lengthscale = 'normal'))
plot(fout)
summary(fout)
vbayesGP::extractELBO(fout)  # ELBO
## End(Not run)
```

---

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", "sparseprec"),
  control = list(),
  minibatch = FALSE,
  verbose = TRUE,
  seed = sample.int(.Machine$integer.max, 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), "fullrank", or "sparseprec", 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 mixed model, the "sparseprec" option utilizes a Gaussian with sparse precision matrix whereas the "fullrank" option uses a Gaussian with a block-diagonal covariance matrix.

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 $n/100$. Note. this option is not applicable for the "sparseprec" option of `covstr` and the random effects models.

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:

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

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

$$y_{ir} = x_{ir}^\top\beta + f(z_{ir}) + u_{ir}^\top b_i + \epsilon_{ir},\ \ \epsilon_{ir} \overset{iid}{\sim} N(0,\sigma^2),$$

$$b_i = (b_{i1}, \dots, b_{iq})^\top \overset{iid}{\sim} N(0, \Sigma_b).$$

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:

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

3) Horseshoe prior:

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

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.

Tan, L. S. L. and Nott, D. J. (2018), "Gaussian variatioal approximation with sparse precision matrices", *Statistics and Computing*, 28, 259-275.

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)
```

`nhanes` *National Health and Nutrition Examination Survey (NHANES) dataset*

Description

The NHANES dataset in the context of an Environmental Mixtures Workshop held in 2018/2019 at the Mailman School of Public Health, Columbia University.

Format

'data.frame'

Source

https://github.com/lizzyagibson/SHARP.Mixtures.Workshop/tree/1f2da3a14bb096d99b2c45a69d11053b0ef60088.

Examples

```
data("nhanes")
head(nhanes)
```

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

Description

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

Usage

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

Arguments

`x`	`gpmim` object, result of **gvagpmim**.
`...`	unused argument.

Author(s)

Seongil Jo

See Also

gvagpmim

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

Description

Provides a plot of bivariate 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.gprfitBivarLevels` *Plot interactions*

Description

Provides a plot of the predictor-response function of a single predictor in Z for the second predictor in Z fixed at various quantiles

Usage

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

Arguments

`x`	`gpr` object, result of **predictorResponseBivarLevels**.
`...`	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

plot.risks — *Plot Risk Summaries*

Description

Provides a plot of risk summaries

Usage

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

Arguments

`x`	`risks` object, result of `computeOverallRisk`, `computeSingVarRisk`, or `computeSingVarInt`
`...`	unused argument

Author(s)

Seongil Jo

See Also

gvagpr computeOverallRisk computeSingVarRisk computeSingVarInt

predict.gpmim *Extract GPMIM Predicted Values*

Description

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

Usage

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

Arguments

`object`	an object of class `gpmim`.
`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

gvagpmim

predict.gpr *Extract GPR and GGPR 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, gvaggpr

`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,
  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
`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)
```

`predictorResponseBivarLevels`
Plot cross-sections of the bivariate predictor-response function

---

Description

Function to plot the `f` function of a particular variable at different levels (quantiles) of a second variable

Usage

```
predictorResponseBivarLevels(
  object,
  Z = NULL,
  qs = c(0.25, 0.5, 0.75),
  both_pairs = TRUE
)
```

Arguments

`object`	object obtained from running the function `predictorResponseBivar`
`Z`	an n-by-M matrix of predictor variables to be included in nonparametric part.
`qs`	vector of quantiles at which to fix the second variable
`both_pairs`	flag indicating whether, if `h(z1)` is being plotted for z2 fixed at different levels, that they should be plotted in the reverse order as well (for `h(z2)` at different levels of z1)

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 quantile at which the second predictor is fixed, 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)
pred.resp.bivar.levels <- vbayesGP::predictorResponseBivarLevels(pred.resp.df = pred.resp.bivar,
Z = Z, qs = c(0.1, 0.5, 0.9))
## 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

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
## Using only a 10-by-10 point grid to make example run quickly
pred.resp.bivar12 <- vbayesGP::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.gpmim — *Print basic summary of gpmim and ggpmim model fit*

Description

print method for class "gpmim"

Usage

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

Arguments

`x`	an object of class `gpmim`.
`...`	unused argument.

Author(s)

Seongil Jo

See Also

gvagpmim, gvaggpmim

`print.gpr` *Print basic summary of gpr and ggpr 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, gvaggpr

summary.gpmim *Summarizing gpmim and ggpmim model fits*

Description

summary method for class "gpmim"

Usage

```
## S3 method for class 'gpmim'
summary(
  object,
  q = c(0.025, 0.975),
  digits = 5,
  nsamples = 1000,
  ntuning = 10,
  ...
)
```

Arguments

`object`	an object of class `gpmim`
`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
`ntuning`	the number of chosen values of the tuning parameter for variable selection, defaults to 10
`...`	unused argument.

Author(s)

Seongil Jo

See Also

gvagpmim, gvaggpmim

summary.gpr *Summarizing gpr and ggpr 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,
  ntuning = 10,
  ...
)
```

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
`ntuning`	the number of chosen values of the tuning parameter for variable selection, defaults to 10
`...`	unused argument.

Author(s)

Seongil Jo

See Also

gvagpr, gvaggpr

Index