Package ‘bfp’

May 14, 2018

Version 0.0-39
Date 2018-05-14
Title Bayesian Fractional Polynomials
Depends R (>= 3.0.0)
Imports Rcpp (>= 0.11.0)
LinkingTo Rcpp
Suggests doBy, SoDA, Hmisc, mfp
SystemRequirements GNU make, C++11
Description Implements the Bayesian paradigm for fractional polynomial models under the assumption of normally distributed error terms, see Sabanes Bove, D. and Held, L. (2011) <doi:10.1007/s11222-010-9170-7>.
License GPL (>= 2)
Author Daniel Sabanes Bove [aut, cre], Isaac Gravestock [aut], Robert Davies [cph], Stephen Moshier [cph]
Maintainer Daniel Sabanes Bove <daniel.sabanesbove@gmx.net>
Repository CRAN
Repository/R-Forge/Project bfp
Repository/R-Forge/Revision 150
Repository/R-Forge/DateTimeStamp 2018-05-14 10:35:40
Date/Publication 2018-05-14 14:37:07 UTC
NeedsCompilation yes
as.data.frame.BayesMfp

Description

Convert the `BayesMfp` object to a data frame with the saved models.

Usage

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

Arguments

- `x` valid `BayesMfp` object
- `row.names` optional rownames (default is to keep the names of the `BayesMfp` list)
- `freq` should empirical frequencies of the models in the sampling path be given? (default)
- `...` unused
BayesMfp

Author(s)
Daniel Sabanès Bové

See Also
summary.BayesMfp

Examples

## generate a BayesMfp object
set.seed(19)

x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)

y <- rt(n=15, df=2)

test <- BayesMfp(y ~ bfp(x2, max = 4) + uc(x1 + x3), nModels = 100,
                method="exhaustive")

## get the models data frame
as.data.frame(test)

BayesMfp
Bayesian model inference for multiple fractional polynomial models

Description
Bayesian model inference for multiple fractional polynomial models is conducted by means of
either exhaustive model space evaluation or posterior model sampling.

Usage
BayesMfp(formula = formula(data), data = parent.frame(), family =
gaussian, priorSpecs = list(a = 4, modelPrior = "flat"), method =
c("ask", "exhaustive", "sampling"), subset = NULL, na.action = na.omit,
verbose = TRUE, nModels = NULL, nCache=1e9L, chainlength = 1e5L)

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

uc(x)

Arguments

formula  model formula

data  optional data.frame for model variables (defaults to the parent frame)

family  distribution and link: only gaussian("identity") supported at the moment
BayesMfp

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)
nModels     how many best models should be saved? (default: 1% of the explored models or the chainlength, 1 would mean only the maximum a posteriori [MAP] model)
nCache      maximum number of best models to be cached at the same time during the model sampling (only has an effect if sampling has been chosen as method)
chainlength length of the model sampling chain (only has an effect if sampling has been chosen as method)
x           variable
max         maximum degree for this FP (default: 2)
scale       use pre-transformation scaling to avoid numerical problems? (default)
rangeVals   extra numbers if the scaling should consider values in this range. Use this argument if you have test data with larger range than the training range.

Details

The formula is of the form \( y \sim bfp(x_1, \text{max} = 4) + uc(x_2 + x_3) \), 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.

The prior specifications are a list:

- **a** hyperparameter for hyper-g prior which must be greater than 3 and is recommended to be not greater than 4 (default is 4)
- **modelPrior** choose if a flat model prior (default, "flat"), a model prior favoring sparse models explicitly ("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

Returns an object of class BayesMfp that inherits from list. It is essentially a list of models. Each model is a list and has the following components:

- **powers** a list of numeric vectors, where each vector contains the powers of the covariate that its name denotes.
BayesMfp

ucTerms an integer vector of the indices of uncertain fixed form covariates that are present in the model.

logM log marginal likelihood

logP log prior probability

posterior normalized posterior probability, and if model sampling was done, the frequency of the model in the sampling algorithm

postExpectedg posterior expected covariance factor g

postExpectedShrinkage posterior expected shrinkage factor t=g/(g + 1)

R2 usual coefficient of determination for the linear model

Subsetting the object with [.BayesMfp returns again a BayesMfp object with the same attributes, which are

numVisited the number of models that have been visited (exhaustive search) or cached (model sampling)

inclusionProbs BMA inclusion probabilities for all uncertain covariates

linearInclusionProbs BMA probabilities for exactly linear inclusion of FP covariates

logNormConst the (estimated) log normalizing constant f(D)

chainlength length of the Markov chain, only present if method = "sampling"

call the original call

formula the formula by which the appropriate untransformed design matrix can be extracted

x the shifted and scaled design matrix for the data

xCentered the column-wise centered x

y the response vector

yMean the mean of the response values

SST sum of squares total

indices a list with components that describe the positions of uncertain covariate groups, fractional polynomial terms and fixed variables in the design matrix

termNames a list of character vectors containing the names of uncertain covariate groups, fractional polynomial terms and fixed variables

shiftScaleMax matrix with 4 columns containing preliminary transformation parameters, maximum degrees and cardinalities of the powersets of the fractional polynomial terms

priorSpecs the utilized prior specifications

randomSeed if a seed existed at function call (get(".Random.seed", .GlobalEnv)), it is saved here

Note

logNormConst may be unusable due to necessary conversion from long double to double!

Various methods for posterior summaries are available.
See Also

BayesMfp Methods, BmaSamples

Examples

```r
## generate some data
set.seed(19)

x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)
y <- rt(n=15, df=2)

## run an exhaustive model space evaluation with a flat model prior and
## a uniform prior (a = 4) on the shrinkage factor t = g/(1 + g):
test <- BayesMfp(y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 100,
               method="exhaustive")
test

## now the same with a dependent model prior:
test2 <- BayesMfp(y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 100,
                 priorSpecs = list(a = 4, modelPrior = "dependent"),
                 method="exhaustive")
test2
```

Description

Print the object (print), get fitted values (fitted) and corresponding residuals (residuals).

Usage

```r
## S3 method for class 'BayesMfp'
print(x, ...)
## S3 method for class 'BayesMfp'
fitted(object, design = getDesignMatrix(object), post =
       getPosteriorParms(object, design = design), ...)
## S3 method for class 'BayesMfp'
residuals(object, ...)
```

Arguments

- `x`: valid `BayesMfp` object
- `object`: valid `BayesMfp` object, only the first model will be used.
bmaPredict

Design

- `design`: design matrix of the first model in the object, which can be supplied by the caller if it is computed beforehand.
- `post`: posterior parameters of the normal-gamma distribution (defaults to the posterior expected mean, marginalized over the covariance factor g).
- `...`: unused.

Author(s)

- Daniel Sabanés Bové

See Also

- `BayesMfp, BmaSamples` Methods

Examples

```r
## generate a BayesMfp object
set.seed(19)

x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)

y <- rt(n=15, df=2)

test <- BayesMfp(y ~ bfp(x2, max = 4) + uc (x1 + x3), nModels = 100, method="exhaustive")

## the print method

test

## extract fitted values and corresponding residuals
fitted(test)
residuals(test)
```

bmaPredict

- **BMA prediction for new data points**

Description

Make a Bayesian model averaged prediction for new data points, from those models saved in a `BayesMfp` object.

Usage

```r
bmaPredict(BayesMfpObject, postProbs = posteriors(BayesMfpObject), newdata)
```
Arguments

BayesMfpObject  BayesMfp object with the models over which the predictions should be averaged
postProbs  vector of posterior probabilities, which are then normalized to the weights of the model average (defaults to the normalized posterior probability estimates)
newdata  new covariate data as data.frame

Value

The predicted values as a vector.

Note

Note that this function is not an S3 predict method for BmaSamples objects, but a function working on BayesMfp objects (because we do not need BMA samples to do BMA point predictions).

Author(s)

Daniel Sabanés Bové

See Also

BmaSamples Methods

Examples

```r
## generate a BayesMfp object
set.seed(19)

x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)

y <- rt(n=15, df=2)

test <- BayesMfp(y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 100,
               method="exhaustive")

## predict new responses at (again random) covariates
bmaPredict(test,
            newdata = list(x1 = rnorm(n=15),
                            x2 = rbinom(n=15, size=5, prob=0.2) + 1,
                            x3 = rexp(n=15)))
```
Description

Draw samples from the Bayesian model average over the models in saved in a BayesMfp-object.

Usage

BmaSamples(object, sampleSize = length(object) * 10, postProbs =
posteriors(object), gridList = list(), gridSize = 203, newdata=NULL,
verbose = TRUE, includeZeroSamples=FALSE)

Arguments

object valid BayesMfp object containing the models over which to average
sampleSize sample size (default is 10 times the number of models)
postProbs vector of posterior probabilities (will be normalized within the function, defaults
to the normalized posterior probabilities)
gridList optional list of appropriately named grid vectors for FP evaluation, default is a
length (gridSize - 2) grid per covariate additional to the observed values (two
are at the minimum and maximum)
gridSize see above (default: 203)
newdata new covariate data.frame with exactly the names (and preferably ranges) as be-
fore (default: no new covariate data)
verbose should information on sampling progress be printed? (default)
includeZeroSamples should the function and coefficient samples include zero samples, from models
where these covariates are not included at all? (default: FALSE, so the zero
samples are not included)

Value

Return an object of class BmaSamples, which is a list with various elements that describe the
BayesMfp object over which was averaged, model frequencies in the samples, the samples them-
elves etc:
priorSpecs the utilized prior specifications
termNames a list of character vectors containing the names of uncertain covariate groups,
 fractional polynomial terms and fixed variables
shiftScaleMax matrix with 4 columns containing preliminary transformation parameters, max-
imum degrees and cardinalities of the powersets of the fractional polynomial terms
y the response vector
the shifted and scaled design matrix for the data

if a seed existed at function call (get(".Random.seed", .GlobalEnv)), it is saved here

The table of model frequencies in the BMA sample

data frame containing the normalized posterior probabilities of the models in the underlying `BayesMfp` object, corresponding log marginal likelihoods, model prior probabilities, posterior expected covariance and shrinkage factors, coefficients of determination, powers and inclusions, and finally model average weights and relative frequencies in the BMA sample.

sample size

BMA samples of the regression variance

BMA samples of the shrinkage factor

samples of the intercept

named list of the FP function samples, where each element contains one FP covariate and is a matrix (samples x grid), with the following attributes:

whereObsVals where in the scaled grid are the originally observed covariate values? (integer vector of the indexes)

scaledGrid numeric vector with the positions of the scaled grid points, corresponding to the columns of the samples matrix

counter how often has this covariate been included in the BMA sample? (identical to the number of rows in the samples matrix)

named list of the uncertain fixed form covariates, where each element contains the coefficient samples of one group: in a matrix with the attribute `counter` as number of samples in the rows, and the columns are appropriately named to correspond to the single design variables.

fitted values of all models in object, in a matrix with layout models x observations.

samples from the predictive distribution at the covariates given in `newdata`

means of the predictive distribution at the covariates given in `newdata`

See Also

BmaSamples Methods, BayesMfp

Examples

```r
## construct a BayesMfp object
set.seed(19)
x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)
y <- rt(n=15, df=2)
```
test <- BayesMfp (y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 200, method="exhaustive")

## now draw samples from the Bayesian model average
testBma <- BmaSamples (test)
testBma

## We can also draw predictive samples for new data points, but then
## we need to supply the new data to BmaSamples:
newdata <- data.frame(x1 = rnorm(15),
                        x2 = rbinom(n=15, size=5, prob=0.2) + 1,
                        x3 = rexp(n=15))
testBma <- BmaSamples(test, newdata=newdata)
predict(testBma)

## test that inclusion of zero samples works
testBma <- BmaSamples (test, includeZeroSamples=TRUE)
testBma

BmaSamples Methods

Other methods for BmaSamples objects

Description
Print the object (print), get fitted values (fitted) and corresponding residuals (residuals).

Usage

## S3 method for class 'BmaSamples'
print(x, ...)
## S3 method for class 'BmaSamples'
fitted(object, ...)
## S3 method for class 'BmaSamples'
residuals(object, ...)

Arguments

x valid BmaSamples object
object valid BmaSamples object
... unused

Author(s)
Daniel Sabanés Bove

See Also
predict.BmaSamples, summary.BmaSamples
Examples

```r
## construct a BayesMfp object
set.seed(19)

x1 <- rnorm (n=15)
x2 <- rbinom (n=15, size=20, prob=0.5)
x3 <- rexp (n=15)

y <- rt (n=15, df=2)

test <- BayesMfp (y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 200, method="exhaustive")

## now draw samples from the Bayesian model average
testBma <- BmaSamples (test)

## the print method:
testBma

## the fitted method:
fitted(testBma)

## and the corresponding residuals:
residuals(testBma)
```

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.

Author(s)

Daniel Sabanès Bové
Extract.BayesMfp

Examples

## draw standard normal variates
test <- rnorm(n=1000)

## estimate the 95% HPD interval with these samples:
empiricalHpd(theta=test, level=0.95)

## compare with true HPD:
qnorm(p=c(0.025, 0.975))

---

Extract.BayesMfp

Extract method for BayesMfp objects

Description

Extract a subset of models from a BayesMfp object.

Usage

## S3 method for class 'BayesMfp'
x[...]

Arguments

x valid BayesMfp object

... transports the indexes of the models

Author(s)

Daniel Sabanès Bove

See Also

BayesMfp

Examples

## generate a BayesMfp object
set.seed(19)

x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)
y <- rt(n=15, df=2)

test <- BayesMfp(y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 100, method="exhaustive")
findModel

Find a specific fractional polynomial model in a BayesMfp object

Description
Returns the index of the wished model if it is present in the model list, and otherwise returns NA.

Usage
findModel(model, BayesMfpObject)

Arguments
- model: the specific model: a list with entries powers and ucTerms
- BayesMfpObject: an object of class BayesMfp

Details
See BayesMfp for the description of a model.

Value
Index of model in BayesMfpObject if it is present in the model list, otherwise NA.

Note
The searched model must have exactly the same construction as the models in BayesMfpObject. See the example below for the recommended use.

Examples

```r
## extract the top ten models
test[1:10]

findModel

## construct a BayesMfp object
set.seed(92)

x1 <- rnorm(15)
x2 <- rbinom(n=15, size=20, prob=0.6)
x3 <- rexp(15)
y <- rt(15, df=2)

test <- BayesMfp(y ~ bfp(x2, max = 4) + uc(x1 + x3), nModels=2000, method="exhaustive")

## copy one model
myModel <- test[[1]]

## and modify it!
myModel$powers["x2"] <- c(1, 2)
stopifnot(identical(findModel(myModel, test), 31L))
```
**getPosteriorParms**

*Extract updated posterior parameters for the normal inverse gamma distribution from a model, given a shrinkage factor.*

**Description**

Conditional on a fixed shrinkage factor $t = g/(g+1)$, the posterior joint distribution of the effects and the regression variance is normal inverse gamma. With this function, you can compute the parameters of this distribution.

**Usage**

```r
getPosteriorParms(x, shrinkage=x[[1]]$postExpectedShrinkage,
    design = getDesignMatrix(x))
```

**Arguments**

- `x`  
  a valid `BayesMfp`-Object, only first list element will be recognized
- `shrinkage`  
  shrinkage factor used in the computations (defaults to the posterior expected shrinkage factor in the model `x[[1]]`)
- `design`  
  (centered) design matrix for the model

**Value**

A list with four parameters:

- `aStar`  
  the first parameter of the inverse gamma distribution
- `VStar`  
  the covariance matrix part of the multivariate normal distribution
- `mStar`  
  the expectation of the multivariate normal distribution
- `bStar`  
  the second parameter of the inverse gamma distribution

**Author(s)**

Daniel Sabanés Bové

**Examples**

```r
## construct a BayesMfp object
data.seed(19)

x1 <- rnorm (n=15)
x2 <- rbinom (n=15, size=20, prob=0.5)
x3 <- rexp (n=15)
y <- rt (n=15, df=2)

test <- BayesMfp (y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 200, method="exhaustive")
```
# inclusionProbs

## Description

Compute (model averaged) posterior variable inclusion probabilities for the uncertain variables (including FP variables) based on a `BayesMfp` object.

## Usage

```r
inclusionProbs(BayesMfpObject, postProbs = posteriors(BayesMfpObject, ind = 1))
```

## Arguments

- `BayesMfpObject`: valid `BayesMfp` object
- `postProbs`: posterior probabilities to weight the models (defaults to the normalized probability estimates)

## Value

Named numeric vector with the estimated variable inclusion probabilities. Note that these can differ noticeably from the “global” inclusion probabilities computed from all discovered models, from which only the best were saved in the `BayesMfp` object.

## Author(s)

Daniel Sabanés Bové

## Examples

```r
## construct a BayesMfp object
set.seed(19)

x1 <- rnorm (n=15)
x2 <- rbinom (n=15, size=20, prob=0.5)
x3 <- rexp (n=15)

y <- rt (n=15, df=2)

test <- BayesMfp (y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 200, method="exhaustive")

## now get the local inclusion probabilities
local <- inclusionProbs(test)

## they can be compared with the global inclusion probabilities
local - attr(test, "inclusionProbs")
```
Description

This is the Ozone data discussed in Breiman and Friedman (JASA, 1985, p. 580). These data are for 330 days in 1976. All measurements are in the area of Upland, CA, east of Los Angeles.

Usage

data(ozone)

Format

A data frame with 366 observations on the following 13 variables.

- month: month of the year
- day: day of the month
- weekday: day of the week: a factor with levels Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
- hourAverageMax: maximum 1-hour average ozone level [ppm]
- pressure500Height: 500 millibar pressure height [meters]
- windSpeed: wind speed [mph]
- humidity: relative humidity [%]
- tempSandburg: temperature at Sandberg, CA [degrees F]
- tempElMonte: temperature at El Monte, CA [degrees F]
- inversionBaseHeight: inversion base height [feet]
- pressureGradientDaggett: pressure gradient from LAX to Daggett, CA [mm Hg]
- inversionBaseTemp: inversion base temperature [degrees F]
- visibility: visibility [miles]

Source

plotCurveEstimate

Generic function for plotting a fractional polynomial curve estimate

Description

Plot a fractional polynomial curve estimate for either a single model or a Bayesian model average over `BayesMfp` objects. Optionally, credible intervals and / or bands can be added to the plot.

Usage

```r
plotCurveEstimate(model, termName, plevel = 0.95, slevel = plevel,
plot = TRUE, legendPos = "topleft", rug = FALSE, partialResids=TRUE,
hpd=TRUE,..., main = NULL)
```

Arguments

- `model`: an object of class `BayesMfp` or `BmaSamples`
- `termName`: string denoting an FP term, as written by the `summary` method
- `plevel`: credible level for pointwise intervals, and NULL means no pointwise intervals (default: 0.95)
- `slevel`: credible level for simultaneous credible band (SCB), NULL means no SCB (defaults to `plevel`)
- `plot`: if FALSE, only return values needed to produce the plot, but do not plot (default is TRUE, so a plot is made)
- `legendPos`: position of coefficient estimates (for `BayesMfp`) or sample size (for `BmaSamples`) in the plot, NULL suppresses the printing (default is “topleft”)
- `rug`: add a rug to the plot? (default: FALSE)
- `partialResids`: add partial residuals to the plot? (default: TRUE)
- `hpd`: use HPD intervals (TRUE, default) or quantile-based (FALSE) intervals?
- `...`: further arguments in case of a `BayesMfp` object (see details) and arguments for plotting with `matplot`
- `main`: optional main argument for the plot

Details

Further arguments for application on a `BayesMfp` object:

- `grid`: vector of unscaled abscissae, default is a length `gridSize` grid over the observed range specified by providing the argument NULL.
- `post`: list with posterior parameters of the model, which may be provided manually to accelerate plotting in a loop
- `gridSize`: default number of grid points used when no grid is supplied (default is 201)
- `numSim`: number of simulations for estimation of the SCB (default is 500)
plotCurveEstimate

Value

a list of various plotting information:

- **original**: grid on the original covariate scale
- **grid**: grid on the transformed scale
- **mode**: mode curve values, only for `BayesMfp` object
- **mean**: pointwise mean curve values, only for `BmaSamples` object
- **median**: pointwise median curve values, only for `BmaSamples` object
- **plower**: lower boundaries for pointwise intervals
- **pupper**: upper boundaries for pointwise intervals
- **slower**: lower boundaries for SCB
- **supper**: upper boundaries for SCB
- **obsVals**: observed values of the covariate on the original scale
- **sampleSize**: sample size underlying the curve estimate, only for `BmaSamples` object
- **partialResids**: partial residuals
- **transform**: vector of shift and scale parameter

See Also

`BayesMfp`, `BmaSamples`

Examples

```r
## construct a BayesMfp object
set.seed(19)
x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)
y <- rt(n=15, df=2)

test <- BayesMfp (y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 100,
method="exhaustive")

## plot the x2 curve estimate for the 20-th best model
p1 <- plotCurveEstimate (test[20], "x2")

## look at the returned list
str(p1)

## plot the BMA curve estimate for the same covariate
testBma <- BmaSamples (test)
p2 <- plotCurveEstimate (testBma, "x2")

## look at the returned list
str(p2)

## try the new options:
plotCurveEstimate (testBma, "x2", partialResids=FALSE, hpd=FALSE)
```
Extract posterior model probability estimates from BayesMfp objects

Description

Extract posterior model probability estimates (either normalized estimates or sampling frequencies) from BayesMfp objects.

Usage

posteriors(BayesMfpObject, ind = 1)

Arguments

BayesMfpObject a valid BayesMfp object, containing the models the probabilities of which one wants to estimate

ind ind = 1 means normalized posteriors, ind = 2 means sampling frequencies

Value

The vector of probability estimates.

Author(s)

Daniel Sabanés Bové

Examples

```r
## construct a BayesMfp object
set.seed(19)
x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)
y <- rt(n=15, df=2)

test <- BayesMfp(y ~ bfp(x1, max = 2) + bfp(x2, max = 2) + uc(x3), nModels = 100,
method="exhaustive")

## this works:
posteriors(test)

## this must not work:
## SoDA::muststop(posteriors(test, ind=2))

## only if we do model sampling there are model frequencies:
test2 <- BayesMfp(y ~ bfp(x1, max = 2) + bfp(x2, max = 2) + uc(x3), nModels = 100,
method="sampling")
posteriors(test2, ind=2)
```
predict.BayesMfp  Predict method for BayesMfp objects

Description

Predict new responses from a single multiple FP model.

Usage

```r
## S3 method for class 'BayesMfp'
predict(object, newdata, ...)
```

Arguments

- `object`: valid `BayesMfp` object, from which only the first model will be used.
- `newdata`: new covariate data with exactly the names (and preferably ranges) as for the original `BayesMfp` call
- `...`: unused

Author(s)

Daniel Sabanés Bové

See Also

`bmaPredict`

Examples

```r
## generate a BayesMfp object
set.seed(19)

x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)

y <- rt(n=15, df=2)

test <- BayesMfp(y ~ bfp (x2, max = 4) + uc (x1 + x3), nModels = 100, method="exhaustive")

## predict new responses at (again random) covariates
predict(test,
   newdata = list(x1 = rnorm (15),
                 x2 = rbinom (n=15, size=5, prob=0.2) + 1,
                 x3 = rexp (15)))
```
predict.BmaSamples

Predict method to extract point and interval predictions from BmaSamples objects

Description

Predict new responses from a Bayesian model average over FP models, from which predictive samples have already been produced.

Usage

```r
## S3 method for class 'BmaSamples'
predict(object, level=0.95, hpd=TRUE, ...)
## S3 method for class 'predict.BmaSamples'
predict(x, ...)
```

Arguments

- `object`: valid BmaSamples object
- `level`: credible level for the credible intervals (default: 95%)
- `hpd`: should empirical hpd intervals be used (default) or simple quantile-based?
- `...`: unused
- `x`: object of S3 class predict.BmaSamples

Details

This function summarizes the predictive samples saved in the BmaSamples object. Using these functions, one can obtain predictive credible intervals, as opposed to just using the function bmaPredict, which only gives the means of the predictive distributions.

Value

A list of class predict.BmaSamples, which has then a separate print method. The elements of the list are:

- `intervalType`: which credible intervals have been computed (either “HPD” or “equitailed”)
- `level`: the credible level
- `newdata`: the covariate data for the predicted data points (just copied from object)
- `sampleSize`: the sample size (just copied from object)
- `nModels`: the number of models (just copied from object)
- `summaryMat`: the summary matrix for the predictions, with median, mean, lower and upper credible interval borders.

Author(s)

Daniel Sabanès Bové
Simultaneous credible band computation (Besag algorithm)

Description

Simultaneous credible band computation

Usage

scrBesag(samples, 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
level the credible level (default: 0.95)

Details

Calculates a series of simultaneous credible bounds for one parameter type.

Value

matrix with ‘lower’ and ‘upper’ rows
**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**  
```r
scrHpd(samples, mode = apply(samples, 2, median), level = 0.95)
```

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

**Details**  
This function first computes the matrix of absolute distances of the samples to the mode vector. Then based on this distance matrix, a one-sided SCB as described in Besag et al. (1995) is computed, which is then mapped back to the samples.

**Value**  
A matrix with rows “lower” and “upper”, with the lower and upper SCB bounds.

**Author(s)**  
Daniel Saban\'es Bov\'e

**References**  
Summary of BayesMfp object

See Also

empiricalHpd

Examples

```r
## create some samples
time <- 1:10
nSamples <- 50
samples <- t(replicate(nSamples,
    time * rnorm(1) + rexp(1)) + 
    rnorm(length(time) * nSamples)
matplot(time, t(samples), type="l", lty=1, col=1,
    xlab="time", ylab="response")

## now test the function: 50% credible band
scb <- scHpd(samples, level=0.5)
matlines(time, t(scb), col=2, lwd=2, lty=1)
```

Summary of BayesMfp object

*Calculate and print the summary of a BayesMfp object*

Description

Calculate and print the summary of a BayesMfp object, using S3 methods for the class.

Usage

```r
## S3 method for class 'BayesMfp'
summary(object, level=0.95, table=TRUE,
    shrinkage=NULL, ...)
## S3 method for class 'summary.BayesMfp'
print(x, ...)
```

Arguments

- **object**
  a valid BayesMfp object
- **x**
  a return value of `summary.BayesMfp`
- **level**
  credible level for coefficients HPD intervals (default: 0.95)
- **table**
  should a data.frame of the models be included? (default)
- **shrinkage**
  shrinkage factor used, where NULL defaults to the posterior expected shrinkage factor
- **...**
  only used by `summary.BayesMfp` to pass arguments to `as.data.frame.BayesMfp`
Summary of BayesMfp object

Value

`summary.BayesMfp` returns a list with S3 class `summary.BayesMfp`, where the arguments “call”, “numVisited”, “termNames”, “shiftScaleMax”, “inclusionProbs”, “chainlength” (only for model sampling results) are copied from the attributes of the `BayesMfp` object, please see its help page for details.

The other elements are:

- `dataframe` the model overview as data.frame (only if `table=TRUE` was specified)
- `localInclusionProbs` local variable inclusion probability estimates
- `nModels` number of models contained in object

If there are multiple models in object, the list element `postProbs` contains the exact (for exhaustively explored model spaces) or estimated (if model sampling has been done) posterior model probabilities.

If object contains only one FP model, then this one is summarized in more detail:

- `level` used credible level for coefficients HPD intervals
- `shrinkage` used shrinkage factor
- `summaryMat` matrix with posterior summaries of the single coefficients: “mode” gives the posterior mode, “HPDlower” and “HPDuppper” give the boundaries of the HPD intervals with specified credible level
- `sigma2Sum` posterior summary for the regression variance: again mode, and lower and upper HPD bounds are given in a rowvector.

Note

Note that if you extract the summary of a single model with these functions, you ignore the uncertainty about the shrinkage factor $t = g/(g+1)$ by plugging in the number `shrinkage`. If you want to incorporate this uncertainty, you must run `BmaSamples` on this model and call the corresponding method `summary.BmaSamples`.

Author(s)

Daniel Sabanès Bové

See Also

`summary.BmaSamples`

Examples

```r
## generate a BayesMfp object
set.seed(19)

x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)
```
y <- rt(n=15, df=2)

test <- BayesMfp(y ~ bfp(x2, max = 4) + uc(x1 + x3), nModels = 100, 
method="exhaustive")

## summary of multiple models:
summary(test)

## summary of just one model (no. 10):
summary(test[10])

## internal structure is usually not interesting:
str(summary(test[10]))

Summary of BmaSamples object

Calculate and print the summary of a BmaSamples object

Description

Calculate and print the summary of a BmaSamples object, using S3 methods for the class.

Usage

## S3 method for class 'BmaSamples'
summary(object, level = 0.95, hpd = TRUE, ...)
## S3 method for class 'summary.BmaSamples'
print(x, table = TRUE, ...)

Arguments

object 
a valid BmaSamples object
level 
credible level for coefficients credible intervals
hpd 
should emprical hpd intervals be used (default) or simple quantile-based?
x 
a return value of summary.BmaSamples
table should the model table been shown? (default)
... unused

Value

The summary method returns an S3 object, where “sampleSize”, “modelData” and “modelFregs” are copied from the BmaSamples object, please see its help page for the details. “intervalType” and “level” copy the function’s parameters.

“summaryMat” contains the posterior summaries for the intercept and uncertain fixed form co-variates. “sigma2Sum” and “shrinkageSum” contain the posterior summaries for the regression variance and the shrinkage factor, respectively. The summaries are always the median, mean, lower and upper credible bounds for the coefficients.
Author(s)
Daniel Sabanes Bove

See Also
summary.Nbayesmfp

Examples

```r
## generate a BmaSamples object
set.seed(19)

x1 <- rnorm(n=15)
x2 <- rbinom(n=15, size=20, prob=0.5)
x3 <- rexp(n=15)

y <- rt(n=15, df=2)

test <- BayesMfp(y ~ bfp(x2, max = 4) + uc (x1 + x3), nModels = 100,
  method="exhaustive")

testBma <- BmaSamples(test)

## look at the summary
summary(testBma)

## and its structure
str(summary(testBma))
```
Index

*Topic datasets
  ozone, 17

*Topic htest
  empiricalHpd, 12
  inclusionProbs, 16
  scrBesag, 23
  scrHpd, 24

*Topic methods
  as.data.frame.BayesMfp, 2
  BayesMfp Methods, 6
  BmaSamples Methods, 11
  Extract.BayesMfp, 13
  predict.BayesMfp, 21
  predict.BmaSamples, 22
  Summary of BayesMfp object, 25
  Summary of BmaSamples object, 27

*Topic multivariate
  scrHpd, 24

*Topic regression
  BayesMfp, 3
  bmaPredict, 7
  BmaSamples, 9
  getPosteriorParms, 15
  plotCurveEstimate, 18

*Topic utilities
  findModel, 14
  posteriors, 20
  [.BayesMfp, 5
  [.BayesMfp (Extract.BayesMfp), 13
  as.data.frame.BayesMfp, 2, 25
  BayesMfp, 2, 3, 6–10, 13–16, 18–21, 25, 26
  BayesMfp Methods, 6, 6
  bfp, 4
  bfp(BayesMfp), 3
  bmaPredict, 7, 21–23
  BmaSamples, 6, 8, 9, 11, 18, 19, 22, 26, 27
  BmaSamples Methods, 10, 11
  empiricalHpd, 12, 25
  Extract.BayesMfp, 13
  findModel, 14
  fitted.BayesMfp (BayesMfp Methods), 6
  fitted.BmaSamples (BmaSamples Methods), 11
  getPosteriorParms, 15
  inclusionProbs, 16
  matplot, 18
  ozone, 17
  plotCurveEstimate, 18
  posteriors, 20
  predict.BayesMfp, 21
  predict.BmaSamples, 11, 22, 22
  print.BayesMfp (BayesMfp Methods), 6
  print.BmaSamples (BmaSamples Methods), 11
  print.predict.BmaSamples
    (predict.BmaSamples), 22
  print.summary.BayesMfp (Summary of
    BayesMfp object), 25
  print.summary.BmaSamples (Summary of
    BmaSamples object), 27
  residuals.BayesMfp (BayesMfp Methods), 6
  residuals.BmaSamples (BmaSamples
    Methods), 11
  scrBesag, 23
  scrHpd, 24
  summary, 18
  Summary of BayesMfp object, 25
  Summary of BmaSamples object, 27
  summary.BayesMfp, 3, 23, 26, 28
summary.BayesMfp (Summary of BayesMfp object), 25
summary.BmaSamples, 11, 26, 27
summary.BmaSamples (Summary of BmaSamples object), 27
uc, 4
uc (BayesMfp), 3