Package 'ppgmmga'

November 17, 2023

Version 1.3

Date 2023-11-17

Title Projection Pursuit Based on Gaussian Mixtures and Evolutionary Algorithms

Description Projection Pursuit (PP) algorithm for dimension reduction based on Gaussian Mixture Models (GMMs) for density estimation using Genetic Algorithms (GAs) to maximise an approximated negentropy index. For more details see Scrucca and Serafini (2019) <doi:10.1080/10618600.2019.1598871>.

Depends R (>= 3.4)

Imports Rcpp (>= 1.0.0), mclust (>= 5.4), GA (>= 3.1), ggplot2 (>= 2.2.1), cli, crayon, utils, stats

LinkingTo Rcpp, RcppArmadillo (>= 0.7)

Suggests knitr (>= 1.8), rmarkdown (>= 2.0)

VignetteBuilder knitr

License GPL (>= 2)

Repository CRAN

URL https://github.com/luca-scr/ppgmmga

BugReports https://github.com/luca-scr/ppgmmga/issues

ByteCompile true

NeedsCompilation yes

Encoding UTF-8

RoxygenNote 6.1.1

Author Alessio Serafini [aut] (<https://orcid.org/0000-0002-8579-5695>), Luca Scrucca [aut, cre] (<https://orcid.org/0000-0003-3826-0484>)

Maintainer Luca Scrucca <luca.scrucca@unipg.it>

Date/Publication 2023-11-17 22:40:08 UTC

R topics documented:

`ppgmmga-package` *Projection pursuit based on Gaussian mixtures and evolutionary algorithms for data visualisation*

Description

An R package implementing a Projection Pursuit (PP) algorithm based on finite Gaussian Mixture Models (GMMs) for density estimation using Genetic Algorithms (GAs) to maximise an approximated negentropy index. The **ppgmmga** algorithm provides a method to visualise high-dimensional data in a lower-dimensional space.

Details

An introduction to **ppgmmga** package is provided in the accompaying vignette A quick tour of ppgmmga.

Author(s)

Serafini A. <srf.alessio@gmail.com>
Scrucca L. <luca.scrucca@unipg.it>

References

Scrucca, L. and Serafini, A. (2019) Projection pursuit based on Gaussian mixtures and evolutionary algorithms. *Journal of Computational and Graphical Statistics*, 28:4, 847–860. DOI: 10.1080/10618600.2019.1598871

See Also

`ppgmmga`, `plot.ppgmmga`, `ppgmmga-class`, `ppgmmga.options`, `summary.ppgmmga`

nclass.numpy *Compute the Number of Classes for a Histogram*

Description

Compute the number of classes for a histogram as the maximum of the "Sturges" and "FD" (Freedman Diaconis) estimators as in numpy library for Python.

Usage

```
nclass.numpy(x, ...)
```

Arguments

x	A vector of values.
...	Further arguments passed to or from other methods.

Author(s)

Scrucca L. <luca.scrucca@unipg.it>

See Also

nclass.Sturges, nclass.FD

Examples

```
## Not run:
library(ggplot2)
x <- rnorm(100)
ggplot() + geom_histogram(aes(x), col = "grey92", bins = nclass.numpy(x))
x <- rnorm(1000)
ggplot() + geom_histogram(aes(x), col = "grey92", bins = nclass.numpy(x))
n = c(50, seq(100,1000,by=100))
brks = rep(NA, length(n))
for(i in seq(n)) brks[i] = nclass.numpy(rnorm(n[i]))
ggplot() + geom_point(aes(x = n, y = brks))

## End(Not run)
```

`plot.ppgmmga` *Plots the data onto the projection subspace estimated by the ppgmmga algorithm*

Description

Plot method for objects of class `'ppgmmga'`.

Usage

```
## S3 method for class 'ppgmmga'
plot(x,
     class = NULL,
     dim = seq(x$d),
     drawAxis = TRUE,
     bins = nclass.numpy,
     ...)
```

Arguments

`x`	An object of class `'ppgmmga'` obtained from a call to `ppgmmga` function.
`class`	A numeric or character vector indicating the classification of the observations/cases to be plotted.
`dim`	A numeric vector indicating the dimensions to use for plotting. By default, all the dimensions of the projection subspace (i.e. `x$d`) are used. Subsets of all the available dimensions can also be provided (see example below.) The resulting graph depends on the dimension: in the 1D case a histogram is provided, a scatterplot in the 2D case, a scatterplot matrix in higher dimensions.
`drawAxis`	A logical value specifying whether or not the axes should be included in the 2D scatterplot. By default is to `TRUE`.
`bins`	An R function to be used for computing the number of classes for the histogram. By default `nclass.Sturges` is used. Users may provide a different function. This argument only applies to 1D graphs.
`...`	further arguments.

Details

Plots the cloud of points onto a subspace after appling the Projection Pursuit algorithm based on Gaussian mixtures and Genetic algorithm implemented in ppgmmga function.

Value

Returns a object of class `ggplot`.

Author(s)

Serafini A. <srf.alessio@gmail.com>
Scrucca L. <luca.scrucca@unipg.it>

References

Scrucca, L. and Serafini, A. (2019) Projection pursuit based on Gaussian mixtures and evolutionary algorithms. *Journal of Computational and Graphical Statistics*, 28:4, 847–860. DOI: 10.1080/10618600.2019.1598871

See Also

ppgmmga

Examples

```
## Not run:
data(iris)
X <- iris[,-5]
Class <- iris$Species

# 1D
pp1 <- ppgmmga(data = X, d = 1, approx = "UT")
summary(pp1, check = TRUE)
plot(pp1)
plot(pp1, Class)

# 2D
pp2 <- ppgmmga(data = X, d = 2, approx = "UT")
summary(pp2, check = TRUE)
plot(pp2)
plot(pp2, Class)

# 3D
pp3 <- ppgmmga(data = X, d = 3)
summary(pp3, check = TRUE)
plot(pp3)
plot(pp3, Class)
plot(pp3, Class, dim = c(1,3))
plot(pp3, Class, dim = c(2,3))

## End(Not run)
```

---

ppgmmga *Projection pursuit based on Gaussian mixtures and evolutionary algorithms for data visualisation*

Description

A Projection Pursuit (PP) method for dimension reduction seeking "interesting" data structures in low-dimensional projections. A negentropy index is computed from the density estimated using Gaussian Mixture Models (GMMs). Then, the PP index is maximised by Genetic Algorithms (GAs) to find the optimal projection basis.

Usage

```
ppgmmga(data,
        d,
        approx = c("UT", "VAR", "SOTE", "none"),
        center = TRUE,
        scale = TRUE,
        GMM = NULL,
        gatype = c("ga", "gaisl"),
        options = ppgmmga.options(),
        seed = NULL,
        verbose = interactive(), ...)
```

Arguments

`data`	A nxp matrix containing the data with rows corresponding to observations and columns corresponding to variables.
`d`	An integer specifying the dimension of the subspace onto which the data are projected and visualised.
`approx`	A string specifying the type of computation to perform to obtain the negentropy for GMMs. Possible values are: `"UT"` = Unscented Trasformation approximation (default); `"VAR"` = VARiational approximation; `"SOTE"` = Second Order Taylor Expansion approximation; `"none"` = exact calculation (no approximation, experimental).
`center`	A logical value indicating whether or not the data are centred. By default is set to `TRUE`.
`scale`	A logical value indicating whether or not the data are scaled. By default is set to `TRUE`.
`GMM`	An object of class `'densityMclust'` specifying a Gaussian mixture density estimate as returned by `densityMclust`.
`gatype`	A string specifying the type of genetic algoritm to be used to maximised the negentropy. Possible values are: `"ga"` = simple genetic algorithm (`ga`); `"gaisl"` = island genetic algorithm (`gaisl`).
`options`	A list of options containing all the important arguments to pass to `densityMclust` function of the **mclust** package, and to `ga` function of the **GA** package. See

	ppgmmga.options for the available options. Note that by setting the options argument does not change the global options provided by ppgmmga.options, but only the options for a single call to ppgmmga.
seed	An integer value with the random number generator state. It may be used to replicate the results of ppgmmga algorithm.
verbose	A logical value controlling if the evolution of GA search is shown. By default is TRUE reporting the number of iteration, average and best fitness value.
...	Further arguments passed to or from other methods.

Details

Projection pursuit (PP) is a features extraction method for analysing high-dimensional data with low-dimension projections by maximising a projection index to find out the best orthogonal projections. A general PP procedure can be summarised in few steps: the data may be transformed, the PP index is chosen and the subspace dimension is fixed. Then, the PP index is optimised.

For clusters visualisation the negentropy index is considerd. Since such index requires an estimation of the underling data density, Gaussian mixture models (GMMs) are used to approximate such density.

Genetic Algorithms are then employed to maximise the negentropy with respect to the basis of the projection subspace.

Value

Returns an object of class 'ppgmmga'. See ppgmmga-class for a description of the object.

Author(s)

Serafini A. <srf.alessio@gmail.com>
Scrucca L. <luca.scrucca@unipg.it>

References

Scrucca, L. and Serafini, A. (2019) Projection pursuit based on Gaussian mixtures and evolutionary algorithms. *Journal of Computational and Graphical Statistics*, 28:4, 847–860. DOI: 10.1080/10618600.2019.1598871

See Also

summary.ppgmmga, plot.ppgmmga, ppgmmga-class

Examples

```
## Not run:
data(iris)
X <- iris[,-5]
Class <- iris$Species

# 1-dimensional PPGMMGA
```

```
PP1D <- ppgmmga(data = X, d = 1)
summary(PP1D)
plot(PP1D, bins = 11)
plot(PP1D, bins = 11, Class)

# 2-dimensional PPGMMGA

PP2D <- ppgmmga(data = X, d = 2)
summary(PP2D)
plot(PP2D)
plot(PP2D, Class)

## Unscented Transformation approximation

PP2D_1 <- ppgmmga(data = X, d = 2, approx = "UT")
summary(PP2D_1)
plot(PP2D_1, Class)

## VARiational approximation

PP2D_2 <- ppgmmga(data = X, d = 2, approx = "VAR")
summary(PP2D_2)
plot(PP2D_2, Class)

## Second Order Taylor Expansion approximation

PP2D_3 <- ppgmmga(data = X, d = 2, approx = "SOTE")
summary(PP2D_3)
plot(PP2D_3, Class)

# 3-dimensional PPGMMGA

PP3D <- ppgmmga(data = X, d = 3,)
summary(PP3D)
plot(PP3D, Class)

# A rotating 3D plot can be obtained using:
# if(!require("msir")) install.packages("msir")
# msir::spinplot(PP3D$Z, markby = Class,
#                col.points = ppgmmga.options("classPlotColors")[1:3])

## End(Not run)
```

`ppgmmga-class` *Class* `'ppgmmga'`

Description

An S3 class object for ppgmmga algorithm

Objects from the class

Object can be created by calls to the ppgmmga function.

Values

data The input data matrix.

d The dimension of the projection subspace.

approx The type of approximation used for computing negentropy.

GMM An object of class 'densityMclust' containing the Gaussian mixture density estimation. See densityMclust for details.

GA An object of class 'ga' containing the Genetic Algorithm search. See ga for details.

Negentropy The value of maximised negentropy.

basis The matrix basis of the projection subspace.

Z The matrix of projected data.

Author(s)

Serafini A. <srf.alessio@gmail.com>
Scrucca L. <luca.scrucca@unipg.it>

See Also

ppgmmga, plot.ppgmmga, summary.ppgmmga

---

ppgmmga.options *Default values for* **ppgmmga** *package*

Description

Set or retrieve default values to be used by the **ppgmmga** package.

Usage

```
ppgmmga.options(...)
```

Arguments

...	A single character vector, or a named list with components. In the one argument case, the form name = value can be used to change a single option. In the multiple arguments case, the form list(name1 = value1, name2 = value2) can be used to change several arguments. If no arguments are provided, then the function returns all the current options. For the available options see the Details section below.

Details

This function can be used to set or retrieve the values to be used by the **ppgmmga** package.

The function globally sets the arguments for the current session of R. The default options are restored with a new R session. To temporarily change the options for a single call to ppgmmga function, look at `options` argument in `ppgmmga`.

Available options are:

`modelNames` A string specifying the GMM to fit. See `mclustModelNames` for the available models.

`G` An integer value or a vector of integer values specifying the number of mixture components. If more than a single value is provided, the best model is selected using the BIC criterion. By default `G = 1:9`.

`initMclust` A string specifying the type of initialisation to be used for the EM algorithm. See `mclust.options` for more details.

`popSize` The GA population size. By default `popSize = 100`.

`pcrossover` The probability of crossover. By default `pcrossover = 0.8`.

`pmutation` The probability of mutation. By default `pmutation = 0.1`.

`maxiter` An integer value specifying the maximum number of iterations before stopping the GA. By default `maxiter = 1000`.

`run` An integer value indicating the number of generations without improvment in the best value of fitness fuction. `run = 100`.

`selection` An R function performing the selection genetic operator. See `ga_Selection` for details. By default `selection = gareal_lsSelection`.

`crossover` An R function performing the crossover genetic operator. See `ga_Crossover` for details. By default `crossover = gareal_laCrossover`.

`mutation` An R function performing the mutation genetic operator. See `ga_Mutation` for details. By default `mutation = gareal_raMutation`.

`parallel` A logical value specifying whether or not GA should be run in parallel. By default `parallel = FALSE`.

`numIslands` An integer value specifying the number of islands to be used in the Island Genetic Algorithm. By default `numIslands = 4`.

`migrationRate` A value specifying the fraction of migration between islands. By default `migrationRate = 0.1`.

`migrationInterval` An integer values specifying the number of generations to run before each migration. By default `migrationInterval = 10`.

`optim` A logical value specifying whether or not a local search should be performed. By default `optim = TRUE`.

`optimPoptim` A value specifying the probability a local search is performed at each GA generation. By default `optimPoptim = 0.05`.

`optimPressel` A value in $[0, 1]$ specifying the pressure selection. Values close to 1 tend to assign higher selection probabilities to solutions with higher fitness, whereas values close to 0 tend to assign equal selection probability to any solution. By default `optimPressel = 0.5`.

`optimMethod` A string specifying the general-purpose optimisation method to be used for local search. See `optim` for the available algorithms. By default `optimMethod = "L-BFGS-B"`.

`optimMaxit` An integer value specifying the number of iterations for the local search algorithm. By default `optimMaxit = 100`.

`orth` A string specifying the method employed to orthogonalise the matrix basis. Available methods are the QR decomposition `"QR"`, and the Singular Value Decomposition `"SVD"`. By default `orth = "QR"`.

`classPlotSymbols` A vector whose entries are either integers corresponding to graphics symbols or single characters for indicating classifications when plotting data. Classes are assigned symbols in the given order.

`classPlotColors` A vector whose entries correspond to colors for indicating classifications when plotting data. Classes are assigned colors in the given order.

For more details about options related to Gaussian mixture modelling see `densityMclust`, and for those related to genetic algorithms see `ga` and `gaisl`.

Author(s)

Serafini A. <srf.alessio@gmail.com>
Scrucca L. <luca.scrucca@unipg.it>

References

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016) mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. *The R journal*, 8(1), 205-233. https://journal.r-project.org/archive/2016-1/scrucca-fop-murphy-etal.pdf

Scrucca, L. (2013) GA: A Package for Genetic Algorithms in R. *Journal of Statistical Software*, 53(4), 1-37. http://www.jstatsoft.org/v53/i04/

Scrucca, L. (2017) On some extensions to GA package: hybrid optimisation, parallelisation and islands evolution. *The R Journal*, 9/1, 187-206. https://journal.r-project.org/archive/2017/RJ-2017-008

Scrucca, L. and Serafini, A. (2019) Projection pursuit based on Gaussian mixtures and evolutionary algorithms. *Journal of Computational and Graphical Statistics*, 28:4, 847–860. DOI: 10.1080/10618600.2019.1598871

See Also

`ppgmmga`

Examples

```
## Not run:
ppgmmga.options()

# Print a single option
ppgmmga.options("popSize")

# Change (globally) an option
```

```
ppgmmga.options("popSize" = 10)
ppgmmga.options("popSize")

## End(Not run)
```

`summary.ppgmmga` *Summary for projection pursuit based on Gaussian mixtures and evolutionary algorithms for data visualisation*

Description

Summary method for objects of class `'ppgmmga'`.

Usage

```
## S3 method for class 'ppgmmga'
summary(object, check = (object$approx != "none"), ...)

## S3 method for class 'summary.ppgmmga'
print(x, digits = getOption("digits"), ...)
```

Arguments

`object`	An object of class `'ppgmmga'` as returned by `ppgmmga`.
`check`	A logical value specifying whether or not a Monte Carlo negentropy approximation check should be performed. By default is `FALSE` for exact negentropy calculation and `TRUE` for approximated negentropy.
`x`	An object of class `summary.ppgmmga`.
`digits`	The number of significant digits.
`...`	Further arguments passed to or from other methods.

Value

The summary function returns an object of class `summary.ppgmmga` which can be printed by the corresponding print method. A list with the information from the `ppgmmga` algorithm is returned.

If the optional argument `check = TRUE` then the value of negentropy is compared to the Monte Carlo negentropy calculated for the same optimal projection basis selected by the algorithm. By default, it allows to check if the value returned by the employed approximation is closed to the Monte Carlo approximation of to the "true" negentropy. The ratio between the approximated value returned by the algorithm and the value computed with Monte Carlo is called Relative Accuracy. Such value should be close to 1 for a good approximation.

Author(s)

Serafini A. <srf.alessio@gmail.com>
Scrucca L. <luca.scrucca@unipg.it>

See Also

ppgmmga

Index