Package: gamselBayes (via r-universe)

September 29, 2024

Version 2.0-1

Date 2023-09-03

Title Bayesian Generalized Additive Model Selection

Maintainer Matt P. Wand <matt.wand@uts.edu.au>

Description Generalized additive model selection via approximate Bayesian inference is provided. Bayesian mixed model-based penalized splines with spike-and-slab-type coefficient prior distributions are used to facilitate fitting and selection. The approximate Bayesian inference engine options are: (1) Markov chain Monte Carlo and (2) mean field variational Bayes. Markov chain Monte Carlo has better Bayesian inferential accuracy, but requires a longer run-time. Mean field variational Bayes is faster, but less accurate. The methodology is described in He and Wand (2023) <arXiv:2201.00412>.

License GPL (>= 2)

Depends R (>= 3.5.0)

Imports Rcpp, methods

Suggests Ecdat

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Virginia X. He [aut] (<https://orcid.org/0000-0002-0238-5018>), Matt P. Wand [aut, cre] (<https://orcid.org/0000-0003-2555-896X>)

Repository CRAN

Date/Publication 2023-09-03 12:50:02 UTC

Contents

`checkChains` *Check Markov chain Monte Carlo samples*

Description

Facilitates a graphical check of the Markov chain Monte Carlo samples ("chains") corresponding to key quantities for the predictors selected as having an effect.

Usage

```
checkChains(fitObject,colourVersion = TRUE,paletteNum = 1)
```

Arguments

`fitObject`	`gamselBayes()` fit object.
`colourVersion`	Boolean flag: TRUE = produce a colour version of the graphical check (the default) FALSE = produce a black-and-white version of the graphical check.
`paletteNum`	The palette of colours for the graphical display when colourVersion is TRUE. Two palettes, numbered 1 and 2, are available. The value of `paletteNum` specifies the palette to use. The default value is 1.

Details

A graphic is produced that summarises the Markov chain Monte Carlo samples ("chains") corresponding to key quantities for the predictors selected as having an effect. If the predictor is found to have a linear effect then the chain for its coefficient is graphically checked. It the predictor is found to have a non-linear effect then the chain for the vertical slice of the penalized spline fit at the median of the predictor sample is graphically checked. The columns of the graphic are the following summaries of each chain: (1) trace (time series) plot, (2) lag-1 plot in which each chain value is plotted against its previous value and (3) sample autocorrelation function plot as produced by the R function acf(). A rudimentary graphical assessment of convergence involveschecking that the trace plots have flat-lined, rather than having any noticeable trends. If the latter occurs that a longer warmup is recommended.

Value

Nothing is returned.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

Examples

```
library(gamselBayes)

# Generate some simple regression-type data:

set.seed(1) ; n <- 1000 ; x1 <- rbinom(n,1,0.5) ;
x2 <- runif(n) ; x3 <- runif(n) ; x4 <- runif(n)
y <- x1 + sin(2*pi*x2) - x3 + rnorm(n)
Xlinear <- data.frame(x1) ; Xgeneral <- data.frame(x2,x3,x4)

# Obtain a gamselBayes() fit for the data:

fitMCMC <- gamselBayes(y,Xlinear,Xgeneral)
summary (fitMCMC)

# Obtain a graphic for checking the chains:

checkChains(fitMCMC)

if (require("Ecdat"))
{
   # Obtain a gamselBayes() fit for data on schools in California, U.S.A.:

   Caschool$log.avginc <- log(Caschool$avginc)
   mathScore <- Caschool$mathscr
   Xgeneral <- Caschool[,c("mealpct","elpct","calwpct","compstu","log.avginc")]
   fitMCMC <- gamselBayes(y = mathScore,Xgeneral = Xgeneral)
   summary(fitMCMC)

   # Obtain a graphic for checking the chains:

   checkChains(fitMCMC)
}
```

---

effectTypes *Tabulate the estimated effect types from a Bayesian generalized additive model object*

Description

Produces a tabular summary of the estimated effect types from a gamselBayes() fit object.

Usage

```
effectTypes(fitObject)
```

Arguments

fitObject	gamselBayes() fit object.

Details

Two tables are printed to standard output. The first table lists the names of the predictors that are estimated as having a linear effect. The second table lists the names of the predictors that are estimated as having a nonlinear effect.

Value

Nothing is returned.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

Examples

```
library(gamselBayes)

# Generate some simple regression-type data:

set.seed(1) ; n <- 1000 ; x1 <- rbinom(n,1,0.5)
x2 <- runif(n) ; x3 <- runif(n) ; x4 <- runif(n)
y <- x1 + sin(2*pi*x2) - x3 + rnorm(n)
Xlinear <- data.frame(x1) ; Xgeneral <- data.frame(x2,x3,x4)

# Obtain a gamselBayes() fit for the data:

fit <- gamselBayes(y,Xlinear,Xgeneral)

# Tabulate the estimated effect types:

effectTypes(fit)
```

---

effectTypesVector — *Obtain the estimated effect types from a Bayesian generalized additive model object*

Description

Extracts the vector of estimated effect types from a gamselBayes() fit object.

Usage

```
effectTypesVector(fitObject)
```

Arguments

`fitObject`	`gamselBayes()` fit object.

Details

The result is a vector of character strings having the same length as the total number of predictors inputted through `Xlinear` and `Xgeneral`. The character strings are one of "linear", "nonlinear" and "zero" according to whether each predictor is estimated as having a linear effect, nonlinear effect of zero effect. The ordering in the returned vector matches that of the columns of `Xlinear` and then the columns of `Xgeneral`.

Value

A vector of character strings having the same length as the number of predictors, which conveys the estimated effect types.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

Examples

```
library(gamselBayes)

# Generate some simple regression-type data:

set.seed(1) ; n <- 1000 ; x1 <- rbinom(n,1,0.5) ;
x2 <- runif(n) ; x3 <- runif(n) ; x4 <- runif(n)
y <- x1 + sin(2*pi*x2) - x3 + rnorm(n)
Xlinear <- data.frame(x1) ; Xgeneral <- data.frame(x2,x3,x4)

# Obtain a gamselBayes() fit for the data:

fit <- gamselBayes(y,Xlinear,Xgeneral)

# Obtain the vector of effect types:

effectTypesVector(fit)
```

---

gamselBayes *Bayesian generalized additive model selection including a fast variational option*

Description

Selection of predictors and the nature of their impact on the mean response (linear versus non-linear) is a fundamental problem in regression analysis. This function uses the generalized additive models framework for estimating predictors effects. An approximate Bayesian inference approach and has two options for achieving this: (1) Markov chain Monte Carlo and (2) mean field variational Bayes.

Usage

```
gamselBayes(y,Xlinear = NULL,Xgeneral = NULL,method = "MCMC",lowerMakesSparser = NULL,
            family = "gaussian",verbose = TRUE,control = gamselBayes.control())
```

Arguments

y	Vector containing the response data. If 'family = "gaussian"' then the response data are modelled as being continuous with a Gaussian distribution. If 'family = "binomial"' then the response data must be binary with 0/1 coding.
Xlinear	Data frame with number of rows equal to the length of y. Each column contains data for a predictor which potentially has a linear or zero effect, but not a nonlinear effect. Binary predictors must be inputted through this matrix.
Xgeneral	A data frame with number of rows equal to the length of y. Each column contains data for a predictor which potentially has a linear, nonlinear or zero effect. Binary predictors cannot be inputted through this matrix.
method	Character string for specifying the method to be used: "MCMC" = Markov chain Monte Carlo, "MFVB" = mean field variational Bayes.
lowerMakesSparser	A threshold parameter between 0 and 1, which is such that lower values lead to sparser fits.
family	Character string for specifying the response family: "gaussian" = response assumed to be Gaussian with constant variance, "binomial" = response assumed to be binary. The default is "gaussian".
verbose	Boolean variable for specifying whether or not progress messages are printed to the console. The default is TRUE.
control	Function for controlling the spline bases, Markov chain Monte Carlo sampling, mean field variational Bayes and other specifications.

Details

Generalized additive model selection via approximate Bayesian inference is provided. Bayesian mixed model-based penalized splines with spike-and-slab-type coefficient prior distributions are used to facilitate fitting and selection. The approximate Bayesian inference engine options are: (1) Markov chain Monte Carlo and (2) mean field variational Bayes. Markov chain Monte Carlo has better Bayesian inferential accuracy, but requires a longer run-time. Mean field variational Bayes is faster, but less accurate. The methodology is described in He and Wand (2021) <arXiv:2201.00412>.

Value

An object of class gamselBayes, which is a list with the following components:

method	the value of method.
family	the value of family.
Xlinear	the inputted design matrix containing predictors that can only have linear effects.

`Xgeneral` the inputted design matrix containing predictors that are potentially have non-linear effects.

`rangex` the value of the control parameter `rangex`.

`intKnots` the value of the control parameter `intKnots`.

`truncateBasis` the value of the control parameter `truncateBasis`.

`numBasis` the value of the control parameter `numBasis`.

`MCMC` a list such that each component is the retained Markov chain Monte Carlo (MCMC)sample for a model parameter. The components are:
`beta0` = overall intercept.
`betaTilde` = linear component coefficients without multiplication by the `gammaBeta` values.
`gammaBeta` = linear component coefficients spike-and-slab auxiliary indicator variables.
`sigmaBeta` = standard deviation of the linear component coefficients.
`rhoBeta` = the Bernoulli distribution probability parameter of the linear component coefficients spike-and-slab auxiliary indicator variables.
`uTilde` = spline basis function coefficients without multiplication by the `gammaUMCMC` values. The MCMC samples are stored in a list. Each list component corresponds to a predictor that is treated as potentially having a non-linear effect, and is a matrix with columns corresponding to the spline basis function coefficients for that predictor and rows corresponding to the retained MCMC samples.
`gammaU` = spline basis coefficients spike-and-slab auxiliary indicator variables. The MCMC samples are stored in a list. Each list component corresponds to a predictor that is treated as potentially having a non-linear effect, and is a matrix with columns corresponding to the spline basis function coefficients for that predictor and rows corresponding to the retained MCMC samples.
`rhoU` = the Bernoulli distribution probability parameters of the spline basis component coefficients spike-and-slab auxiliary indicator variables. The MCMC samples are stored in a matrix. Each column corresponds to a predictor that is treated as potentially having a non-linear effect. The rows of the matrix correspond to the retained MCMC samples.
`sigmaEps` = error standard deviation.

`MFVB` a list such that each component is the mean field variational Bayes approximate posterior density function, or q-density, parameters. The components are:
`beta0` = a vector with 2 entries, consisting of the mean and variance of the Univariate Normal q-density of the overall intercept.
`betaTilde` = a two-component list containing the Multivariate Normal q-density parameters of linear component coefficients without multiplication by the means of the `gammaBeta` q-densities. The list components are: `mu.q.betaTilde`, the mean vector; `Sigma.q.betaTilde`, the covariance matrix.
`gammaBeta` = a vector containing the Bernoulli q-density means of the linear component coefficients spike-and-slab auxiliary indicator variables.
`sigmaBeta` = a vector with 2 entries, consisting of the Inverse Gamma q-density shape and rate parameters of the variance of the linear component coefficients.
`rhoBeta` = a vector with 2 entries, consisting of the Beta q-density shape parameters of the Bernoulli probability parameter of the linear component coefficients spike-and-slab auxiliary indicator variables.

uTilde = a two-component list containing the Multivariate Normal q-density parameters of the spline basis function coefficients without multiplication by the means of the gammaU q-densities. The list components are: mu.q.uTilde, the mean vectors for each predictor that is treated as potentially having a non-linear effect; sigsq.q.uTilde, the diagonal entries of the covariance matrices of each predictor that is treated as potentially having a non-linear effect.
gammaU = a list containing the q-density means of the spline basis coefficients spike-and-slab auxiliary indicator variables. Each list component corresponds to a predictor that is treated as potentially having a non-linear effect.
rhoU = a two-component list with components A.q.rho.u and B.q.rho.u. The A.q.rho.u list component is a vector of Beta q-density first (one plus the power of rho) shape parameters corresponding to the spline basis coefficients spike-and-slab auxiliary indicator variables for each predictor that is treated as potentially having a non-linear effect. The B.q.rho.u list component is a vector of Beta q-density second (one plus the power of 1-rho) shape parameters corresponding to the spline basis coefficients spike-and-slab auxiliary indicator variables for each predictor that is treated as potentially having a non-linear effect.
sigmaEps = a vector with 2 entries, consisting of the Inverse Gamma q-density shape and rate parameters of the error variance.

effectTypeHat	an array of character strings, with entry either "zero", "linear" or "nonlinear", signifying the estimated effect type for each candidate predictor.
meanXlinear	an array containing the sample means of each column of Xlinear.
sdXlinear	an array containing the sample standard deviations of each column of Xlinear.
meanXgeneral	an array containing the sample means of each column of Xgeneral.
sdXgeneral	an array containing the sample standard deviations of each column of Xgeneral.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

References

Chouldechova, A. and Hastie, T. (2015). Generalized additive model selection. <arXiv:1506.03850v2>.

He, V.X. and Wand, M.P. (2021). Bayesian generalized additive model selection including a fast variational option. <arXiv:2021.PLACE-HOLDER>.

Examples

```
library(gamselBayes)

# Generate some simple regression-type data:

set.seed(1) ; n <- 1000 ; x1 <- rbinom(n,1,0.5) ;
x2 <- runif(n) ; x3 <- runif(n) ; x4 <- runif(n)
y <- x1 + sin(2*pi*x2) - x3 + rnorm(n)
Xlinear <- data.frame(x1) ; Xgeneral <- data.frame(x2,x3,x4)

# Obtain a gamselBayes() fit for the data, using Markov chain Monte Carlo:
```

```
fitMCMC <- gamselBayes(y,Xlinear,Xgeneral)
summary(fitMCMC) ; plot(fitMCMC) ; checkChains(fitMCMC)

# Obtain a gamselBayes() fit for the data, using mean field variational Bayes:

fitMFVB <- gamselBayes(y,Xlinear,Xgeneral,method = "MFVB")
summary(fitMFVB) ; plot(fitMFVB)

if (require("Ecdat"))
{
   # Obtain a gamselBayes() fit for data on schools in California, U.S.A.:

   Caschool$log.avginc <- log(Caschool$avginc)
   mathScore <- Caschool$mathscr
   Xgeneral <- Caschool[,c("mealpct","elpct","calwpct","compstu","log.avginc")]

   # Obtain a gamselBayes() fit for the data, using Markov chain Monte Carlo:

   fitMCMC <- gamselBayes(y = mathScore,Xgeneral = Xgeneral)
   summary(fitMCMC) ; plot(fitMCMC) ; checkChains(fitMCMC)

   # Obtain a gamselBayes() fit for the data, using mean field variational Bayes:

   fitMFVB <- gamselBayes(y = mathScore,Xgeneral = Xgeneral,method = "MFVB")
   summary(fitMFVB) ; plot(fitMFVB)
}
```

`gamselBayes.control` *Controlling Bayesian generalized additive model selection*

Description

Function for optional use in calls to `gamselBayes()` to control spline basis dimension, hyperparameter choice and other specifications for generalized additive model selection via using Bayesian inference.

Usage

```
gamselBayes.control(numIntKnots = 25,truncateBasis = TRUE,numBasis = 12,
                    sigmabeta0 = 100000,sbeta = 1000,sepsilon = 1000,
                    su= 1000,rhoBeta = 0.5,rhoU = 0.5,nWarm = 1000,
                    nKept = 1000,nThin = 1,maxIter = 1000,toler = 1e-8,
                    msgCode = 1)
```

Arguments

`numIntKnots`	The number of interior knots used in construction of the Demmler-Reinsch spline basis functions. The value of `numIntKnots` must be an integer between 8 and 50.

`truncateBasis` Boolean flag:
TRUE = truncate the Demmler-Reinsch spline basis to the value specified by `numBasis` (the default)
FALSE = produce a black-and-white version of the graphical check.

`numBasis` The number of spline basis functions retained after truncation when `truncateBasis` is `TRUE`. The value of `numBasis` cannot exceed `numIntKnots+2`. The default value of `numBasis` is 12.

`sigmabeta0` The standard deviation hyperparameter for the Normal prior distribution on the intercept parameter for the standardized version of the data used in Bayesian inference. The default is 100000.

`sbeta` The standard deviation hyperparameter for the Normal prior distribution on the linear component coefficients parameters for the standardized version of the data used in Bayesian inference. The default is 1000.

`sepsilon` The scale hyperparameter for the Half-Cauchy prior distribution on the error standard deviation parameter for the standardized version of the data used in Bayesian inference. The default is 1000.

`su` The scale hyperparameter for the Half-Cauchy prior distribution on the spline basis coefficients standard deviation parameter for the standardized version of the data used in Bayesian inference. The default is 1000.

`rhoBeta` The probability parameter for the Bernoulli prior distribution on the linear component coefficients spike-and-slab auxiliary indicator variables probability parameter. The default is 0.5.

`rhoU` The probability parameter for the Bernoulli prior distribution on the spline basis coefficients spike-and-slab auxiliary indicator variables probability parameter. The default is 0.5.

`nWarm` The size of the Markov chain Monte Carlo warmup, a positive integer, when method is Markov chain Monte Carlo. The default is 1000.

`nKept` The size of the kept Markov chain Monte Carlo samples, a positive integer, when the method is Markov chain Monte Carlo. The default is 1000.

`nThin` The thinning factor for the kept Markov chain Monte Carlo samples, a positive integer, when the method is Markov chain Monte Carlo. The default is 1.

`maxIter` The maximum number of mean field variational Bayes iterations, a positive integer, when the method is mean field variational Bayes. The default is 1000.

`toler` The convergence tolerance for mean field variational Bayes iterations, a positive number less than 0.5, when the method is mean field variational Bayes. Convergence is deemed to have occurred if the relative change in the logarithm of the approximate marginal likelihood falls below `toler`. The default is 1e-8.

`msgCode` Code for the nature of progress messages printed to the screen. If msgCode=0 then no progress messages are printed. If msgCode=1 then a messages printed each time approximately each 10% of the sampling is completed. If msgCode=2 then a messages printed each time approximately each 10% of the sampling is completed. The default is 1.

Value

A list containing values of each of the seventeen control parameters, packaged to supply the `control` argument to `gamselBayes`. The values for `gamselBayes.control` can be specified in the call to `gamselBayes`.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

References

He, V.X. and Wand, M.P. (2021). Generalized additive model selection via Bayesian inference. Submitted.

Examples

```
library(gamselBayes)

# Generate some simple regression-type data:

set.seed(1) ; n <- 1000 ; x1 <- rbinom(n,1,0.5)
x2 <- runif(n) ; x3 <- runif(n) ; x4 <- runif(n)
y <- x1 + sin(2*pi*x2) - x3 + rnorm(n)
Xlinear <- data.frame(x1) ; Xgeneral <- data.frame(x2,x3,x4)

# Obtain a gamselBayes() fit for the data:

fit <- gamselBayes(y,Xlinear,Xgeneral)
summary(fit) ; plot(fit) ; checkChains(fit)

# Now modify some of the control values:

fitControlled <- gamselBayes(y,Xlinear,Xgeneral,control = gamselBayes.control(
                             numIntKnots = 35,truncateBasis = FALSE,
                             sbeta = 10000,su = 10000,nWarm = 2000,nKept = 1500))
summary(fitControlled) ; plot(fitControlled) ; checkChains(fitControlled)
```

---

gamselBayesUpdate *Update a* `gamselBayes()` *fit object.*

Description

Facilitates updating of `gamselBayes` fit object when two key parameters controlling model selection are modified. Use of `gamselBayesUpdate()` allows for fast tweaking of such parameters without another, potentially time-consuming, call to `gamselBayes()`.

Usage

```
gamselBayesUpdate(fitObject,lowerMakesSparser = NULL)
```

Arguments

fitObject	gamselBayes() fit object.
lowerMakesSparser	A threshold parameter between 0 and 1, which is such that lower values lead to sparser fits.

Details

The gamselBayesUpdate() function is applicable when a gamselBayes() fit object has been obtained for particular data inputs y, Xlinear and Xgeneral (as well as other tuning-type inputs) and the analyst is interested in changing the value of the parameter that controls model selection. This parameter is named lowerMakesSparser, and is described above. A call to gamselBayesUpdate() with a new value of lowerMakesSparse produces an updated gamselBayes() fit object with, potentially, different effect type estimates.

Value

An object of class gamselBayes with the same components as those produced by the gamselBayes() function. See help(gamselBayes) for details.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

Examples

```
library(gamselBayes)

# Generate some regression-type data:

set.seed(1) ;  n <- 5000  ; numPred <- 15
Xgeneral <- as.data.frame(matrix(runif(n*numPred),n,numPred))
names(Xgeneral) <- paste("x",1:numPred,sep="")

y <- as.vector(0.1 + 0.4*Xgeneral[,1] - 2*pnorm(3-6*Xgeneral[,2])
               - 0.9*Xgeneral[,4] + cos(3*pi*Xgeneral[,5]) + 2*rnorm(n))

# Obtain and assess a gamselBayes() fit:

fitOrig <- gamselBayes(y,Xgeneral = Xgeneral)
summary(fitOrig) ; plot(fitOrig)
print(fitOrig$effectTypesHat)

# Update the gamselBayes() fit object with a new value of
# the "lowerMakesSparser" parameter:

fitUpdated <-  gamselBayesUpdate(fitOrig,lowerMakesSparser = 0.6)
summary(fitUpdated) ; plot(fitUpdated)
print(fitUpdated$effectTypesHat)
```

gamselBayesVignette *Display the package's vignette.*

Description

The vignette of the gamselBayes package is displayed using the default PDF file browser. It provides a detailed description of use of the package for Bayesian generalized additive model selection.

Usage

```
gamselBayesVignette()
```

Value

Nothing is returned.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

Examples

```
gamselBayesVignette()
```

plot.gamselBayes *Plot components of the selected generalized additive model from a* gamselBayes() *fit*

Description

The estimated non-linear components of the generalized additive model selected via gamselBayes are plotted.

Usage

```
## S3 method for class 'gamselBayes'
plot(x,credLev = 0.95,gridSize = 251,nMC = 5000,varBand = TRUE,
    shade = TRUE,yscale = "response",rug = TRUE,rugSampSize = NULL,estCol = "darkgreen",
     varBandCol = NULL,rugCol = "dodgerblue",mfrow = NULL,xlim = NULL,ylim = NULL,
    xlab = NULL,ylab = NULL,mai = NULL,pages = NULL,cex.axis = 1.5,cex.lab = 1.5,...)
```

Arguments

x	A gamselBayes() fit object.
credLev	A number between 0 and 1 such that the credible interval band has (100*credLev)% approximate pointwise coverage. The default value is 0.95.
gridSize	A number of grid points used to display the density estimate curve and the pointwise credible interval band. The default value is 251.
nMC	The size of the Monte Carlo sample, a positive integer, for carrying out approximate inference from the mean field variational Bayes-approximate posterior distributions when the method is mean field variational Bayes. The default value is 5000.
varBand	Boolean flag specifying whether or not a variability band is included: TRUE = add a pointwise approximate (100*credLev)% credible set variability band (the default) FALSE = only plot the density estimate, without a variability band.
shade	Boolean flag specifying whether or not the variability band is displayed using shading: TRUE = display the variability band using shading (the default) FALSE = display the variability band using dashed curves.
yscale	Character string specifying the vertical axis scale for display of estimated non-linear functions: "response" = display on the response scale) "link" = display on the link scale.
rug	Boolean flag specifying whether or not rug-type displays for predictor data are used: TRUE = show the predictor data using rug-type displays (the default) FALSE = do not show the predictor data.
rugSampSize	The size of the random sample sample of each predictor to be used in rug-type displays.
estCol	Colour of the density estimate curve. The default value is "darkgreen".
varBandCol	Colour of the pointwise credible interval variability band. If shade=TRUE then the default value is "palegreen". If shade=FALSE then the default value is estCol.
rugCol	Colour of rug plot that shows values of the predictor data. The default value is "dodgerblue".
mfrow	An optional two-entry vector for specifying the layout of the nonlinear fit displays.
xlim	An optional two-column matrix for specification of horizontal frame limits in the plotting of the estimated non-linear predictor effects. The number of rows in xlim must equal length(effectTypesVector(x)=="nonlin"). If any of the rorows of xlim contain NA then the default horizontal frame limits value for that predictor is used. Therefore, if there are several predictors selected as non-linear and horizontal frame adjustments are required for a few of them then then xlim can be a matrix with mainly NA values, and with non-NA frame specification limits in the relevant rows.

ylim	The same as xlim, except for vertical frame limits.
xlab	An optional vector of character strings containing labels for the horizontal axes. The number of entries in xlab must equal length(effectTypesVector(x)=="nonlin").
ylab	The same as xlab, except for vertical axis labels.
mai	An optional numerical vector of length 4 for specification of inner margin dimensions of each panel, ordered clockwise from below the panel.
pages	An optional positive integer that specifies the number of pages used to display the non-linear function estimates.
cex.axis	An optional positive scalar value for specification of the character expansion factor for the axis values.
cex.lab	An optional positive scalar value for specification of the character expansion factor for the labels.
...	Place-holder for other graphical parameters.

Details

The estimated non-linear components of the selected generalized additive model are plotted. Each plot corresponds to a slice of the selected generalized additive model surface with all other predictors set to their median values. Pointwise credible intervals unless varBand is FALSE.

Value

Nothing is returned.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

Examples

```
library(gamselBayes)

# Generate some simple regression-type data:

set.seed(1) ; n <- 1000 ; x1 <- rbinom(n,1,0.5) ;
x2 <- runif(n) ; x3 <- runif(n) ; x4 <- runif(n)
y <- x1 + sin(2*pi*x2) - x3 + rnorm(n)
Xlinear <- data.frame(x1) ; Xgeneral <- data.frame(x2,x3,x4)

# Obtain a gamselBayes() fit for the data and print out a summary:

fit <- gamselBayes(y,Xlinear,Xgeneral)
summary(fit)

# Plot the predictor effect(s) estimated as being non-linear:

plot(fit)

# Plot the same fit(s) but with different colours and style:
```

```
plot(fit,shade = FALSE,estCol = "darkmagenta",varBandCol = "plum",
     rugCol = "goldenrod")

if (require("Ecdat"))
{
   # Obtain a gamselBayes() fit for data on schools in California, U.S.A.:

   Caschool$log.avginc <- log(Caschool$avginc)
   mathScore <- Caschool$mathscr
   Xgeneral <- Caschool[,c("mealpct","elpct","calwpct","compstu","log.avginc")]
   fit <- gamselBayes(y = mathScore,Xgeneral = Xgeneral)
   summary(fit)

   # Plot the predictor effect(s) estimated as being non-linear:

   plot(fit)
}
```

`predict.gamselBayes` *Obtain predictions from a* `gamselBayes()` *fit*

Description

The estimated non-linear components of the generalized additive model selected via `gamselBayes` are plotted.

Usage

```
## S3 method for class 'gamselBayes'
predict(object,newdata,type = "response",...)
```

Arguments

`object`	A `gamselBayes()` fit object.
`newdata`	A two-component list the following components: A data frame containing new data on the predictors that are only permitted to have a linear or zero effect and, if not NULL, must have the same names as the `Xlinear` component of `object`. A data frame containing new data for the predictors that are permitted to have a linear, nonlinear or zero effect and, if not NULL, must have the same names as the `Xgeneral` component of `object`. If both `Xlinear` and `Xgeneral` are not NULL then they must have the same numbers of rows.
`type`	A character string for specifying the type of prediction, with the following options: "link", "response" or "terms", which leads to the value as described below.
`...`	A place-holder for other prediction parameters.

Value

A vector or data frame depending on the value of type:
If type="link" then the value is a vector of linear predictor-scale fitted values.
If type="response" and family="binomial" then the value is a vector of probability-scale fitted values. Otherwise (i.e. family="binomial") the value is the vector of predictor-scale fitted values.
If type="terms" then the value is a a data frame with number of columns equal to the total number of predictors. Each column is the contribution to the vector of linear predictor-scale fitted values from each predictor. These contributions do not include the intercept predicted value. The intercept predicted value is included as an attribute of the returned data frame.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

Examples

```
library(gamselBayes)

# Generate some simple regression-type data:

n <- 1000 ; x1 <- rbinom(n,1,0.5) ; x2 <- runif(n) ; x3 <- runif(n) ; x4 <- runif(n)
y <- x1 + sin(2*pi*x2) - x3 + rnorm(n)
Xlinear <- data.frame(x1) ; Xgeneral <- data.frame(x2,x3,x4)
names(Xlinear) <- c("x1") ; names(Xgeneral) <- c("x2","x3","x4")

# Obtain and summarise a gamselBayes() fit for the data:

fit <- gamselBayes(y,Xlinear,Xgeneral)
summary(fit)

# Obtain some new data:

nNew <- 10
x1new <- rbinom(nNew,1,0.5) ; x2new <- runif(nNew) ; x3new <- runif(nNew)
x4new <- runif(nNew)
XlinearNew <- data.frame(x1new) ; names(XlinearNew) <- "x1"
XgeneralNew <- data.frame(x2new,x3new,x4new)
names(XgeneralNew) <- c("x2","x3","x4")

newdataList <- list(XlinearNew,XgeneralNew)

# Obtain predictions at the new data:

predObjDefault <- predict(fit,newdata=newdataList)
print(predObjDefault)
```

`summary.gamselBayes` *Summarise components of the selected generalized additive model from a* `gamselBayes()` *fit*

Description

Inference summaries of the estimated linear component coefficients of the generalized additive model selected via gamselBayes are tabulated.

Usage

```
## S3 method for class 'gamselBayes'
summary(object,credLev = 0.95,sigFigs = 5,nMC = 10000,...)
```

Arguments

object	A gamselBayes() fit object.
credLev	A number between 0 and 1 such that the credible interval band has (100*credLev)% approximate pointwise coverage. The default value is 0.95.
sigFigs	The number of significant figures used for the entries of the summary table.
nMC	The size of the Monte Carlo sample, a positive integer, for carrying out approximate inference from the mean field variational Bayes-approximate posterior distributions when the method is mean field variational Bayes. The default value is 10000.
...	Place-holder for other summary parameters.

Details

If the selected generalized additive model has at least one predictor having a linear effect then a data frame is returned. The columns of the data correspond to posterior means and credible interval limits of the linear effects coefficients.

Value

A data frame containing linear effect Bayesian inferential summaries.

Author(s)

Virginia X. He <virginia.x.he@student.uts.edu.au> and Matt P. Wand <matt.wand@uts.edu.au>

Examples

```
library(gamselBayes)

# Generate some simple regression-type data:

set.seed(1) ; n <- 1000 ; x1 <- rbinom(n,1,0.5) ;
x2 <- runif(n) ; x3 <- runif(n) ; x4 <- runif(n)
y <- x1 + sin(2*pi*x2) - x3 + rnorm(n)
Xlinear <- data.frame(x1) ; Xgeneral <- data.frame(x2,x3,x4)

# Obtain a gamselBayes() fit for the data and print out a summary:

fit <- gamselBayes(y,Xlinear,Xgeneral)
```

```
summary(fit)

# Print the summary with different values of some of the arguments:

summary(fit,credLev=0.99,sigFigs=3)

if (require("Ecdat"))
{
   # Obtain a gamselBayes() fit for data on schools in California, U.S.A.:

   Caschool$log.avginc <- log(Caschool$avginc)
   mathScore <- Caschool$mathscr
   Xgeneral <- Caschool[,c("mealpct","elpct","calwpct","compstu","log.avginc")]
   fit <- gamselBayes(y = mathScore,Xgeneral = Xgeneral)
   summary(fit)
}
```

Index