

Package: mvBayes (via r-universe)

June 2, 2026

Type Package

Title Multivariate Bayesian Regression

Version 1.2.1.9000

Description Fit, plot, and predict a multivariate response, using an arbitrary univariate Bayesian regression model to independently fit basis components (e.g., principal components) of the response (Francom et al., 2025 <[DOI:10.1137/24M1644092](https://doi.org/10.1137/24M1644092)>).

License MIT + file LICENSE

Imports MASS, methods, BASS, latex2exp, cli, splines

Suggests fdasrvf, BayesPPR, knitr, rmarkdown

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.0.0

URL <https://github.com/sandialabs/mvBayesR>

BugReports <https://github.com/sandialabs/mvBayesR/issues>

VignetteBuilder knitr

Config/pak/sysreqs pari-gp

Repository <https://sandialabs.r-universe.dev>

Date/Publication 2026-06-01 12:23:29 UTC

RemoteUrl <https://github.com/sandialabs/mvbayesr>

RemoteRef HEAD

RemoteSha 381df5ecf147ebc5bc113dd4fd2ac18f44a704c4

Contents

basisSetup	2
basisSetupElastic	3
mvBayes	4
mvBayesElastic	6

mvCV	8
mvSobol	9
plot.basisSetup	10
plot.mvBayes	11
plot.sobol	12
predict.mvBayes	14
traceplot	15

Index	17
--------------	-----------

basisSetup	<i>Basis Expansion Calculation</i>
------------	------------------------------------

Description

Calculates a basis expansion of a specified type.

Usage

```
basisSetup(
  Y,
  basisType = "pca",
  customBasis = NULL,
  nBasis = NULL,
  propVarExplained = 0.99,
  center = TRUE,
  scale = FALSE
)
```

Arguments

Y	A matrix of dimension 'c(n, nMV)', where 'n' is the number of observations of 'nMV' response variables
basisType	The type of basis functions to use. Options are 'pca', 'pns', 'jfpca', 'jfpcah', 'splinet', 'bspline', or 'legendre'.
customBasis	Optional user-provided basis of dimension 'c(nBasis, nMV)', only used if 'basisType==custom'.
nBasis	integer number of basis components to use (optional). By default, 'propVarExplained' is used to choose nBasis.
propVarExplained	Proportion (between 0 and 1) of variation to explain when choosing the number of basis components. Only used if 'is.null(nBasis)'.
center	logical: whether or to center 'Y' before basis computations.
scale	logical: whether to scale 'Y' before basis computations.

Value

object of class 'basisSetup' with plot method.

See Also

Used by the main function [mvBayes](#)

<code>basisSetupElastic</code>	<i>Elastic FDA Basis Expansion Calculation</i>
--------------------------------	--

Description

Calculates a basis expansion of a functional data. Used by the `mvBayes` function to calculate a basis expansion of a response matrix.

Usage

```
basisSetupElastic(
  Y,
  warpData = NULL,
  basisType = "jfpcah",
  nBasis = NULL,
  propVarExplained = 0.99,
  srvf = FALSE
)
```

Arguments

<code>Y</code>	A matrix of dimension $n \times q$, where n is the number of observations of q multi-variate/functional variables
<code>warpData</code>	'time_warping' object from 'fdasrvf' if 'basisType=jfpca' or 'jfpcah'
<code>basisType</code>	The type of basis functions to use. Options are 'jfpca' or 'jfpcah'.
<code>nBasis</code>	An integer specifying the number of basis functions to use. The default is NULL, in which case <code>propVarExplained</code> is used to choose <code>nBasis</code> .
<code>propVarExplained</code>	Proportion (between 0 and 1) of variation to explain when choosing the number of principal components. Only used if <code>nBasis</code> is NULL (which is the default).
<code>srvf</code>	use SRVF if 'basisType=jfpca' or 'jfpcah' default = TRUE

Value

An object of class "basisSetup" containing information about the basis decomposition: "Y" is the original response matrix, "Ycenter" and "Yscale" are vectors of length q specifying the centering and scaling values used for each dimension of Y, "basisType" is the user-specified type of basis, the q -vector "varExplained" specifies the amount of variance explained by each basis in the expansion, matrix "basis" is the $q \times q$ basis matrix (e.g., the principal component vectors in the case of pca), "coefs" is the $n \times q$ matrix of observation-specific basis weights (e.g., the "scores" in the case of pca), and "truncError" is the $n \times q$ matrix of residuals, after accounting for the first `nBasis` bases.

See Also

Used by the main function [mvBayes](#)

mvBayes

Multivariate Bayesian Regression

Description

Wrapper to fit a multivariate response, using an arbitrary univariate Bayesian regression model to independently fit basis components (e.g., principal components) of the response.

Usage

```
mvBayes(
  bayesModel,
  X,
  Y,
  basisType = "pca",
  customBasis = NULL,
  nBasis = NULL,
  propVarExplained = 0.99,
  center = TRUE,
  scale = FALSE,
  nCores = 1,
  samplesExtract = NULL,
  residSDExtract = NULL,
  idxSamplesArg = "idxSamples",
  ...
)
```

Arguments

bayesModel	A Bayesian regression model-fitting function, with first argument taking an $n \times p$ input matrix or data.frame, and second argument taking an n -vector of numeric responses.
X	A matrix of predictors of dimension $n \times p$, where n is the number of training examples and p is the number of inputs (features).
Y	A response matrix of dimension $n \times q$, where q is the number of multivariate/functional responses.
basisType	The type of basis functions to use. Options are 'pca', 'pns', 'splinet', 'bspline', or 'legendre'.
customBasis	Optional user-provided basis of dimension 'c(nBasis, nMV)', only used if 'basisType==custom'.
nBasis	An integer specifying the number of basis functions to use. The default is NULL, in which case propVarExplained is used to choose nBasis.

propVarExplained	Proportion (between 0 and 1) of variation to explain when choosing the number of principal components. Only used if nBasis is NULL (which is the default).
center	A logical argument specifying whether or not to center the responses before computing the basis decomposition.
scale	A logical argument specifying whether or not to scale by the standard deviation of each component of the response before computing the basis decomposition.
nCores	An integer less than or equal to nBasis, specifying the number of threads to use when fitting independent Bayesian models.
samplesExtract	function taking the output of bayesModel ('bm') and extracting posterior samples of all parameters of interest. If 'NULL', mvBayes tries to access 'bm\$samples'; if unsuccessful, an object called 'samples' is created with attribute 'residSD'.
residSDExtract	function taking the output of bayesModel ('bm') and extracting posterior samples of the residual standard deviation ('residSD'). If 'NULL', mvBayes tries to access 'bm\$samples\$residSD'; if unsuccessful, 'residSD' is the standard deviation of the residuals.
idxSamplesArg	str Name of an optional argument of 'predict' controlling which posterior samples are used for posterior prediction.
...	Additional arguments to bayesModel.

Details

First uses the basisSetup function to decompose the response into nBasis components, then independently fits bayesModel to each of those components.

Value

An object of class "mvBayes", which is a list containing "X", an object called "basisInfo" of class "basisSetup" containing information about the basis decomposition, "bayesModel", and "bmList", which contains a list of length nBasis containing fitted model objects for each basis component.

See Also

[basisSetup](#) for computing the basis decomposition, [predict.mvBayes](#) for prediction, [plot.mvBayes](#) for plotting the model fit, [traceplot](#) for monitoring posterior convergence, and [mvSobol](#) for sensitivity analysis.

Examples

```
## simulate data (Friedman function with first variable as functional)
f = function(x) {
  10 * sin(pi * x[, 1] * x[, 2]) + 20 * (x[, 3] - .5)^2 + 10 * x[, 4] + 5 *
  x[, 5]
}
sigma = 1 # noise sd
n = 500 # number of observations
n_grid = 50 # size of functional variable grid
t_grid = seq(0, 1, length.out = n_grid) # functional grid
```

```

X = matrix(runif(n * 9), n, 9) # 9 non-functional variables, only first 4 matter
X_expanded = cbind(rep(t_grid, each = n), kronecker(rep(1, n_grid), X)) # to get y
Y = matrix(f(X_expanded), nrow = n) + rnorm(n * n_grid, 0, sigma)

## fit Multivariate BASS
fit = mvBayes(
  BASS::bass, X, Y, nBasis = 3, # mvBayes parameters
  samplesExtract = function(bm) list(
    s2 = bm$s2,
    nbasis = bm$nbasis,
    beta.prec = bm$beta.prec
  ),
  idxSamplesArg = 'mcmc.use'
)

```

mvBayesElastic

Multivariate Elastic FDA Bayesian Regression

Description

Wrapper to fit a elastic functional response, using an arbitrary univariate Bayesian regression model to independently fit basis components (e.g., principal components) of the response.

Usage

```

mvBayesElastic(
  bayesModel,
  X,
  Y,
  warpData = NULL,
  basisType = "jfpcah",
  nBasis = NULL,
  propVarExplained = 0.99,
  nCores = 1,
  samplesExtract = NULL,
  residSDExtract = NULL,
  idxSamplesArg = "idxSamples",
  srvf = FALSE,
  idx = NULL,
  ...
)

```

Arguments

bayesModel	A Bayesian regression model-fitting function, with first argument taking an n x p input matrix or data.frame, and second argument taking an n-vector of numeric responses.
------------	--

X	A matrix of predictors of dimension $n \times p$, where n is the number of training examples and p is the number of inputs (features).
Y	A response matrix of dimension $n \times q$, where q is the number of multivariate/functional responses.
warpData	'time_warping' object from 'fdasrvf' if 'basisType=jfpca' or 'jfpcah'
basisType	The type of basis functions to use. Options are 'jfpca', 'jfpcah'.
nBasis	An integer specifying the number of basis functions to use. The default is NULL, in which case propVarExplained is used to choose nBasis.
propVarExplained	Proportion (between 0 and 1) of variation to explain when choosing the number of principal components. Only used if nBasis is NULL (which is the default).
nCores	An integer less than or equal to nBasis, specifying the number of threads to use when fitting independent Bayesian models.
samplesExtract	function taking the output of bayesModel ('bm') and extracting posterior samples of all parameters of interest. If 'NULL', mvBayes tries to access 'bm\$samples'; if unsuccessful, an object called 'samples' is created with attribute 'residSD'.
residSDExtract	function taking the output of bayesModel ('bm') and extracting posterior samples of the residual standard deviation ('residSD'). If 'NULL', mvBayes tries to access 'bm\$samples\$residSD'; if unsuccessful, 'residSD' is the standard deviation of the residuals.
idxSamplesArg	str Name of an optional argument of 'predict' controlling which posterior samples are used for posterior prediction.
srvf	use SRVF if 'basisType=jfpca' or 'jfpcah' default = TRUE
idx	vector of indices to subset 'warpData'
...	Additional arguments to bayesModel.

Details

First uses the basisSetup function to decompose the response into nBasis components, then independently fits bayesModel to each of those components.

Value

An object of class "mvBayes", which is a list containing "X", an object called "basisInfo" of class "basisSetup" containing information about the basis decomposition, "bayesModel", and "bmList", which contains a list of length nBasis containing fitted model objects for each basis component.

See Also

[basisSetup](#) for computing the basis decomposition, [predict.mvBayes](#) for prediction, [plot.mvBayes](#) for plotting the model fit, [traceplot](#) for monitoring posterior convergence, and [mvSobol](#) for sensitivity analysis.

Description

Wrapper to fit and perform cross-validation for a multivariate Bayesian regression model, using the mvBayes function.

Usage

```
mvCV(
  bayesModel,
  X,
  Y,
  kFolds = NULL,
  nRep = 1,
  nTrain = NULL,
  nTest = NULL,
  seed = NULL,
  coverageTarget = 0.95,
  idxSamples = "all",
  uqTruncMethod = c("gaussian", "empirical"),
  ...
)
```

Arguments

bayesModel	A Bayesian regression model-fitting function, with first argument taking an nxp input matrix or data.frame, and second argument taking an n-vector of numeric responses.
X	A matrix of predictors of dimension nxp, where n is the total number of examples (including training and test sets) and p is the number of inputs (features).
Y	A response matrix of dimension nxq, where q is the number of multivariate/functional responses.
kFolds	Number of test sets to partition data. If kFolds=NULL (default), test sets are instead formed by random sample, via the nReps, nTrain, and nTest arguments.
nRep	Number of repetitions of CV process. Only used when kFolds=NULL (default).
nTrain	Number of examples to use in the training set. Only used when kFolds=NULL (default). If nTrain=NULL, nTrain is set to n - nTest; unless nTest is also NULL, in which case nTrain is set to ceiling(n/2).
nTest	Number of examples to use in the test set. Only used when kFolds=NULL (default). If nTest=NULL, nTest is set to n - nTrain.
seed	Randomization seed, for replication of the train/test split. The seed is un-initialized immediately after assigning the train/test split. If NULL, no seed is set.
coverageTarget	level of coverage desired (default: 0.95)

`idxSamples` which samples to use in CV (default: "all")
`uqTruncMethod` method to use for UQ truncation (c("gaussian", "empirical"))
`...` Additional arguments to mvBayes, including arguments to bayesModel.

Details

First separates the data into randomly chosen test and training sets (user-specified train/test splits and k-fold cv are forthcoming), then fits `mvBayes(bayesModel, X, Y, ...)` to the training set and evaluates predictive performance on the test set. Repeats this process `nRep` times.

Value

An object of class "mvBayesCV", which is a list containing the out-of-sample rmse for each replication, the fitting and prediction times, and the function call. Other prediction metrics, including coverage of prediction intervals, are forthcoming.

See Also

[mvBayes](#), [predict.mvBayes](#) for prediction

mvSobol

Sobol' Indices for Multivariate Response

Description

Given an object of class "mvBayes" from the `mvBayes()` function, `mvSobol()` calculates the Sobol' indices for each input variable, assuming the response is multivariate. Currently only works if `object$bayesModel` is the `BASS::bass()` function, or if `object$bayesModel` is compatible with the `anova()` function.

Usage

```

mvSobol(
  object,
  totalSobol = TRUE,
  idxSamples = "final",
  nMC = NULL,
  showPlot = FALSE,
  ...
)
  
```

Arguments

`object` An object of class "mvBayes" containing the multivariate Bayesian model fit for a specific basis expansion of a response matrix Y.
`totalSobol` whether to compute the total Sobol' index (summing all interactions)
`idxSamples` Use final MCMC sample

nMC	Number of Monte Carlo iterations to compute, if None will attempt 'anova' if available. Parameter is $\log_2(n)$, i.e., it makes 2^n points
showPlot	whether to compute generate a plot along with the returned Sobol' object
...	Additional arguments to <code>BASS::sobolBasis</code> (other emulator-specific Sobol' calculations to be implemented in the future). Unnecessary for anova-compatible models.

Value

An object of class "bassSob" if `object$bayesModel` is the `BASS::bass()` function, or "mvSobol" otherwise. Contains information about the Sobol' decomposition: See `?BASS::sobolBasis` for more info.

See Also

See [mvBayes](#)

plot.basisSetup

Visualizing the Basis Decomposition

Description

Given an object of class "basisSetup" from the `basisSetup()` function, `plot.basisSetup()` visualizes the decomposition, colored by basis.

Usage

```
## S3 method for class 'basisSetup'
plot(
  x,
  nBasis = NULL,
  propVarExplained = NULL,
  nPlot = NULL,
  idxMV = NULL,
  xscale = "linear",
  xlabel = "Multivariate Index",
  file = NULL,
  title = NULL,
  ...
)
```

Arguments

`x` An object of class "basisSetup" containing information about a basis decomposition of a response matrix `Y`.

nBasis	An integer specifying the number of basis functions to plot. If both nBasis and propVarExplained are NULL, object\$nBasis is used, but anything \leq object\$nBasis is allowed.
propVarExplained	Proportion (between 0 and 1) of variation to represent when choosing the number of bases to plot. It is only used if nBasis is NULL. If both are NULL, object\$nBasis bases are plotted.
nPlot	A positive integer specifying the number of samples to plot. Default is $\min(n, 1000)$, where n is $nrow(Xtest)$ (or $nrow(object\$X)$ if $Xtest$ is NULL).
idxMV	A vector describing time indices
xscale	string whether to plot on a "linear" scale or "log"
xlabel	string for the xlabel
file	An optional location at which the plot will be saved. If NULL, no file is saved.
title	An optional title to be printed at the top of the plot.
...	additional plot arguments

Value

Top left: response Y. Top right: Y decomposed and colored by basis. Bottom left: Truncation error due to dimension reduction in the representation of Y. Bottom right: Percent variance explained by each basis.

See Also

See [mvBayes](#)

plot.mvBayes

Plot the Bayesian Model Fit of a Multivariate Response

Description

Given an object of class "mvBayes" from the mvBayes() function, plot.mvBayes() plots a few aspects of the Bayesian model fit, colored by basis.

Usage

```
## S3 method for class 'mvBayes'
plot(
  x,
  Xtest = NULL,
  Ytest = NULL,
  idxSamples = "final",
  nPlot = NULL,
  idxMV = NULL,
  xscale = "linear",
```

```

    xlabel = "Multivariate Index",
    title = NULL,
    file = NULL,
    ...
)

```

Arguments

x	An object of class "mvBayes" containing the multivariate Bayesian model fit of a response matrix Y.
Xtest	A matrix or data.frame of inputs at which the model fit will be evaluated. If NULL, the training inputs X will be used.
Ytest	A matrix of responses for which the model fit will be evaluated. Should correspond to Xtest. If NULL, the training responses Y will be used.
idxSamples	str which samples to use
nPlot	A positive integer specifying the number of samples to plot. Default is min(n, 1000), where n is nrow(Xtest) (or nrow(object\$X) if Xtest is NULL).
idxMV	A vector describing the sample points
xscale	'linear' or 'log' of x-axis
xlabel	str for x-axis label
title	An optional title to be printed at the top of the diagnostic plots.
file	An optional location at which the plot will be saved. If NULL, no file is saved.
...	additional plot arguments

Value

Top left: Residuals after applying mvBayes(), plotted on top of Ytest (centered if centered=TRUE in the mvBayes() call). Top right: Residuals decomposed and colored by basis. Bottom left: R² values for each component. Bottom right:

See Also

See [mvBayes](#), [predict.mvBayes](#)

plot.sobol

Plot Sobol Decomposition

Description

Given an object of class "mvSobol" from the mvSobol() function

Usage

```
## S3 method for class 'sobol'
plot(
  x,
  totalSobol = TRUE,
  labels = NULL,
  idxMV = NULL,
  xscale = "linear",
  xlabel = "Multivariate Index",
  yOverlay = NULL,
  yOverlayLabel = "Overlay",
  waterfall = FALSE,
  title = NULL,
  file = NULL,
  ...
)
```

Arguments

x	An object of class "mvSobol" containing the Sobol Indices
totalSobol	A boolean to plot the total sobol (default = 'TRUE')
labels	A character vector of length <= 8 containing the names of the parameters
idxMV	A vector defining the time points
xscale	string whether to plot on a "linear" scale or "log"
xlabel	string for the xlabel
yOverlay	A boolean if to overlay the experimental function (default = 'NULL')
yOverlayLabel	A string defining the overlay label
waterfall	bool whether to plot sobol as a waterfall (functional pie-chart) default ('FALSE')
title	An optional title to be printed at the top of the traceplots.
file	An optional location at which the plots will be saved. If NULL, no file is saved.
...	additional plot arguments

Value

no return value

See Also

See [mvBayes](#)

 predict.mvBayes

Posterior Predictive Samples of a Multivariate Response

Description

Given an object of class "mvBayes" from the mvBayes() function, predict.mvBayes() obtains posterior predictive samples at specific input locations. Residual variance from the basis truncation can (optionally) be inserted as well.

Usage

```
## S3 method for class 'mvBayes'
predict(
  object,
  Xtest,
  idxSamples = "default",
  addResidError = FALSE,
  addTruncError = FALSE,
  returnPostCoefs = FALSE,
  idxSamplesArg = NULL,
  ...
)
```

Arguments

object	An object of class "mvBayes" containing the multivariate Bayesian model fit of a response matrix Y.
Xtest	A matrix or data.frame of inputs at which predictions are desired.
idxSamples	A str describing which samples to use
addResidError	A logical to add back in residual error
addTruncError	A logical indicating whether or not to insert random truncation error into the predictions, where the truncation comes from the reduced dimensionality in the representation of Y.
returnPostCoefs	A logical indicating whether or not to output predictions of Ynew (i.e., the observation-specific basis weighting) in addition to predictions of Y.
idxSamplesArg	str Name of an optional argument of 'predict' controlling which posterior samples are used for posterior prediction.
...	Additional arguments to predict.bayesModel, where "bayesModel" is the Bayesian model used in fitting, specified in object\$bayesModel.

Value

If `getPostCoefs==FALSE`, `predict.mvBayes()` outputs an array of dimension `c(nSamples, ntest, nMV)`, where `nSamples` is the number of posterior samples obtained in fitting `bayesModel`, and `n` and `nMV` are respectively the number of rows and columns in `Y`. Elements of this array are posterior predictive samples of `Y`. Otherwise, a list of length two is output, with elements `Ypost` giving samples from the posterior predictive distribution of `Y`, and `postCoefs` giving samples from the posterior predictive distribution of `Ynew`, i.e., from the observation-specific basis weights.

See Also

See [mvBayes](#)

 traceplot

Traceplots from the Bayesian Model Fit of a Multivariate Response

Description

Given an object of class "mvBayes" from the `mvBayes()` function, `traceplot()` plots traceplots of user-specified variables from the Bayesian model fit, colored by basis.

Usage

```
traceplot(
  object,
  modelParams = NULL,
  labels = NULL,
  title = NULL,
  file = NULL,
  ...
)
```

Arguments

<code>object</code>	An object of class "mvBayes" containing the multivariate Bayesian model fit of a response matrix <code>Y</code> .
<code>modelParams</code>	A character vector of length ≤ 8 containing the names of the parameters for which traceplots will be made. If <code>NULL</code> , selects "plottable" attributes of <code>object\$bmList[[1]]</code> .
<code>labels</code>	A character vector of the same length as <code>modelParams</code> , containing the y-axis labels that will be printed for each traceplot. The default is to simply use "modelParams".
<code>title</code>	An optional title to be printed at the top of the traceplots.
<code>file</code>	An optional location at which the traceplots will be saved. If <code>NULL</code> , no file is saved.
<code>...</code>	additional plot arguments

Value

no return value

See Also

See [mvBayes](#)

Index

basisSetup, [2](#), [5](#), [7](#)
basisSetupElastic, [3](#)

mvBayes, [3](#), [4](#), [4](#), [9–13](#), [15](#), [16](#)
mvBayesElastic, [6](#)
mvCV, [8](#)
mvSobol, [5](#), [7](#), [9](#)

plot.basisSetup, [10](#)
plot.mvBayes, [5](#), [7](#), [11](#)
plot.sobol, [12](#)
predict.mvBayes, [5](#), [7](#), [9](#), [12](#), [14](#)

traceplot, [5](#), [7](#), [15](#)