Package 'glmBfp'

June 16, 2020

Imports survival, rms, statmod, methods, coda, Runuran (>= 0.12), Rcpp (>= 0.11.6)

License GPL (>= 2)

Title Bayesian Fractional Polynomials for GLMs

LinkingTo Rcpp, RcppArmadillo

LazyLoad yes

Description Implements the Bayesian paradigm for fractional polynomials in generalized linear models, described by Held, Gravestock, Sabanes Bove (2015) <doi:10.1214/14-STS510>. See package 'bfp' for the treatment of normal models.

Version 0.0-60

Date 2020-06-12

Depends R (>= 2.12.0)

Encoding UTF-8

Suggests MASS

Collate 'GPrior-classes.R' 'posteriors.R' 'GlmBayesMfp-methods.R' 'GlmBayesMfpSamples-class.R' 'GlmBayesMfpSamples-methods.R' 'McmcOptions-class.R' 'McmcOptions-methods.R' 'RcppExports.R' 'helpers.R' 'computeModels.R' 'constructNewdataMatrix.R' 'getFpTransforms.R' 'getDesignMatrix.R' 'coxTBF.R' 'evalZdensity.R' 'formula.R' 'fpScale.R' 'fpTrans.R' 'getFamily.R' 'getLogGPrior.R' 'getLogMargLikEstimate.R' 'getMarginalZ.R' 'sampleGlm.R' 'sampleBma.R' 'getModelCoefs.R' 'glmBayesMfp.R' 'glmBfp-package.R' 'hpds.R' 'inclusionProbs.R' 'optimize.R' 'plotCurveEstimate.R' 'predictCoxTBF.R' 'testCox.R' 'uncenteredDesignMatrix.R' 'writeFormula.R'

RoxygenNote 7.1.0

Author Isaac Gravestock [aut, cre],
Daniel Sabanes Bove [aut],
David Maisonave [ctb],
Terry Therneau [ctb],
R Core Team [ctb]

Maintainer Isaac Gravestock <isaac.gravestock@gmail.com>

Repository CRAN

NeedsCompilation yes

Date/Publication 2020-06-16 14:10:15 UTC

R topics documented:

glmBfp-package *Bayesian inference for fractional polynomial models from the GLM and Cox family*

Description

Bayesian inference for fractional polynomial models from the GLM and Cox family

Author(s)

Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

as.data.frame.GlmBayesMfp *Convert a GlmBayesMfp object into a data frame*

Description

Convert a GlmBayesMfp object into a data frame

Usage

```
## S3 method for class 'GlmBayesMfp'
as.data.frame(x, row.names = NULL, ..., freq = FALSE)
```

Arguments

x	valid GlmBayesMfp object
row.names	optional rownames (default is to keep the names of the GlmBayesMfp list)
...	unused
freq	should empirical frequencies of the models in the sampling path be given? (not default)

Value

The data frame with the following columns:

posterior the posterior model probabilities

logMargLik the log marginal likelihood of the models

logPrior the log prior probabilities of the models

Additionally, for each uncertain fixed form covariates a column with the inclusion status, and for each fractional polynomial a column with the powers are returned.

Author(s)

Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

See Also

glmBayesMfp

---

bfp *Mark a covariate for transformation with fractional polynomials*

Description

Using this function, you can mark a covariate for transformation with fractional polynomials in the formula for glmBayesMfp.

Using this function, you can mark a covariate or a group of combined covariates for joint variable selection ("uncertain covariate fixed form covariate groups") in the formula for glmBayesMfp.

Usage

```
bfp(x, max = 2, scale = TRUE, rangeVals = NULL)

uc(x)
```

Arguments

x	the covariate name
max	maximum degree for this FP
scale	use pre-transformation shifting and scaling to avoid numerical problems?
rangeVals	extra numbers if the shifting and scaling should consider values in this range

Value

name of the provided covariate, with the other function parameters as attached attributes

Just the name of the provided covariate

`computeModels` *Compute model information for a given list of model configurations and glmBayesMfp output.*

Description

If we want to compute the marginal likelihood and information necessary for generating posterior samples for new models not encountered in the model search done by `glmBayesMfp`, this function can be used: Provide it with the models `configurations` to be interpreted in the context of the `object` of class `GlmBayesMfp`. The result is again of the latter class, but contains only the new models (similarly as the whole model space would consist of these and an exhaustive search would have been conducted).

Usage

```
computeModels(
  configurations,
  object,
  verbose = length(configurations) > 100L,
  debug = FALSE
)
```

Arguments

`configurations`	list of the model configurations
`object`	the `GlmBayesMfp` object
`verbose`	be verbose? (default: only for more than 100 configurations)
`debug`	be even more verbose and echo debug-level information? (not by default)

Value

The `GlmBayesMfp` object with the new models. This can directly be used as input for `sampleGlm`.

Author(s)

Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

convert2Mcmc *Convert samples to mcmc objects*

Description

Convert samples to mcmc objects

Usage

```
convert2Mcmc(samples, mcmcOptions)
```

Arguments

samples	samples matrix or vector
mcmcOptions	the McmcOptions object which was chosen for the production of samples

Value

an S3 class "mcmc" object

coxTBF *Fit Cox models using glmBayesMfp*

Description

A simplified formula based interface to glmBayesMfp to fit Cox models. Can return Maximum a posteriori (MAP) model, Median probability model (MPM) or Bayesian model average (BMA). Provides global empirical Bayes and AIC/BIC based model inference.

Usage

```
coxTBF(
  formula,
  data,
  type,
  baseline = "shrunk",
  globalEB = FALSE,
  IC = FALSE,
  sep = FALSE,
  keepModelList = FALSE,
  ...,
  overrideConfig
)
```

Arguments

formula	model formula with Surv object as LHS and uc or bfp variables as RHS.
data	data.frame for model variables
type	type of model to fit, one of "MAP","MPM","BMA","BMAFull"
baseline	how to calculate the baseline hazard function. "cox" uses unshrunken coefficients. "shrunk" refits baseline with shrunken coefficients (default).
globalEB	use global empirical bayes estimate of g (default=FALSE)
IC	use information criteria based model selection (default=FALSE). Either "AIC" or "BIC".
sep	estimate baseline hazard for each estimate of model coefficients (default=FALSE).
keepModelList	keep the model list returned by glmBayesMfp for MAP and MPM models (default=FALSE).
...	additional arguments to pass to glmBayesMfp
overrideConfig	replaces the the MAP model with the given configuration, which is passed to computeModels

Value

An object of S3 class TBFcox or TBFcox.sep if sep=TRUE.

CustomGPrior *Constructor for the custom g-prior class*

Description

Constructor for the custom g-prior class

Usage

```
CustomGPrior(logDens)
```

Arguments

logDens	the log prior density function for g

Value

a new CustomGPrior object

CustomGPrior-class *The custom g-prior class*

Description

This class wraps around a custom log prior density for the covariance factor g.

See Also

the constructor CustomGPrior

empiricalHpd *Construct an empirical HPD interval from samples*

Description

Construct an empirical highest posterior density (HPD) interval from samples which have been drawn from the distribution of a quantity of interest.

Usage

```
empiricalHpd(theta, level)
```

Arguments

theta	the vector of samples
level	the credible level

Value

A vector with the estimated lower and upper bounds of the HPD interval.

See Also

scrHpd

Extract.GlmBayesMfp *Extract method for GlmBayesMfp objects*

Description

Extract a subset of models from a GlmBayesMfp object.

Usage

```
## S3 method for class 'GlmBayesMfp'
x[...]
```

Arguments

x	valid GlmBayesMfp object
...	transports the indexes of the models

Value

The subsetted object.

Author(s)

Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

See Also

glmBayesMfp

fpTrans *Transform formula variables*

Description

Simple function to apply the Box Tidwell transformation to a variables in a formula. Variable is first shifted and scaled NewVar = (Var+shift)/scale then transformed and optionally centered. Can be used in formulas as poly() is.

Usage

```
fpTrans(var, powers = 1, scale = 1, shift = 0, center = TRUE)
```

Arguments

var	the variable to transform
powers	one or more powers
scale	value to scale the variable after shifting (default=1)
shift	value to shift the variable by (default=0)
center	center the variable when tranforming.

Value

the transformed vector

---

glmBayesMfp *Bayesian model inference for fractional polynomial GLMs and Cox models*

Description

Bayesian model inference for fractional polynomial models from the generalized linear model family or the Cox model is conducted by means of either exhaustive model space evaluation or posterior model sampling. The approach is based on analytical marginal likelihood approximations, using integrated Laplace approximation. Alternatively, test-based Bayes factors (TBFs) are used.

Usage

```
glmBayesMfp(
  formula = formula(data),
  censInd = NULL,
  data = parent.frame(),
  weights,
  offset,
  family,
  phi = 1,
  tbf = FALSE,
  empiricalBayes = FALSE,
  fixedg = NULL,
  priorSpecs = list(gPrior = HypergPrior(), modelPrior = "sparse"),
  method = c("ask", "exhaustive", "sampling"),
  subset,
  na.action = na.omit,
  verbose = TRUE,
  debug = FALSE,
  nModels,
  nCache = 1e+09,
  chainlength = 10000,
  nGaussHermite = 20,
```

```
  useBfgs = FALSE,
  largeVariance = 100,
  useOpenMP = TRUE,
  higherOrderCorrection = FALSE,
  fixedcfactor = FALSE,
  empiricalgPrior = FALSE,
  centerX = TRUE
)
```

Arguments

formula	model formula
censInd	censoring indicator. Default is NULL, but if a non-NULL vector is supplied, this is assumed to be logical (TRUE = observed, FALSE = censored) and Cox regression is performed.
data	optional data.frame for model variables (defaults to the parent frame)
weights	optionally a vector of positive weights (if not provided, a vector of one's)
offset	this can be used to specify an _a priori_ known component to be included in the linear predictor during fitting. This must be a numeric vector of length equal to the number of cases (if not provided, a vector of zeroes)
family	distribution and link (as in the glm function). Needs to be explicitly specified for all models except the Cox model.
phi	value of the dispersion parameter (defaults to 1)
tbf	Use TBF methodology to compute the marginal likelihood? (not default) Must be TRUE if Cox regression is done.
empiricalBayes	rank the models in terms of *conditional* marginal likelihood, using an empirical Bayes estimate of g? (not default) Due to coding structure, the prior on g must be given in priorSpecs although it does not have an effect when empiricalBayes==TRUE.
fixedg	If this is a number, then it is taken as a fixed value of g, and as with the empiricalBayes option, the models are ranked in terms of conditional marginal likelihood. By default, this option is NULL, which means that g is estimated in a fully or empirical Bayesian way.
priorSpecs	prior specifications, see details
method	which method should be used to explore the posterior model space? (default: ask the user)
subset	optional subset expression
na.action	default is to skip rows with missing data, and no other option supported at the moment
verbose	should information on computation progress be given? (default)
debug	print debugging information? (not default)
nModels	how many best models should be saved? (default: 1% of the total number of (cached) models). Must not be larger than nCache if method == "sampling".
nCache	maximum number of best models to be cached at the same time during the model sampling, only has effect if method = sampling

chainlength	length of the model sampling chain (only has an effect if sampling has been chosen as method)
nGaussHermite	number of quantiles used in Gauss Hermite quadrature for marginal likelihood approximation (and later in the MCMC sampler for the approximation of the marginal covariance factor density). If empiricalBayes or a fixed g is used, this option has no effect.
useBfgs	Shall the BFGS algorithm be used in the internal maximization (not default)? Else, the default Brent optimize routine is used, which seems to be more robust. If empiricalBayes or a fixed g is used, this option has no effect and always the Brent optimize routine is used.
largeVariance	When should the BFGS variance estimate be considered "large", so that a reestimation of it is computed? (Only has an effect if useBfgs == TRUE, default: 100)
useOpenMP	shall OpenMP be used to accelerate the computations? (default)
higherOrderCorrection	should a higher-order correction of the Laplace approximation be used, which works only for canonical GLMs? (not default)
fixedcfactor	If TRUE sets the c factor assuming alpha is set to 0. Otherwise take alpha=mean(y)
empiricalgPrior	If TRUE uses the the observed isnformation matrix instead of X'X in the g prior. (Experimental)
centerX	Center the data before fitting (FALSE)

Details

The formula is of the form `y ~ bfp(x1,max = 4) + uc(x2 + x3)`, that is, the auxiliary functions `bfp` and `uc` must be used for defining the fractional polynomial and uncertain fixed form covariates terms, respectively. There must be an intercept, and no other fixed covariates are allowed. All `max` arguments of the `bfp` terms must be identical. `y` is the response vector for GLMs or the vector of survival times for Cox regression. Note that Cox regression is only implemented with TBFs.

The prior specifications are a list:

gPrior A g-prior class object. Defaults to a hyper-g prior. See `GPrior` for more information.

modelPrior choose if a flat model prior (`"flat"`), a model prior favoring sparse models explicitly (default, `"sparse"`), or a dependent model prior (`"dependent"`) should be used.

If `method = "ask"`, the user is prompted with the maximum cardinality of the model space and can then decide whether to use posterior sampling or the exhaustive model space evaluation.

Note that if you specify only one FP term, the exhaustive model search must be done, due to the structure of the model sampling algorithm. However, in reality this will not be a problem as the model space will typically be very small.

Value

An object of S3 class `GlmBayesMfp`.

GlmBayesMfpSamples-subsetting *Subset method for GlmBayesMfpSamples objects*

Description

Index the samples to select a subset of samples.

Usage

```
## S4 method for signature 'GlmBayesMfpSamples,ANY,missing,missing'
x[i]
```

Arguments

x	the GlmBayesMfpSamples object
i	the vector defining the subset of samples

Value

The subset of the same class.

Note

The function call will fail if any of the saved bfpCurves or ucCoefs does not have enough samples to be subset by i !

See Also

GlmBayesMfpSamples

HypergPrior *Constructor for the hyper-g prior class*

Description

Constructor for the hyper-g prior class

Usage

```
HypergPrior(a = 4)
```

Arguments

a	the hyperparameter which must be larger than 3, and should not be larger than 4 in order not to favour too much shrinkage a priori (default: 4)

Value

a new HypergPrior object

HypergPrior-class *The hyper-g prior class*

Description

The slots are:

a the hyperparameter

See Also

the constructor HypergPrior

IncInvGammaGPrior *Constructor for the incomplete inverse gamma g-prior class*

Description

Constructor for the incomplete inverse gamma g-prior class

Usage

```
IncInvGammaGPrior(a = 0.001, b = 0.001)
```

Arguments

a	the first positive hyperparameter (default: 0.001)
b	the second positive hyperparameter (default: 0.001)

Value

a new IncInvGammaGPrior object

IncInvGammaGPrior-class

The incomplete inverse gamma g-prior class

Description

The slots are:

a the first hyperparameter

b the second hyperparameter

See Also

the constructor `IncInvGammaGPrior`

inclusionProbs

Compute posterior inclusion probabilites based on GlmBayesMfp object

Description

Compute (model averaged) posterior inclusion probabilites for the uncertain variables (including FP variables) based on a GlmBayesMfp object.

Usage

```
inclusionProbs(
  GlmBayesMfpObject,
  postProbs = posteriors(GlmBayesMfpObject, type = "normalized")
)
```

Arguments

`GlmBayesMfpObject`	the GlmBayesMfp object
`postProbs`	the vector of posterior model probabilities, defaults to the normalized probabilities in `GlmBayesMfpObject`

Value

the resulting inclusion probabilities vector

InvGammaGPrior *Constructor for the inverse gamma g-prior class*

Description

Constructor for the inverse gamma g-prior class

Usage

```
InvGammaGPrior(a = 0.001, b = 0.001)
```

Arguments

a	the first positive hyperparameter (default: 0.001)
b	the second positive hyperparameter (default: 0.001)

Value

a new InvGammaGPrior object

InvGammaGPrior-class *The inverse gamma g-prior class*

Description

The slots are:

a the first hyperparameter

b the second hyperparameter

See Also

the constructor InvGammaGPrior

logMargLiks — *Extract the log marginal likelihood estimates from a GlmBayesMfp object*

Description

Extract the log marginal likelihood estimates from a GlmBayesMfp object

Usage

```
logMargLiks(GlmBayesMfpObject)
```

Arguments

GlmBayesMfpObject	the object

Value

the vector of log marginal likelihood estimates

logPriors — *Extract the log prior values from a GlmBayesMfp object*

Description

Extract the log prior values from a GlmBayesMfp object

Usage

```
logPriors(GlmBayesMfpObject)
```

Arguments

GlmBayesMfpObject	the object

Value

the vector of log prior values

McmcOptions *Constructor for class McmcOptions*

Description

Note that the argument `samples` is included for convenience only - you can specify it instead of `iterations`.

Usage

```
McmcOptions(
  iterations = as.integer(burnin + (step * samples)),
  burnin = 10000L,
  step = 10L,
  samples = 10000L
)
```

Arguments

`iterations`	number of MCMC iterations (default: `110,000`)
`burnin`	number of burn-in iterations which are not saved (default: `10,000`)
`step`	only every step-th iteration is saved after the burn-in (default: `10`)
`samples`	number of resulting samples (by default `10,000` will result)

Value

the freshly built object of class `McmcOptions`

plotCurveEstimate *Function for plotting a fractional polynomial curve estimate*

Description

Plot a fractional polynomial curve estimate using samples from a single GLM / Cox model or a model average.

Usage

```
plotCurveEstimate(
  samples,
  termName,
  plevel = 0.95,
  slevel = plevel,
  plot = TRUE,
  rug = FALSE,
```

```
  addZeros = FALSE,
  ...
)
```

Arguments

samples	an object of class GlmBayesMfpSamples, produced by sampleGlm and sampleBma.
termName	string denoting an FP term, as written by the as.data.frame method
plevel	credible level for pointwise HPD, and NULL means no pointwise HPD (default: 0.95). The pointwise intervals are plotted in blue color.
slevel	credible level for simultaneous credible band (SCB), NULL means no SCB (defaults to plevel). The simultaneous intervals are plotted in green color.
plot	if FALSE, only return values needed to produce the plot, but do not plot (default is TRUE, so a plot is made)
rug	add a rug to the plot? (default: FALSE)
addZeros	include zero samples for models where the covariate is not included? (default: FALSE) If TRUE, this changes the interpretation of the samples, and therefore curve estimates based on these samples: it is no longer conditional on inclusion of the covariate, but marginally over all models, also those not including the covariate.
...	further arguments for plotting with matplot

Value

a list of various plotting information:

original	grid on the original covariate scale
grid	grid on the transformed scale
mean	pointwise mean curve values
plower	lower boundaries for pointwise HPD
pupper	upper boundaries for pointwise HPD
slower	lower boundaries for SCB
supper	upper boundaries for SCB
obsVals	observed values of the covariate on the original scale
partialResids	not implemented: partial residuals
transform	vector of shift and scale parameter

posteriors *Extract posterior model probability estimates from a GlmBayesMfp object*

Description

Extract posterior model probability estimates from a GlmBayesMfp object

Usage

```
posteriors(GlmBayesMfpObject, type = c("normalized", "sampling"))
```

Arguments

GlmBayesMfpObject	the object
type	type of posterior model probability estimates to be extracted from GlmBayesMfpObject

Value

the requested probs from all models

predict.TBFcox *Prediction methods for CoxTBF objects*

Description

Predicts survival probabilities at given times. Compatible with predictSurvProb functions required by pec package.

Usage

```
## S3 method for class 'TBFcox'
predict(object, newdata, times, ...)
```

Arguments

object	a model fitted with coxTBF
newdata	a dataframe with the same variables as the original data used to fit the object
times	a vector of times to predict survival probability for
...	not used.

Value

A data frame of survival probabilities with rows for each row of newdata and columns for each time.

predict.TBFcox.BMA *Prediction methods for CoxTBF objects for BMA models*

Description

Predicts survival probabilities at given times. Compatible with predictSurvProb functions required by pec package. Predicts BMA objects.

Usage

```
## S3 method for class 'TBFcox.BMA'
predict(object, newdata, times, ...)
```

Arguments

object	a model fitted with coxTBF
newdata	a dataframe with the same variables as the original data used to fit the object
times	a vector of times to predict survival probability for
...	not used.

Value

A data frame of survival probabilities with rows for each row of newdata and columns for each time.

predict.TBFcox.sep *Prediction methods for CoxTBF objects with separate estimates*

Description

Predicts survival probabilities at given times. Compatible with predictSurvProb functions required by pec package. Predicts objects with fitted with sep=TRUE

Usage

```
## S3 method for class 'TBFcox.sep'
predict(object, newdata, times, ...)
```

Arguments

object	a model fitted with coxTBF
newdata	a dataframe with the same variables as the original data used to fit the object
times	a vector of times to predict survival probability for
...	not used.

Value

A data frame of survival probabilities with rows for each row of newdata and columns for each time.

print.GlmBayesMfp *Print a GlmBayesMfp object.*

Description

Print a GlmBayesMfp object.

Usage

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

Arguments

x	valid GlmBayesMfp object
...	unused

Value

Only used for its side effect

Author(s)

Daniel Sabanes Bove <daniel.sabanesbove@ifspm.uzh.ch>

See Also

glmBayesMfp

sampleBma *Produce posterior samples from a Bayesian model average over GLMs / Cox models*

Description

Based on the result list from glmBayesMfp, sample from the Bayesian model average (BMA) over the models contained in this list.

Usage

```
sampleBma(
  object,
  mcmc = McmcOptions(),
  postProbs = posteriors(object),
  nMargLikSamples = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

object	valid GlmBayesMfp object containing the models over which to average
mcmc	MCMC options object with class McmcOptions, specifying the number of required BMA samples (via sampleSize(mcmc)), and the burn-in and thinning parameters applied to each model (see above). If TBF is used, each sample is accepted, and the number of samples is given by sampleSize(mcmc).
postProbs	vector of posterior probabilites (will be normalized within the function) for the weighting of the models in object (defaults to the normalized posterior probabilities)
nMargLikSamples	If this is non-NULL, it specified the number of samples used for the marginal likelihood estimate for each model (see above).
verbose	should information on computation progress be given? (default)
...	optional further arguments already available for sampling from a single model: gridList, gridSize, newdata, weights, marginalZApprox, debug, useOpenMP. See sampleGlm for the meanings.

Details

If TBF methodology is used (which is specified within the glmBayesMfp object), then Monte Carlo (MC) sampling is used. If the fully Bayesian, generalized hyper-g prior methodology is used, then the sampling is based on MCMC. Therefore, instead of only specifying the required number of samples and the model probabilities, one also needs to specify the burn-in length and the thinning parameter, which will be applied to every model from which at least one sample is included in the average. Alternatively, you can ask for MCMC marginal likelihood estimates for all models in the list. Then at least nMargLikSamples will be produced for each model, whether included in the BMA sample or not.

Value

The result is a list with the following elements:

modelData data frame containing the result from the as.data.frame function, and in addition BMA probabilities, BMA frequencies in the sample, acceptance ratios of the MCMC runs and optionally marginal likelihood estimates / standard errors.

samples an object of S4 class GlmBayesMfpSamples containing the samples from the BMA.

---

sampleGlm *Produce posterior samples from one GLM / Cox model*

Description

Based on the result list from glmBayesMfp, for the first model in the list MCMC samples are produced. In parallel to the sampling of coefficients and FP curve points, optionally the marginal likelihood of the model is estimated with MCMC samples. This provides a check of the integrated Laplace approximation used in the model sampling. If TBF methodology is used, then no MCMC is necessary, instead ordinary Monte Carlo samples from an approximate posterior distribution are obtained.

Usage

```
sampleGlm(
  object,
  mcmc = McmcOptions(),
  estimateMargLik = TRUE,
  gridList = list(),
  gridSize = 203L,
  newdata = NULL,
  fixedZ = NULL,
  marginalZApprox = NULL,
  verbose = TRUE,
  debug = FALSE,
  useOpenMP = TRUE,
  correctedCenter = FALSE
)
```

Arguments

object	the GlmBayesMfp object, from which only the first model will be processed (at least for now . . .)
mcmc	MCMC options object with class McmcOptions. If TBF is used, each sample is accepted, and the number of samples is given by sampleSize(mcmc).
estimateMargLik	shall the marginal likelihood be estimated in parallel? (default) Only has an effect if full Bayes and not TBF is used.
gridList	optional list of appropriately named grid vectors for FP evaluation. Default is length (gridSize - 2) grid per covariate additional to the observed values (two are at the endpoints)
gridSize	see above (default: 203)
newdata	new covariate data.frame with exactly the names (and preferably ranges) as before (default: no new covariate data) Note that there is no option for offsets for new data at the moment. Just add the offsets to the predictions slot of samples in the return list yourself.
fixedZ	either NULL (default) or a (single) fixed z value to be used, in order to sample from the conditional posterior given this z. If object was constructed by the empirical Bayes machinery, this will default to the estimated z with maximum conditional marginal likelihood. If object was constructed with the option fixedg, then the fixed value will be used by default.
marginalZApprox	method for approximating the marginal density of the log covariance factor z, see getMarginalZ for the details (default: same preference list as in getMarginalZ) If TBF are used in conjunction with incomplete inverse gamma hyperprior on g = exp(z), then the posterior distribution of g is again of this form. Therefore this option does not have any effect in that case, because the samples are directly obtained from that posterior distribution.
verbose	should information on computation progress be given? (default)

debug	print debugging information? (not default)
useOpenMP	shall OpenMP be used to accelerate the computations? (default)
correctedCenter	If TRUE predict new data based on the centering of the original data.

Value

Returns a list with the following elements:

samples an object of S4 class GlmBayesMfpSamples

coefficients samples of all original coefficients in the model (nCoefs x nSamples)

acceptanceRatio proportion of accepted Metropolis-Hastings proposals

logMargLik if estimateMargLik is TRUE, this list is included: it contains the elements numeratorTerms and denominatorTerms for the numerator and denominator samples of the Chib Jeliazkov marginal likelihood estimate, highDensityPointLogUnPosterior is the log unnormalized posterior density at the fixed parameter and the resulting estimate and standardError.

---

sampleSize *Compute the number of samples for a given MCMC options triple*

Description

Compute the number of samples for a given MCMC options triple

Usage

```
sampleSize(mcmcOptions)
```

Arguments

mcmcOptions	the McmcOptions object

Value

the resulting sample size

scrHpd *Calculate an SCB from a samples matrix*

Description

Calculate an SCB from a samples matrix, which minimizes the absolute distances of the contained samples to a mode vector, at each gridpoint. Therefore the SCB might be considered an "HPD SCB".

Usage

```
scrHpd(samples, mode = apply(samples, 1, median), level = 0.95)
```

Arguments

samples	m by n matrix where m is the number of parameters, n is the number of samples and hence each (multivariate) sample is a column in the matrix samples
mode	mode vector of length m (defaults to the vector of medians)
level	credible level for the SCB (default: 0.95)

Value

A matrix with columns "lower" and "upper", with the lower and upper SCB bounds, respectively.

References

Besag, J.; Green, P.; Higdon, D. \& Mengersen, K. (1995): "Bayesian computation and stochastic systems (with discussion)", *Statistical Science*, 10, 3-66.

See Also

empiricalHpd

Index