Package: spikeSlabGAM (via r-universe)

October 29, 2026

Version 1.1-20

Title Bayesian Variable Selection and Model Choice for Generalized Additive Mixed Models

Depends R (>= 2.15.1)

Imports coda, cluster, ggplot2 (>= 2.0.0), gridExtra (>= 2.0.0), interp, MASS, MCMCpack, mvtnorm, R2WinBUGS, reshape, scales, splines, parallel

Suggests mboost, mlbench, gtable, knitr, rmarkdown

Description Bayesian variable selection, model choice, and regularized estimation for (spatial) generalized additive mixed regression models via stochastic search variable selection with spike-and-slab priors.

License MIT + file LICENSE

URL https://github.com/fabian-s/spikeSlabGAM

Collate 'utils.R' 'spikeAndSlab.R' 'terms.R' 'ssGAMDesign.R' 'spikeSlabGAM.R' 'bugs.R' 'plot.R' 'predict.R' 'summary.R'

RoxygenNote 7.1.2

VignetteBuilder knitr

NeedsCompilation yes

Author Fabian Scheipl [aut, cre], Bettina Gruen [ctb]

Maintainer Fabian Scheipl <fabian.scheipl@stat.uni-muenchen.de>

Repository https://cran.r-universe.dev

Date/Publication 2024-10-22 17:30:02 UTC

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

RemoteRef HEAD

RemoteSha 60e6318800efdef9a195cf8a446672ac2d70ec10

Contents

evalTerm . . . 2
fct . . . 3
getPosteriorTerm . . . 4
lin . . . 4
mrf . . . 5
plot.spikeSlabGAM . . . 6
plotTerm . . . 8
predict.spikeSlabGAM . . . 9
print.summary.spikeSlabGAM . . . 10
rnd . . . 11
sm . . . 12
spikeAndSlab . . . 13
spikeSlabGAM . . . 16
srf . . . 19
ssGAM2Bugs . . . 21
ssGAMDesign . . . 21
summary.spikeSlabGAM . . . 23
u . . . 24

Index **25**

`evalTerm` *Get summaries of the posterior (predictive) distribution of the linear predictor of a model term*

Description

Get summaries of the posterior (predictive) distribution of the linear predictor of a model term

Usage

```
evalTerm(
  label,
  model,
  newdata = NULL,
  aggregate = mean,
  quantiles = c(0.1, 0.9),
  returnData = TRUE
)
```

Arguments

`label`	(character) the label of one of the terms in `model`.
`model`	a `spikeSlabGAM` object
`newdata`	`data.frame` on which to evaluate `label`. Defaults to NULL, in which case the term is evaluated on the original data.

aggregate	(function) a summary statistic that is applied over the mcmc-samples of the linear predictor. Defaults to mean.
quantiles	(numeric) a vector of quantiles for borders of credible regions of the linear predictor. Defaults to 10 and 90 percent quantiles, i.e. a (point-wise) 80 percent credible region.
returnData	should the relevant original variables be included in the returned data.frame? Defaults to TRUE.

Value

A data.frame that contains the relevant variables from newdata (if returnData is TRUE), the aggregate-summary and the requested quantiles.

Author(s)

Fabian Scheipl

---

fct *Generate design for a factor covariate*

Description

Generate design for a factor covariate

Usage

```
fct(x, contr = "contr.sum")
```

Arguments

x	a covariate that can be treated as a factor
contr	(character) name of the contrasts associated with the factor. Defaults to "contr.sum".

Value

design matrix for a factor

Author(s)

Fabian Scheipl

getPosteriorTerm *Get the posterior distribution of the linear predictor of a model term*

Description

Get the posterior distribution of the linear predictor of a model term

Usage

```
getPosteriorTerm(label = NULL, model, betaInd = NULL)
```

Arguments

label	(character) one of the terms in model
model	a spikeSlabGAM object
betaInd	(optional) the column indices of the part of the design matrix for which the linear predictor is to be evaluated.

Value

a matrix containing the samples of the linear predictor associated with label; with attribute 'coef' that contains the posterior samples of the associated coefficients.

Author(s)

Fabian Scheipl

lin *Generate orthogonal polynomial base for a numeric covariate without intercept*

Description

Generate orthogonal polynomial base for a numeric covariate without intercept

Usage

```
lin(x, order = 1)
```

Arguments

x	covariate
order	order of the polynomial, defaults to 1

Value

an orthogonal basis for a centered polynomial function in x

Author(s)

Fabian Scheipl

`mrf` *Generate design for a 2-D Gaussian Markov Random Field*

Description

The returned design is (a low-rank approximation to) the matrix square root of the implied covariance of the centered MRF. The function stops if 'islands', i.e. regions without any neighbors are found. Regions without observations have to be removed from the neighborhood matrix and there is currently no `predict`-functionality for regions without observations in the original data.

Usage

```
mrf(x, N, decomposition = c("ortho", "MM"), tol = 1e-10, rankZ = 0.995)
```

Arguments

`x`	a factor: which observation belongs to which region
`N`	the neighborhood (adjacency) matrix: a symmetric matrix with one column/row for every level of `x`, defining the neighborhood structure (either 0-1 or with positive weights, e.g. based on shared boundary length or centroid distances). Has to have rownames and column names that correspond to the levels of `x`, the function checks whether the rows/columns are in the same order as the levels of `x`. Entries on the diagonal are ignored.
`decomposition`	use a (truncated) spectral decomposition of the implied prior covariance of $f(x)$ for a low rank representation with orthogonal basis functions and i.i.d. coefficients (`"ortho"`), or use the mixed model reparameterization for non-orthogonal basis functions and i.i.d. coefficients (`"MM"`). Defaults to `"MM"`.
`tol`	count singular/eigenvalues smaller than this as zero
`rankZ`	how many eigenvectors to retain from the eigen decomposition: either a number > 3 or the proportion of the sum of eigenvalues the retained eigenvectors must represent at least. Defaults to .999.

Value

a transformed design matrix for the Markov Random Field

Author(s)

Fabian Scheipl

References

Fahrmeir, L., Lang, S. (2001) Bayesian inference for generalized additive mixed models based on Markov random field priors. *Applied Statistics*, **50**(2):201–220.

`plot.spikeSlabGAM` *Generates graphical summaries of a fitted model*

Description

This function plots the estimated linear predictors of the terms in a model on a grid of values. By default displays all 3-way, 2-way interactions and main effects present in the model. Starting with ggplot-0.9.2 these are no longer aligned by their axes due to internal changes in grid and ggplot2. Uses **gridExtra**'s `marrangeGrob` to arrange the plots for the terms, also over multiple pages if necessary. This means the graphics device type is temporarily set to the value of `interactive.dev` in interactive use in RStudio if necessary since the `RStudioGD` graphical device does not support opening multiple pages.

Usage

```
## S3 method for class 'spikeSlabGAM'
plot(
  x,
  labels = NULL,
  cumulative = TRUE,
  commonEtaScale = FALSE,
  aggregate = mean,
  quantiles = c(0.1, 0.9),
  gridlength = 20,
  base_size = 12,
  ggElems = list(),
  nrow = min(ceiling(sqrt(length(plotList))), 3),
  ncol = min(ceiling(length(plotList)/nrow), 3),
  interactive.dev = c("x11", "quartz", "windows"),
  ...
)
```

Arguments

`x`	a fitted `spikeSlabGAM` model
`labels`	a character vector of names of model terms to be plotted
`cumulative`	Defaults to TRUE, in which case all lower order terms that are involved in an interaction are cumulated and then plotted (e.g, if a model contains 2 smooth effects and their interaction, ONLY the sum of the marginal smooth and linear terms and all their interactions are plotted.) If FALSE, a separate plot for every term in the model is produced.

commonEtaScale	use the same limits for all vertical axes of the different panels? Defaults to FALSE. Can be useful to compare effect sizes more easily between panels, but tends to mess up the scales.
aggregate	(function) which summary statistic to use for the posterior of the model terms. Defaults to the mean.
quantiles	which quantiles to use for the borders of credible regions. Defaults to 10% and 90% percentiles. Set to NULL to omit plotting credible regions.
gridlength	length of the (univariate) grids for the covariates on which to evaluate the posterior. Defaults to 20.
base_size	default base font size for plot (see e.g. theme_gray)
ggElems	a list of plot elements to give to ggplot. Use this to supply custom themes or colors, fortify the plot(s) with partial residuals etc. Defaults to an empty list. Unless specified differently here, the default ggplot-theme (theme_gray) is changed to a white background with major gridlines in gray ('grey95'), no minor grid lines, and smaller text for the legends.
nrow	number of rows per page, defaults to min(sqrt(no. of plots), 3). See marrangeGrob.
ncol	number of columns per page, defaults to min((no. of plots)/nrow, 3). See marrangeGrob.
interactive.dev	alternative device to use in interactive mode in RStudio if output needs to be spread on multiple pages, since the RStudio graphical device does not support opening multiple displays.
...	arguments passed to marrangeGrob.

Value

a list of ggplot-objects (invisible)

Note

Note that cumulative = TRUE will only find all relevant terms to accumulate if, for all numeric covariates that have a smooth term, the smooth term is specified *after* the linear term in the formula.

Author(s)

Fabian Scheipl

See Also

plotTerm for more details on the specific plots

Examples

```
#see ?spikeSlabGAM
```

plotTerm *Plot the estimated effect of a model term.*

Description

Plots the estimated linear predictor for a model term, i.e a main effect or a (2- or 3-way) interaction. Plots for the joint effect of two numerical covariates show an overlay if quantiles were specified: Regions where the pointwise credible intervals do not contain zero are plotted in muted red (> 0) and blue (< 0), overlaid by coloured contour lines that show the `aggregate` values. Contour lines are shown only inside the convex hull of the original observations. Plots for `srf:lin` terms show the spatially varying coefficient, i.e. the contour lines represent the change in the linear predictor when the `lin`-covariate increases by 1 standard deviation. For this reason, a cumulative plot makes no sense and the routine will set `cumulative = FALSE` with a warning.

Usage

```
plotTerm(
  label,
  m,
  cumulative = TRUE,
  aggregate = mean,
  quantiles = c(0.1, 0.9),
  gridlength = 40,
  contours = 30,
  ggElems = list()
)
```

Arguments

`label`	(character) the label of the effect to be plotted.
`m`	a fitted `spikeSlabGAM` model
`cumulative`	Defaults to TRUE, in which case the lower order terms that are associated with the covariates involved in `label` are cumulated and then plotted. (e.g, if `label` denotes a smoooth term, the sum of the linear and smooth effect is displayed if TRUE, if `label` is a factor-factor interaction, the sum of both main effects and their interaction is displayed etc.) If FALSE, only the marginal effect of `label` is displayed.
`aggregate`	(function) which summary statistic to use for the posterior of the effect. Defaults to the mean.
`quantiles`	which quantiles to use for the borders of credible regions. Defaults to 10 regions. Cannot deal with more than two quantiles.
`gridlength`	length of the (univariate) grids on which to evaluate the posterior.
`contours`	use how many contour lines for the joint effect of two numerical covariates. Defaults to 30.
`ggElems`	a list of plot elements to give to `ggplot`. Use this to supply custom themes, colour schemes etc. Defaults to an empty list, which yields the standard settings.

Details

Limitations: Plots for 4-way (or higher) interactions are not implemented. Requesting such a plot will return the NULL object with a warning. Plots for `mrf` just treat the grouping variable as a conventional factor i.e. will not incorporate any neighborhood or map information.

Value

an object of class `ggplot`. Use `print` or wrap the call to `plotTerm` in parentheses to directly render on a graphic device.

Author(s)

Fabian Scheipl

Examples

```
#see help for spikeSlabGAM
```

`predict.spikeSlabGAM` *Obtain posterior predictive/credible intervals from a spike-and-slab model*

Description

Obtain posterior predictive/credible intervals from a spike-and-slab model

Usage

```
## S3 method for class 'spikeSlabGAM'
predict(
  object,
  newdata = NULL,
  type = c("response", "link", "terms"),
  terms = NULL,
  aggregate = mean,
  quantiles = NULL,
  addIntercept = is.null(terms),
  ...
)
```

Arguments

`object`	a `spikeSlabGAM` model
`newdata`	an optional `data.frame` on which to evaluate predictions. Defaults to NULL, in which case the fitted values for the original data are returned

type	the type of prediction required. The default is on the scale of response, the alternative 'link' is on the scale of the linear predictor. Specifying 'terms' returns a list giving the linear predictors of the terms specified in terms.
terms	an optional character vector of term labels or variable names for which to return fits/predictions/credible regions. If variable names instead of term labels are supplied, the function returns predictions/estimates for *all* terms associated with these variables, i.e. their main effects (usually both linear and smooth for numeric covariates) and all interaction terms they are involved in.
aggregate	(function) the summary statistic of the posterior predictive of the linear predictor. Defaults to mean.
quantiles	(numeric) an optional vector of quantiles for borders of credible regions of the returned values. Defaults to NULL.
addIntercept	include global intercept term in prediction/estimate? Defaults to TRUE if terms = NULL.
...	arguments passed from or to other methods (not used)

Value

If type ="terms", a list of data.frames containing the requested pointwise summary statistics for the supplied terms (use e.g. Reduce("+", ...) to get row-wise sums of the list-entries). Otherwise, a data.frame containing the requested pointwise summary statistics of the posterior predictive of the linear predictor (type ="link") or the conditional expectation of the response (type ="response") is returned.

Author(s)

Fabian Scheipl

---

print.summary.spikeSlabGAM

Print summary for posterior of a spikeSlabGAM *fit*

Description

The model table shows at least the 10 and at most the 20 most probable models as found by SSVS, or enough models to account for 90% of the probability mass in the model space as found by SSVS by default.

Usage

```
## S3 method for class 'summary.spikeSlabGAM'
print(
  x,
  digits = 3,
  printPGamma = TRUE,
  printModels = TRUE,
```

```
  showModels = NULL,
  ...
)
```

Arguments

x	an object of class summary.spikeSlabGAM
digits	see options, defaults to 3
printPGamma	print marginal inclusion probabilities and norm of the linear predictor for each term? Defaults to TRUE
printModels	print most probable models visited by the SSVS? Defaults to TRUE
showModels	how many of th most probable models to display. See details for default.
...	arguments based from or to other methods (not used)

Value

invisibly returns x

Author(s)

Fabian Scheipl

See Also

summary.spikeSlabGAM

---

rnd *Generate design for a random intercept*

Description

Generate design for a random intercept

Usage

```
rnd(x, C = NULL)
```

Arguments

x	a covariate that can be treated as a factor
C	an optional known correlation structure of the random effects associated with x

Value

design matrix for a random intercept for x

Author(s)

Fabian Scheipl

sm *Generate a reparameterized P-spline base*

Description

The returned matrix is a low-rank approximation of the original P-spline basis (unless `decomposition = "asIs"`), that is projected into the complement of the nullspace of the associated penalty (unless `centerBase = FALSE`), i.e. for the default second order difference penalty, the resulting basis cannot reproduce linear or constant functions and parameterizes the "wiggly" part of the influence of `x` only. This means that it very rarely makes sense to run a model with `sm(x)` without also using `lin(x)` or `u(x)`. The projection improves the separability between the linear and smooth parts of the influence of `x` and centers the resulting function estimates s.t $\sum_i f(x_i) = 0$.

Usage

```
sm(
  x,
  K = min(length(unique(x)), 20),
  spline.degree = 3,
  diff.ord = 2,
  rankZ = 0.999,
  centerBase = T,
  centerx = x,
  decomposition = c("ortho", "MM", "asIs"),
  tol = 1e-10
)
```

Arguments

`x`	covariate
`K`	number of basis functions in the original basis (defaults to 20)
`spline.degree`	defaults to 3 for cubic B-plines
`diff.ord`	order of the difference penalty, defaults to 2 for penalizing deviations from linearity
`rankZ`	how many eigenvectors to retain from the eigen decomposition: either a number > 3 or the proportion of the sum of eigenvalues the retained eigenvectors must represent at least. Defaults to .999.
`centerBase`	project the basis of the penalized part into the complement of the column space of the basis of the unpenalized part? defaults to TRUE
`centerx`	vector of x-values used for centering (defaults to `x`)
`decomposition`	use a truncated spectral decomposition of the implied prior covariance of $f(x)$ for a low rank representation with orthogonal basis functions and i.i.d. coefficients (`"ortho"`), or use the mixed model reparameterization for non-orthogonal basis functions and i.i.d. coefficients (`"MM"`) or use basis functions as they are with i.i.d. coefficients (`"asIs"`). Defaults to `"ortho"`.
`tol`	count eigenvalues smaller than this as zero

Value

a basis for a smooth function in x

Author(s)

Fabian Scheipl

References

Kneib, T. (2006). Mixed model based inference in structured additive regression. Dr. Hut. https://edoc.ub.uni-muenchen.de/archive/00005011/

`spikeAndSlab` *Set up and sample a spike-and-slab prior model.*

Description

This function sets up a spike-and-slab model for variable selection and model choice in generalized additive models and samples its posterior. It uses a blockwise Metropolis-within-Gibbs sampler and the redundant multiplicative parameter expansion described in the reference. This routine is not meant to be called directly by the user – `spikeSlabGAM` provides a formula-based interface for specifying models and takes care of (most of) the housekeeping. Sampling of the chains is done in parallel using package `parallel`. A "SOCK" cluster is set up under Windows to do so (and closed after computations are done, I try to clean up after myself), see `makeCluster` etc. Use `options(mc.cores =<foo>)` to set the (maximal) number of processes forked by the parallelization. If `options()$mc.cores` is unspecified, it is set to 2.

Usage

```
spikeAndSlab(
  y,
  X,
  family = c("gaussian", "binomial", "poisson"),
  hyperparameters = list(),
  model = list(),
  mcmc = list(),
  start = list()
)
```

Arguments

`y`	response
`X`	design matrix
`family`	(character) the family of the response, defaults to normal/Gaussian response
`hyperparameters`	a list of hyperparameters controlling the priors (see details)

`model`	a list with information about the grouping structure of the model (see details)
`mcmc`	(optional) list setting arguments for the sampler (see details)
`start`	(optional) list containing the starting values for $\beta, \gamma, \tau^2, \sigma^2, w$ and, optionally, the random seed

Details

Details for model specification:

`hyperparameters w` hyperparameters for the $Beta$-prior for w; defaults to `c(alphaW = 1, betaW = 1)`, i.e. a uniform distribution.

`tau2` hyperparameters for the Γ^{-1}-prior of the hypervariances τ^2; defaults to `c(a1 = 5, a2 = 25)`

`gamma` sets v_0, the ratio between the spike and slab variances, defaults to `c(v0 = 0.00025)`

`sigma2` hyperparameters for Γ^{-1}-prior for error variance; defaults to `c(b1 = 1e-4, b2 = 1e-4)`. Only relevant for Gaussian response.

`varKsi` variance for prior of ξ, defaults to 1

`ksiDF` defaults to 0 for a gaussian prior for ξ, else induces a t-prior for $\xi$with `ksiDF` degrees of freedom.

`model groupIndicators` a factor that maps the columns of X to the different model terms

`H` a matrix containing the hierarchy of the penalized model terms

`n` number of observations

`q` length of β

`scale` scale/weights of the response, defaults to `rep(1, n)`, use this to specify number of trials for binomial data

`offset` defaults to `rep(0,n)`

`mcmc nChains` how many parallel chains to run: defaults to 3

`chainLength` how many samples should be generated per chain, defaults to 500

`burnin` how many initial iterations should be discarded, defaults to 100

`thin` save only every `thin`-th iteration, defaults to 5

`verbose` verbose output and report progress? defaults to TRUE

`returnSamples` defaults to TRUE

`sampleY` generate samples of y and its conditional expectation from posterior predictive? defaults to FALSE

`useRandomStart` use random draw or ridge estimate for beta as starting value? defaults to TRUE, i.e. random starting values.

`blocksize` approx. blocksizes of the updates for α, ξ. Defaults to 50 for gaussian responses and 5/15 for non-gaussian responses.

`scalemode` how to do term-wise rescaling of subvectors of ξ in each iteration: 0 means no rescaling, 1 means rescaling s.t. each mean$(|\xi_g|) = 1$, 2 means rescaling s.t. each max$(|\xi_g|) = 1$

`modeSwitching` probability to do P-IWLS with the mode of the proposal set to the current value, which is useful if the chain gets stuck. Defaults to 0.05. Increase this if acceptance rates are too low.

`reduceRet` don't return data and samples for α, ξ, τ^2? defaults to FALSE

`start beta` starting values for β. Defaults to a modified approximate ridge-penalized ML estimate. See vignette for details on default specification.

`gamma` starting values for γ. Defaults to a vector of 1's if `mcmc$useRandomStart` is `FALSE`, otherwise drawn from the prior.

`tau2` starting values for τ^2. Defaults to the mode of the prior if `mcmc$useRandomStart` is `FALSE`, otherwise drawn from the prior.

`sigma2` starting values for σ^2. Only relevant for Gaussian response. Defaults to the variance of the response divided by the number of covariates if `mcmc$useRandomStart` is `FALSE`, otherwise drawn from the prior.

`w` starting value for w. Defaults to the mean of the prior if `mcmc$useRandomStart` is `FALSE`, otherwise drawn from the prior.

`seed` Sets RNG seed for reproducible results. Parallel chains are seeded with this seed incremented by the number of the chain.

Value

a list with components:

`formula` see arguments

`data` see arguments

`family` see arguments

`y` see arguments

`X` see arguments

`hyperparameters` see arguments

`model` see arguments

`mcmc` see arguments

`start` see arguments

`posteriorPred` a list with entries `mu` and `y` containing samples of the expected values and realizations of the response from the posterior predictive

`postMeans` a list containing the posterior means of the parameters:

`beta` the regression coefficients

`alpha`

`ksi`

`tau` hypervariances of the penalized model terms

`gamma` inclusion indicator variables of the model terms

`pV1` $P(\gamma = 1)$

`w` hyperparameter for `gamma`

`sigma2` error variance (for Gaussian data)

`logLik` log likelihood

`logPost` log of (unnormalized) posterior

`samples` a list containing the posterior samples of the parameters, see above for explanation of the entries

`DIC` a vector with $DIC, pD, \bar{D}, \hat{D}$. Usually doesn't make much sense for this kind of model because of the posterior's multimodality.

`fitted` a matrix with the posterior mean of the linear predictor in the first column and the posterior mean of the expected response in the second.

`runTime` of the sampler, in seconds

Author(s)

Fabian Scheipl, Daniel Sabanes Bove

References

Scheipl, F. (2010) Normal-Mixture-of-Inverse-Gamma Priors for Bayesian Regularization and Model Selection in Structured Additive Regression Models. *LMU Munich, Department of Statistics*: Technical Reports, No.84 (https://epub.ub.uni-muenchen.de/11785/)

`spikeSlabGAM` *Generate posterior samples for a GAMM with spike-and-slab priors*

Description

This function fits structured additive regression models with spike-and-slab priors via MCMC. The spike-and-slab prior results in an SSVS-like term selection that can be used to aid model choice, e.g. to decide between linear and smooth modelling of numeric covariates or which interaction effects are relevant. Model terms can be factors (`fct`), linear/polynomial terms (`lin`), uni- or bivariate splines (`sm`, `srf`), random intercepts (`rnd`) or Markov random fields (`mrf`) and their interactions, i.e. an interaction between two smooth terms yields an effect surface, an interaction between a linear term and a random intercept yields random slopes, an interaction between a linear term and a smooth term yields a varying coefficient term etc.

Usage

```
spikeSlabGAM(
  formula,
  data,
  ...,
  family = "gaussian",
  hyperparameters = list(),
  model = list(),
  mcmc = list(),
  start = list()
)
```

Arguments

formula	the model formula (see details below and ssGAMDesign).
data	the data containing the variables in the function
...	additional arguments for ssGAMDesign
family	(character) the family of the response, defaults to normal/Gaussian response, "poisson" and "binomial" are implemented as well.
hyperparameters	A list of arguments specifying the prior settings. See spikeAndSlab.
model	A list of arguments describing the model structure. See spikeAndSlab. User-supplied groupIndicators and H entries will be overwritten by ssGAMDesign.
mcmc	A list of arguments specifying MCMC sampler options. See spikeAndSlab.
start	A list of starting values for the MCMC sampler. See spikeAndSlab. Use start = list(seed =<YOUR_SEED>) to set the RNG seed for reproducible results.

Details

Implemented types of terms:

u unpenalized model terms associated with a flat prior (no selection)

lin linear/polynomial terms

fct factors

sm univariate smooth functions

rnd random intercepts

mrf Markov random fields

srf surface estimation

Terms in the formula that are not in the list of specials (i.e. the list of term types above) are automatically assigned an appropriate special wrapper, i.e. numerical covariates x are treated as lin(x) + sm(x), factors f (and numerical covariates with very few distinct values, see ssGAMDesign) are treated as fct(f). Valid model formulas have to satisfy the following conditions:

1. All variables that are involved in interactions have to be present as main effects as well, i.e. y ~ x1 + x1:x2 will produce an error because the main effect of x2 is missing.
2. Interactions between unpenalized terms not under selection (i.e. terms of type u) and penalized terms are not allowed, i.e. y ~ u(x1)* x2 will produce an error.
3. If main effects are specified as special terms, the interactions involving them have to be given as special terms as well, i.e. y ~ lin(x1) + lin(x2) + x1:x2 will produce an error.

The default prior settings for Gaussian data work best for a response with unit variance. If your data is scaled very differently, either rescale the response (recommended) or adjust the hyperparameters accordingly.

Sampling of the chains is done in parallel using package multicore if available. If not, a socket cluster set up with snow is used where available. Use options(mc.cores = foo) to set the (maximal) number of processes spawned by the parallelization. If options()$mc.cores is unspecified, snow uses 8.

Value

an object of class spikeSlabGAM with methods summary.spikeSlabGAM, predict.spikeSlabGAM, and plot.spikeSlabGAM.

Author(s)

Fabian Scheipl

References

Fabian Scheipl (2011). spikeSlabGAM: Bayesian Variable Selection, Model Choice and Regularization for Generalized Additive Mixed Models in R. *Journal of Statistical Software*, **43**(14), 1–24.

Fabian Scheipl, Ludwig Fahrmeir, Thomas Kneib (2012). Spike-and-Slab Priors for Function Selection in Structured Additive Regression Models. *Journal of the American Statistical Association*, **107**(500), 1518–1532.

See Also

ssGAMDesign for details on model specification, spikeAndSlab for more details on the MCMC sampler and prior specification, and ssGAM2Bugs for MCMC diagnostics. Check out the vignette for theoretical background and code examples.

Examples

```
## Not run:
## examples not run due to time constraints on CRAN checks.
## full examples below should take about 2-4 minutes.

set.seed(91179)
n <- 400
d <- data.frame(f1 = sample(gl(3, n/3)), f2 = sample(gl(4,
      n/4)), x1 = runif(n), x2 = runif(n), x3 = runif(n))
# true model:
#   - interaction between f1 and x1
#   - smooth interaction between x1 and x2,
#   - x3 and f2 are noise variables without influence on y
nf1 <- as.numeric(d$f1)
d$f <- with(d, 5 * (nf1 + 2 * sin(4 * pi * (x1 - 0.2) *
         (x2 - 0.7)) - nf1 * x1))
d$y <- with(d, scale(f + rnorm(n)))
d$yp <- with(d, rpois(n, exp(f/10)))

# fit & display the model:
m1 <- spikeSlabGAM(y ~ x1 * f1 + f1 * f2 + x3 * f1 +
    x1 * x2, data = d)
summary(m1)

# visualize estimates:
plot(m1)
plot(m1, cumulative = FALSE)
(plotTerm("fct(f1):fct(f2)", m1, aggregate = median))
```

```
print(p <- plotTerm("sm(x1):sm(x2)", m1, quantiles = c(0.25,
      0.75), cumulative = FALSE))

# change MCMC settings and priors:
mcmc <- list(nChains = 3, burnin = 100, chainLength = 1000,
  thin = 3, reduceRet = TRUE)
hyper <- list(gamma = c(v0 = 5e-04), tau2 = c(10,
    30), w = c(2, 1))

# complicated formula example, poisson response:
m2 <- spikeSlabGAM(yp ~ x1 * (x2 + f1) + (x2 + x3 + f2)^2 -
         sm(x2):sm(x3), data = d,
      family = "poisson", mcmc = mcmc,
        hyperparameters = hyper)
summary(m2)
plot(m2)

# quick&dirty convergence diagnostics:
print(b <- ssGAM2Bugs(m2))
plot(b)

## End(Not run)
```

`srf` *Generate design for penalized surface estimation.*

Description

This function generates the design for a 2-D penalized spline using (almost) radial basis functions. Use this type of term to account for *spatial* variation. Smooth interactions between covariates are often better fitted via the interactions of `lin` and `sm` terms, because they allow a decomposition into (linear and smooth) marginal trends and (linear-linear, linear-smooth/"varying coefficients", and smooth-smooth) interactions. This decomposition usually makes no sense for spatial data.

Usage

```
srf(
  coords,
  K = min(50, sum(nd)/4),
  rankZ = 0.999,
  centerBase = TRUE,
  baseType = c("B", "thinPlate"),
  decomposition = c("ortho", "MM", "asIs"),
  tol = 1e-10
)
```

Arguments

coords	a data.frame with two columns containing the coordinates
K	(approximate) number of basis functions in the original basis (defaults to 50). If baseType ="B" you can specify a vector giving the number of marginal basis functions in each direction.
rankZ	how many eigenvectors to retain from the eigen decomposition: either a number > 3 or the proportion of the sum of eigenvalues the retained eigenvectors must represent at least. Defaults to .999.
centerBase	project the basis of the penalized part into the complement of the column space of the basis of the unpenalized part? defaults to TRUE
baseType	Defaults to "B", i.e. a tensor product basis based on marginal cubic B-splines with ridge penalty (i.e. penalizing deviations from the constant). Set to "thinPlate" if cubic thin plate splines are desired, see note below.
decomposition	use a (truncated) spectral decomposition of the implied prior covariance of $f(x, y)$ for a low rank representation with orthogonal basis functions and i.i.d. coefficients ("ortho"), or use the mixed model reparameterization for non-orthogonal basis functions and i.i.d. coefficients ("MM") or use basis functions as they are with i.i.d. coefficients ("asIs"). Defaults to "ortho".
tol	count eigenvalues smaller than this as zero

Details

Note that srf() expects coords to be a data.frame within the larger data.frame supplied to spikeSlabGAM in its data argument, i.e. coords is considered a two-dimensional covariate.

If baseType is 'thinPlate', knot locations for the thin plate spline basis are chosen via a space-filling algorithm (i.e. medoids returned by clara) as suggested in Ruppert/Wand/Carroll, ch. 13.5. Since the thin plate penalty term penalizes deviations from a linear trend, it is recommended to add marginal linear trends and their interaction to the model if baseType ="thinPlate" to improve the fit.

Note that predictions outside the convex hull (sometimes even just close its border) of the original data tend to become rather unstable very quickly.

Value

a design matrix for the 2-D spline.

Author(s)

Fabian Scheipl

References

Ruppert, D., Wand, M.P., Carroll, R.J. (2003). Semiparametric Regression. Cambridge University Press

ssGAM2Bugs — *Convert samples from a model fitted with* spikeSlabGAM *into a* bugs-*object*

Description

Use plot and print on the returned bugs object for convergence diagnostics. Lack of convergence for α or ξ does not necessarily mean that the sampler has not converged for β.

Usage

```
ssGAM2Bugs(m, rm = c("alpha", "ksi", "gamma"))
```

Arguments

m	a model fitted with spikeSlabGAM
rm	which parameter blocks to omit from the conversion. Defaults to omission of α, ξ, and γ. Set to NULL to keep all MCMC samples.

Value

a bugs object which has convenience functions for assessing convergence based on parallel MCMC runs

Author(s)

Fabian Scheipl

Examples

```
#see help for spikeSlabGAM
```

ssGAMDesign — *Generate design and model information for* spikeSlabGAM

Description

This function generates the design matrix for a generalized additive (mixed) model, based on smoothing spline ANOVA-like orthogonal decomposition of the model terms and their interactions. It parses the formula given to spikeSlabGAM to provide all the arguments necessary for the MCMC sampler.

Usage

```
ssGAMDesign(
  formula,
  data,
  specials = c("u", "lin", "fct", "sm", "rnd", "mrf", "srf"),
  minUniqueValues = 6,
  lowRankInteractions = TRUE,
  orthogonalizeInteractions = TRUE,
  decomposition = NULL
)
```

Arguments

formula	the model formula. Follows the usual R syntax described in formula. Terms in the formula that are not in the list of specials are automatically assigned an appropriate special, i.e. numerical covariates x are treated as lin(x) + sm(x), factors f are treated as fct(f) (see specials argument). See spikeSlabGAM for more details.
data	data.frame containing all the variables in the function
specials	a vector of the types of possible model terms. These must be implemented as functions generating a design matrix with a label attribute. See also sm() for an example. The documentation for spikeSlabGAM contains a list of implemented model term types and usage examples.
minUniqueValues	the minimal number of unique values a covariate has to have in order to not be treated as a factor. Defaults to 6.
lowRankInteractions	should a low-rank approximation of the design matrix for interaction terms based on a (truncated) spectral decomposition of the implied covariance matrix be used? defaults to TRUE.
orthogonalizeInteractions	should the design matrices for interaction terms be projected into the complement of the column space of the respective main effects? Can help separate marginal and interaction effects. Defaults to TRUE.
decomposition	which decomposition to use, see sm. Defaults to the default of sm.

Details

Setting lowRankInteractions to FALSE can result in very large models, especially if higher-order interactions or interactions between terms with lots of parameters are involved. Note that numeric covariates with fewer unique values than minUniqueValues are treated as factors unless wrapped in a special argument.

This function is not meant to be called directly by the user, spikeSlabGAM is the user interface.

Value

a list with components:

`response`	the left hand side of the model formula
`Design`	the design matrix of the model specified in formula
`groupIndicators`	a factor that maps the columns of `Design` to the different model terms
`H`	a matrix containing the hierarchy of the penalization groups (not used, included for backwards compatibility)

Author(s)

Fabian Scheipl

`summary.spikeSlabGAM` *Summary for posterior of a* `spikeSlabGAM` *fit*

Description

Returns basic information about prior and model structure, posterior means of inclusion probabilities, term importance and the most probable models found by the SSVS.

Usage

```
## S3 method for class 'spikeSlabGAM'
summary(object, threshold = 0.5, ...)
```

Arguments

`object`	an object of class `spikeSlabGAM`
`threshold`	threshold for inclusion of a model term. Defaults to 0.5.
`...`	arguments passed from or to other methods (not used)

Details

Two scalar summaries of term importance are included:

`P(gamma = 1)` The posterior mean of $P(\gamma = 1)$, i.e. the marginal posterior inclusion probability of the term.

`pi` The scaled dot products of the posterior mean of the linear predictor η_i associated with the i-th term and the total linear predictor η: $\pi_i = \eta_i^T \eta / \eta^T \eta$. They sum to one (but can be negative as well), and (for mutually orthogonal η_i) provide a percentage decomposition of the sum of squares of η. See references for details.

The summary also shows the dimensionality of the basis associated with a term.

The top row in the model table shows the relative frequency of the respective model, the bottom row shows cumulative relative frequencies for the models visited most often.

Value

an object of class `summary.spikeSlabGAM`

Author(s)

Fabian Scheipl

References

Gu, Chong (2002). Smoothing Spline ANOVA models. Springer. (see chapter 3.6)

---

u *Generate design for an always included covariate*

Description

Basically a wrapper for `model.matrix(~ x, ...)`.

Usage

```
u(x, ...)
```

Arguments

x	covariate
...	arguments passed to `model.matrix`

Value

a design matrix for x

Author(s)

Fabian Scheipl

Index

bugs, *21*

clara, *20*
contrasts, *3*

evalTerm, 2

fct, 3, *16, 17*
formula, *22*

getPosteriorTerm, 4
ggplot, *7, 8*

lin, 4, *8*, *12*, *16, 17*, *19*

makeCluster, *13*
marrangeGrob, *6, 7*
model.matrix, *24*
mrf, 5, *9*, *16, 17*

options, *11*

plot.spikeSlabGAM, 6, *18*
plotTerm, *7*, 8
predict.spikeSlabGAM, 9, *18*
print.summary.spikeSlabGAM, 10

Reduce, *10*
rnd, 11, *16, 17*

sm, 12, *16, 17*, *19*, *22*
spikeAndSlab, 13, *17, 18*
spikeSlabGAM, *13*, 16, *20–22*
srf, *8*, *16, 17*, 19
ssGAM2Bugs, *18*, 21
ssGAMDesign, *17, 18*, 21
summary.spikeSlabGAM, *11*, *18*, 23

theme_gray, *7*

u, *12*, *17*, 24