

Package ‘Rbmop’

June 20, 2015

Type Package

Title B-spline MoPs densities and conditional densities

Version 1.0

Date 2015-04-06

Author Gherardo Varando

Maintainer Gherardo Varando <gherardo.varando@gmail.com>

Description Handling, estimation and comparison functions for
B-spline MoPs densities and conditional densities (bmop).

License GPL-3

Imports stats

Suggests rgl,
cubature

R topics documented:

add_knots.bmop	2
AIC.bmop	3
as.bins	3
as.function.bmop	4
BIC.bmop	4
bmopPar	5
bmop_fit	6
bmop_fit.bins	7
bmop_fit.data.frame	7
bmop_fit.default	8
bmop_fit.histogram	9
clean.bmop	9
compare.bmop	10
comparison_plot	10
dim.bmop	11
envelope_plot	12
evaluate.bmop	12
Examples_bmop	13

generate_knots	14
integrate.bmop	15
is.bmop	15
logLik.bmop	16
lower.bmop	16
marginalize.bmop	17
mean.bmop	17
new_bmop	18
normalize.bmop	19
plot.bmop	19
points.bmop	20
print.bmop	21
put_evidence.bmop	21
Rbmop	22
search_bmop	23
summary.bmop	24
upper.bmop	24
Index	25

add_knots.bmop	<i>Add a new knot to a bmop object</i>
----------------	----------------------------------------

Description

Add a new knot to a bmop object

Usage

add_knots.bmop(bmop, value, MARGIN = 1)

Arguments

- | | |
|--------|--------------------------------------------------|
| bmop | a bmop object |
| value | the new knot to be added |
| MARGIN | the dimension where the new knot has to be added |

Value

a bmop object

`AIC.bmop`*Akaike Information Criteria for bmop*

Description

Akaike Information Criteria for bmop

Usage

```
## S3 method for class 'bmop'  
AIC(object, data, corrected = F, ..., k = 2)
```

Arguments

<code>object</code>	a bmop object
<code>data</code>	data.frame, matrix, vector of observations
<code>corrected</code>	logical
<code>...</code>	see AIC
<code>k</code>	penalization coefficients, see AIC

`as.bins`*Bins grouping*

Description

This function extend the histogram class for multi-dimensional datasets

Usage

```
as.bins(data, breaks = nclass.FD, ...)
```

Arguments

<code>data</code>	a dataset
<code>breaks</code>	function or positive integer
<code>...</code>	additional parameters

Value

an object bins, the data are grouped into bins uniformly

<code>as.function.bmop</code>	<i>Convert an bmop object to a function</i>
-------------------------------	---------------------------------------------

Description

Convert an bmop object to a function

Usage

```
## S3 method for class 'bmop'
as.function(x, MIN = 0, ...)
```

Arguments

<code>x</code>	an bmop object
<code>MIN</code>	non negative number, MIN value for evaluation of x
<code>...</code>	compatibility with as.function

Examples

```
bmop<-bmop_fit(rnorm(200))
fun<-as.function(bmop)
fun(0)
fun(3)
plot(fun,from=-2,to=+2)
```

<code>BIC.bmop</code>	<i>Bayesian Information Criteria for bmop</i>
-----------------------	-----------------------------------------------

Description

Bayesian Information Criteria for bmop

Usage

```
## S3 method for class 'bmop'
BIC(object, data, corrected = F, ...)
```

Arguments

<code>object</code>	a bmop object
<code>data</code>	data.frame, matrix, vector of observations
<code>corrected</code>	logical
<code>...</code>	see BIC

bmopPar

*Set Rbmop parameters***Description**

Set appropriate global parameters to be used by Rbmop functions . This is the appropriate way to change those parameters. The use is similar to `par()` in package `graphics`.

Usage

```
bmopPar(...)
```

Arguments

... see details.

Details

This function set parameters used for estimation. The call `bmopPar()` just print the values of the paramters.

`mle=FALSE`: logical. If use maximum-likelihood estimation of bmop coefficient, setting `mle=TRUE` just force `repMax=1`.

`N=NA`: If present, the number of knots in every dimensions. Vector of positive integer, if needed values will be recycled.

`order=3`: The order of the B-spline in every dimensions, vector of positive integer, if needed values will be recycled.

`alpha=3`: The penalization exponent to compute the number of knots. This is the default method to compute the number of knots, with the formula: $\text{floor}(n^{1/\alpha})$, where n is the number of observations in the dataset. If `!is.na(N)` then the number of knots will be set to N^d where d is the number of dimensions in the dataset (num. of variables).

`knotsMethod="uniform"`: "uniform" or "quantiles" knots, how knots are computed by [generate_knots](#).

`k=2`: Coefficient of AIC (penalized likelihood), positive integer or "BIC" string. This is used by [search_bmop](#).

`toll=10^{-10}`: Tollerance for the increment of the likelihood in the mle estimation of the coefficient.

`repMax=100`: maximum number of iteration in the mle estimation of the coefficient

`MIN=10^{-10}`: This is not a learning parameter but instead define the MIN parameter in the evaluation of bmop object. Observe that some functions like [logLik](#) or [plot](#), set this parameter independently.

`autoReduce=200`: This value set the maximum dimension of an accepted dataset as raw-data, for larger dataset, functions [bmop_fit](#) and [search_bmop](#) will be applied over the reduced bins (histogram). Setting it to Inf disable this features.

bmop_fit

*Estimation of bmop density or conditional density***Description**

Estimation of bmop density or conditional density

Usage

```
bmop_fit(data, conditional = F, Min = NULL, Max = NULL, ...)
```

Arguments

data	data.frame, matrix, vector or object of class "histogram" or "bins"
conditional	logic, if TRUE a conditional density is learned
Min	vector of lower bounds
Max	vector of upper bounds
...	see bmopPar

Value

a bmop object, a density if conditional=FALSE or a conditional density if conditional=TRUE. In the latter case the first variable of the dataset is considered as a de conditioned one and the rest of the variables as the conditioning ones. If the dataset has only one variable a normal density is generated discarding the value of conditional.

Examples

```
plot(bmop_fit(rnorm(100)))
plot(bmop_fit(hist(rnorm(100000))))
#####
Data<-data.frame(rnorm(100),rexp(100))
bmop<-bmop_fit(Data)
plot(bmop)
#####
X<-rnorm(100)
Y<-rnorm(100,mean=X)
Data<-data.frame(X,Y)
bmopPar(mle=TRUE)
bmopC<-bmop_fit(Data,conditional=TRUE)
plot(bmopC)
```

bmop_fit.bins	<i>Estimation of bmop density or conditional density</i>
---------------	----------------------------------------------------------

Description

Estimation of bmop density or conditional density

Usage

```
## S3 method for class 'bins'
bmop_fit(data, conditional = F, Min = NULL, Max = NULL,
         bmop = NULL, ...)
```

Arguments

data	histogram or bins object the variables must be in the right order
conditional	logic, if TRUE a conditional density is learned
Min	vector of lower bounds
Max	vector of upper bounds
bmop	a bmop object
...	see bmopPar

Value

a bmop object

bmop_fit.data.frame	<i>Estimation of bmop density or conditional density</i>
---------------------	----------------------------------------------------------

Description

Estimation of bmop density or conditional density

Usage

```
## S3 method for class 'data.frame'
bmop_fit(data, conditional = F, Min = NULL,
         Max = NULL, bmop = NULL, ...)
```

Arguments

data	data.frame, matrix or vector the variables must be in the right order (the columns of data)
conditional	logic, if TRUE a conditional density is learned
Min	vector of lower bounds
Max	vector of upper bounds
bmop	a bmop object
...	see bmopPar

Value

a bmop object

bmop_fit.default	<i>Estimation of bmop density or conditional density</i>
------------------	----------------------------------------------------------

Description

Estimation of bmop density or conditional density

Usage

```
## Default S3 method:
bmop_fit(data, conditional = F, Min = NULL, Max = NULL,
  ...)
```

Arguments

data	data.frame, matrix, vector or object of class "histogram" or "bins"
conditional	logic, if TRUE a conditional density is learned
Min	vector of lower bounds
Max	vector of upper bounds
...	see bmopPar

Value

a bmop object

bmop_fit.histogram	<i>Estimation of bmop density or conditional density</i>
--------------------	----------------------------------------------------------

Description

Estimation of bmop density or conditional density

Usage

```
## S3 method for class 'histogram'  
bmop_fit(data, conditional = F, Min = NULL,  
          Max = NULL, bmop = NULL, ...)
```

Arguments

data	histogram or bins object the variables must be in the right order
conditional	logic, if TRUE a conditional density is learned
Min	vector of lower bounds
Max	vector of upper bounds
bmop	a bmop object
...	see bmopPar

Value

a bmop object

clean.bmop	<i>Clean a bmop object</i>
------------	----------------------------

Description

remove extra informations from bmop object

Usage

```
clean.bmop(object)
```

Arguments

object	a bmop object
--------	---------------

Value

a bmop object

compare.bmop	<i>Compare bmop with true density</i>
--------------	---------------------------------------

Description

Compare bmop with true density

Usage

```
compare.bmop(object, dtrue, measure = "MSE", ...)
```

Arguments

object	bmop object
dtrue	function
measure	string, "MSE", "MAE", "MAX"
...	optional arguments to be passed to dtrue

Examples

```
data<-rnorm(200)
bmop1<-bmop_fit(data)
bmopPar(mle=TRUE)
bmop2<-bmop_fit(data)
compare.bmop(bmop1,dtrue=dnorm)
```

comparison_plot	<i>Plot several bmops and true density</i>
-----------------	--------------------------------------------

Description

Plot several bmops and true density

Usage

```
comparison_plot(bmop.list, dtrue = NULL, colors = NULL, lwd = 3,
  type = "l", type.true = "l", col.true = "red",
  names.bmop = 1:length(bmop.list), legend.display = T,
  legend.pos = "topleft", file = NULL, ylim = c(0, 1), ...)
```

Arguments

bmop.list	list of bmop objects (1 dimensional)
dtrue	function, true density function
colors	NULL or vector of colors
lwd	graphical parameter see par
type	graphical parameter see par
type.true	as type, but just for true density plot
col.true	as col graphical parameter but for dtrue density plot
names.bmop	vector for the names showed in legend
legend.display	logical, to show or not the legend
legend.pos	string for the position of the legend "top", "left", "topright", etc.
file	an optional file name to save the plot
ylim	parameter to set the plot correctly, set to NULL for automatic setting
...	more parameters to be passed to dtrue

Value

invisible()

Examples

```
data<-rnorm(200)
bmop1<-bmop_fit(data)
bmopPar(mle=TRUE)
bmop2<-bmop_fit(data)
comparison_plot(list(bmop1,bmop2),true=dnorm,
                    names.bmop=c("Fast","MLE"))
```

dim.bmop	<i>Dimension of bmop</i>
----------	--------------------------

Description

Dimension of bmop

Usage

```
## S3 method for class 'bmop'
dim(x)
```

Arguments

x	a bmop object
---	---------------

Value

the number of free parameters of x

envelope_plot	<i>Envelop of estimated bmop</i>
---------------	----------------------------------

Description

This function plots various bmop estimations, from differents datasets.

Usage

```
envelope_plot(n = 100, N = 50, rtrue = rnorm, fun = bmop_fit,
  dtrue = dnorm, lwd = 3, type = "l", col.true = "red", ...)
```

Arguments

n	number of bmop densities to learn from different datasets
N	number of observations in every dataset
rtrue	function to generate samples, see rnorm
fun	a learning function of bmop as bmop_fit
dtrue	true density function, see dnorm
lwd	graphical par
type	graphical par
col.true	the color for the true density
...	additional parameters to be passed to rtrue and dtrue

Value

invisible()

Examples

```
envelope_plot(n=50,N=50,rtrue=rexp,dtrue=dexp)
```

evaluate.bmop	<i>Evaluation of a bmop object</i>
---------------	------------------------------------

Description

Evaluation of a bmop object

Usage

```
evaluate.bmop(x, object, MIN = bmopPar()$MIN)
```

Arguments

x	numeric, vector, matrix
object	bmop object
MIN	numeric

Details

The deBoor algorithm is used, implemented in C.

Value

Numeric value (values) of the computed bspline at point (points) x.

References

Carl de Boor, On calculating with B-splines, *Journal of Approximation Theory*, Volume 6, Issue 1, July 1972, Pages 50-62, <http://www.sciencedirect.com/science/article/pii/0021904572900809>.

Examples

```
bmop<-bmop_fit(rnorm(100))
evaluate.bmop(0,bmop)
evaluate.bmop(c(-1,0,+1),bmop)
```

Examples_bmop

Examples of bmop density estimations

Description

Various examples to show the capabilities of bmop estimations.

Usage

```
ex_bmop_gaussian2Mixture(N = 1000, m1 = -3, m2 = 0, lambda = 0.5)

ex_bmop_gaussian3Mixture(N = 1000, m1 = -3, m2 = 0, m3 = +3,
  lambda = c(1, 1, 1))

ex_bmop_gaussianBetaGamma(N = 1000, m1 = -3, m2 = 2, m3 = 3,
  lambda = c(1, 1, 1))
```

Arguments

N	positive integer, the number of observations
m1, m2, m3	location parameters
lambda	mixing coefficient, vector or double

Details

This functions generate datasets of N observations. The function `bmap_fit` is then used to estimate the density. A comparative plot is then returned.

`ex_bmap_gaussian2Mixture` lambda-mixture of two Gaussian densities with unitary variance and means `m1` and `m3`.

`ex_bmap_gaussian3Mixture` lambda-mixture of three Gaussian densities with unitary variance and means `m1`, `m2` and `m3`.

`ex_bmap_gaussianBetaGamma` lambda-mixture of a Gaussian density with unitary variance and mean `m1`, a Beta density with `shape1=2` `shape2=5` `ncp=m2` and a Gamma density with `shape=9` `scale=m3/9`.

Value

All the functions return an invisible list containing the generated dataset, the estimated bmops and the true density function. See example on how, for example, plot the default kernel density estimation on the same dataset.

Examples

```
Ex<-ex_bmap_gaussian2Mixture()
points(density(Ex$data),type="l",col="blue",lwd=3)
```

<code>generate_knots</code>	<i>Generate sequence of knots for bmap objects</i>
-----------------------------	----------------------------------------------------

Description

Generate sequence of knots for bmap objects

Usage

```
generate_knots(data = NULL, N = 5, method = "uniform", Min = NULL,
  Max = NULL)
```

Arguments

<code>data</code>	data.frame matrix vector
<code>N</code>	positive integer, the number of knots
<code>method</code>	string "uniform" or "quantiles"
<code>Min</code>	numeric
<code>Max</code>	numeric

Examples

```
data<-rnorm(200)
bmopE<-bmop_fit(data)
bmopS<-search_bmop(data)
plot(bmopE)
points(bmopS,type="l",col="red")
```

integrate.bmop	<i>Integrate bmop object over the support</i>
----------------	-----------------------------------------------

Description

Integrate bmop object over the support

Usage

```
integrate.bmop(object)
```

Arguments

object	bmop object
--------	-------------

Examples

```
data<-rnorm(200)
bmop<-bmop_fit(data)
integrate.bmop(bmop)
```

is.bmop	<i>Check if an object's class is bmop</i>
---------	-------------------------------------------

Description

Check if an object's class is bmop

Usage

```
is.bmop(object)
```

Arguments

object	an R object
--------	-------------

Details

Just a call of inherits(object,"bmop")

Value

logical, TRUE if "bmop" is present among class(object), FALSE otherwise

Examples

```
bmop<-bmop_fit(rexp(100))
is.bmop(bmop)
```

logLik.bmop	<i>Log-Likelihood of bmop object</i>
-------------	--------------------------------------

Description

Log-Likelihood of bmop object

Usage

```
## S3 method for class 'bmop'
logLik(object, data, ...)
```

Arguments

- object a bmop object
- data matrix, data.frame or vector of observation
- ... some methods require additional arguments, see [AIC](#)

See Also

AIC.bmop

lower.bmop	<i>Lower Limit of bmop</i>
------------	----------------------------

Description

Lower Limit of bmop

Usage

```
lower.bmop(object)
```

Arguments

- object a bmop object

Value

numeric value

marginalize.bmop	<i>Marginalize a bmop</i>
------------------	---------------------------

Description

marginalize a bmop object, that is integrate a bmop multivariate density over some of its dimension.

Usage

```
marginalize.bmop(object, MARGIN = 1, ...)
```

Arguments

object	a bmop object
MARGIN	positive integer or vector of positive integer, the dimensions that has to be marginalized
...	additional parameters

Value

a bmop object over a space of dimension length(MARGIN) the result of integrating over the -MARGIN other dimensions.

Examples

```
data<-data.frame(rnorm(100),rnorm(100))
bmop2d<-bmop_fit(data)
bmop1d<-bmop_fit(data[,1])
bmop1dmargin<-marginalize.bmop(bmop2d,MARGIN=1)
comparison_plot(list(bmop1d,bmop1dmargin),true=dnorm,
names.bmop=c("direct est.", "marginalized"))
```

mean.bmop	<i>Mean value for a bmop density</i>
-----------	--------------------------------------

Description

Mean value for a bmop density

Usage

```
## S3 method for class 'bmop'
mean(x, ...)
```

Arguments

x a bmop object
 ... for compatibility with `mean`

Value

numeric value, the mean of the bmop density

Examples

```
bmop<-bmop_fit(rnorm(100))
mean(bmop)
```

new_bmop

New bmop object

Description

Constructor of bmop object.

Usage

```
new_bmop(knots, order, ctrpoints = NULL, nk = FALSE)
```

Arguments

knots list of numeric vector, knots of the B-spline basis
 order vector order of the B-spline for each variable, values will be recycled eventually
 ctrpoints array of control points
 nk logical, parameter used internally see details on how to use it

Details

The function create an object of class bmop, that is, a list with components

- order vector of orders
- knots list, every element of the list is the vector of knots in one dimension, knots are ordered but can be repeted.
- ctrpoints An array of control points

If nk==FALSE (default method) the knots provided are sorted and the unique function is applied. Then the appropriate knots are computed, repeating the first and last knots many times as the order in the considered dimension, with this choice the bmop object represent a B-spline function regular up to the order-1 derivate in every dimension. If nk==TRUE the bmop object will be created without any check on the knots, use carefully this option as it could result in a wrong defined bmop object.

Value

An object of class bmop, a list with elements order, knots, ctrpoints,

normalize.bmop	<i>Normalize a bmop</i>
----------------	-------------------------

Description

Normalize a bmop object, makes it integrate to one.

Usage

```
normalize.bmop(object, ...)
```

Arguments

object	a bmop object
...	additional parameters

Value

a bmop object proportional to object, but such that integrates to one.

plot.bmop	<i>Plot of bmop object</i>
-----------	----------------------------

Description

Plot of bmop object

Usage

```
## S3 method for class 'bmop'
plot(x, N = 1000, type = "l", contour = TRUE,
     persp = FALSE, file = NULL, MIN = 0, ...)
```

Arguments

x	bmop object
N	positive integer
type	graphical parameter, see par
contour	graphical parameter, see par
persp	logical
file	optional file name, where to save a pdf copy of the plot
MIN	parameter of evaluate.bmop
...	graphical parameters as col, main, ..., see plot or filled.contour

Details

For the 2-d persp (persp==TRUE) plot the rgl package is needed. In order to obtain a nicer result in the contour plot, try to supply a col or color.palette parameter, like in the example below (RColorBrewer package is needed).

Value

plot of 1-d or 2-d (contour and perspective) bmop, an empty plot and a warning message are returned if the bmop object has more than 2 dimensions.

See Also

[points.bmop](#)

Examples

```
## Not run:
bmop<-bmop_fit(data.frame(rnorm(100),rnorm(100)))
colFun<-
grDevices::colorRampPalette(RColorBrewer::brewer.pal(9,name="YlGnBu"))
plot(bmop,color.palette=colFun)

## End(Not run)
```

points.bmop

Plot points from bmop

Description

Plot points from bmop

Usage

```
## S3 method for class 'bmop'
points(x, N = 100, ...)
```

Arguments

x	bmop object
N	number of points to plot
...	graphical parameters see par

Details

As [points](#), this functions provide a way to add the plot of a bmop to an existing plot.

See Also[plot.bmop](#).**Examples**

```
data<-rnorm(200)
bmopE<-bmop_fit(data)
bmopS<-search_bmop(data)
plot(bmopE)
points(bmopS,type="l",col="red")
```

print.bmop	<i>Print bmop objects</i>
------------	---------------------------

Description

Print bmop objects

Usage

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

Arguments

x	bmop object
...	see print

put_evidence.bmop	<i>Put evidence on a conditional bmop</i>
-------------------	-------------------------------------------

Description

Instantiate evidence on a bmop conditional density.

Usage

```
put_evidence.bmop(object, evidence, evd.pos = NULL, MIN = 0,
  normalize = FALSE, ...)
```

Arguments

object	a bmop object
evidence	the value of evidence
evd.pos	the position of evidence
MIN	the MIN value as in bmopPar
normalize	logical, if TRUE the final bmop object will be normalized (usually it is not needed since this function is applied to conditional densities)
...	additional parameters

Value

a bmop object, the result of imposing some evidence

Rbmop	<i>Rbmop: A package for handling and estimating densities and conditional densities with B-spline.</i>
-------	--------------------------------------------------------------------------------------------------------

Description

The main functions of Rbmop package are the following:

- `bmop_fit` for estimation of density or conditional density.
- `bmopPar` function for changing the parameter of the package.
- `compare.bmop` and `comparison_plot` comparison functions.
- Moreover the package provide base functions as `plot`, `print` and conversion methods `as.function.bmop`.

Author(s)

Gherardo Varando <gherardo.varando@gmail.com>, Concha Bielza and Pedro Larranaga

Examples

```
data(trees)
bmop<-bmop_fit(data=trees$Height)
summary(bmop)
#####
###Envelope for exponential density
envelope_plot(n=50,N=50,rtrue=rexp,dtrue=dexp)
#####
##
X<-rnorm(100)
Y<-rnorm(100,mean=X)
data<-data.frame(Y,X)
condbmop<-bmop_fit(data=data,conditional=TRUE)
plot(condbmop,N=20,persp=TRUE)
```

```
#####
## Trimodal-not differentiable density estimation via histogram
f<-function(x){return( (dnorm(x,mean=6)+dnorm(x,mean=-3)+dgamma(x,shape=2))/3 )}
D<-c(rnorm(10000,mean=-3),rgamma(10000,shape=2),rnorm(10000,mean=+6))
bmopPar(mle=TRUE)
bmop<-bmop_fit(hist(D,nclass.FD))
comparison_plot(bmop,f,ylim=NULL)

#####
## Using the examples functions

Ex<-ex_bmop_gaussian2Mixture()
points(density(Ex$data),type="l",col="blue",lwd=3)

####

Ex<-ex_bmop_gaussianBetaGamma(N = 10000,m1 = 8)
points(density(Ex$data),type="l",col="blue",lwd=3)
compare_bmop(Ex$bmop,Ex$true,c("MSE","MAE","MAX"))
```

search_bmop

*Greedy penalized log-likelihood search***Description**

Aproximation of a density $f(x_1, \dots, x_n)$ or conditional density

Usage

```
search_bmop(data, conditional = F, k = Rbmop::bmopPar()$k,
  corrected = FALSE, knotsMethod = Rbmop::bmopPar()$knotsMethod, ...)
```

Arguments

data	data.frame, matrix or vector, the variables must be in the right order (the columns of data)
conditional	logical
k	positive number or "BIC"
corrected	logical
knotsMethod	the method to use in knots generation
...	additional parameters

Value

A bmop object, the aproximations of f .

Examples

```
data<-rnorm(100)
bmopS<-search_bmop(data=data)
plot(bmopS)
```

summary.bmop	Summary of a bmop object
--------------	--------------------------

Description

Summary of a bmop object

Usage

```
## S3 method for class 'bmop'
summary(object, ...)
```

Arguments

- object bmop object
- ... compatibility with [summary](#)

Examples

```
data(trees)
bmop<-bmop_fit(data=trees$Height)
summary(bmop)
```

upper.bmop	Upper Limit of bmop
------------	---------------------

Description

Upper Limit of bmop

Usage

```
upper.bmop(object)
```

Arguments

- object a bmop object

Value

numeric value

Index

add_knots.bmop, 2
AIC, 3, 16
AIC.bmop, 3
as.bins, 3
as.function, 4
as.function.bmop, 4, 22

BIC, 4
BIC.bmop, 4
bmop_fit, 5, 6, 12, 22
bmop_fit.bins, 7
bmop_fit.data.frame, 7
bmop_fit.default, 8
bmop_fit.histogram, 9
bmopPar, 5, 22

clean.bmop, 9
compare.bmop, 10, 22
comparison_plot, 10, 22

dim.bmop, 11
dnorm, 12

envelope_plot, 12
evaluate.bmop, 12, 19
ex_bmop_gaussian2Mixture
 (Examples_bmop), 13
ex_bmop_gaussian3Mixture
 (Examples_bmop), 13
ex_bmop_gaussianBetaGamma
 (Examples_bmop), 13
Examples_bmop, 13

filled.contour, 19

generate_knots, 5, 14

integrate.bmop, 15
is.bmop, 15

logLik, 5

logLik.bmop, 16
lower.bmop, 16

marginalize.bmop, 17
mean, 18
mean.bmop, 17

new_bmop, 18
normalize.bmop, 19

par, 11, 19, 20
plot, 19
plot.bmop, 19, 21
points, 20
points.bmop, 20, 20
print, 21
print.bmop, 21
put_evidence.bmop, 21

Rbmop, 22
Rbmop-package (Rbmop), 22
rnorm, 12

search_bmop, 5, 23
summary, 24
summary.bmop, 24

upper.bmop, 24