Package ‘ppgmmga’

October 15, 2018

Version 1.0.1
Date 2018-10-15
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.
Depends R (>= 3.4)
Imports Rcpp (>= 0.12.5), mclust (>= 5.0), GA (>= 3.1), ggplot2 (>= 2.2.1), ggthemes (>= 3.4.0), cli, crayon
LinkingTo Rcpp, RcppArmadillo (>= 0.7)
Suggests knitr (>= 1.8)
License GPL (>= 2)
URL https://github.com/luca-scr/ppgmmga
BugReports https://github.com/luca-scr/ppgmmga/issues
VignetteBuilder knitr
ByteCompile true
Repository CRAN
NeedsCompilation yes
LazyLoad yes
Encoding UTF-8
Author Alessio Serafini [aut, cre] (https://orcid.org/0000-0002-8579-5695), Luca Scrucca [aut] (https://orcid.org/0000-0003-3826-0484)
Maintainer Alessio Serafini <srf.alessio@gmail.com>
Date/Publication 2018-10-15 18:20:03 UTC
R topics documented:

ppgmmga-package .................................................. 2
plot.ppgmmga .................................................... 3
ppgmmga ............................................................. 4
ppgmmga-class ..................................................... 7
ppgmmga.options .................................................. 8
summary.ppgmmga ................................................ 10

Index 12

---

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 accompanying vignette A quick tour of `ppgmmga`.

Author(s)

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

References


See Also

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

Plot method for objects of class 'ppgmmga'.

Usage

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

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; see `nclass.Sturges` for more details. 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 applying the Projection Pursuit algorithm based on Gaussian mixtures and Genetic algorithm implemented in `ppgmmga` function.

Value

Returns a object of class `ggplot`.
ppgmmga

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

Author(s)

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

References


See Also

ppgmmga

Examples

```r
# 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)
```
Description

A Projection Pursuit (PP) method for dimension reduction seeking "interesting" data structures in low-dimensional projections. An approximated 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"), center = TRUE, scale = TRUE, gmm = NULL, gatype = c("ga", "gaisl"), options = ppgmmga.options(), seed = NULL, verbose = interactive(), ...)

Arguments

data A n x p 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 approximation to use for computing the negentropy for GMMs. Possible values are:

"UT" for Unscented Trasformation approximation.
"VAR" for VARIational approximation.
"SOTE" for Second Order Taylor Expansion approximation.

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 algorithm to be used to maximised the negentropy. Possible values are:

"ga" for simple genetic algorithm (ga).
"gaisl" for 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 has been considered. Since such index requires an estimation of the underlying data density, Gaussian mixture models (GMMs) has been used to approximate such density. GMMs do not have a closed formula for the Negentropy and different closed formula approximations have been implemented. Genetic Algorithms have been employed to maximised the approximated negentropy respect to the system of basis in the desired 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


See Also

summary.ppgmmga, plot.ppgmmga, ppgmmga-class

Examples

data(iris)
X <- iris[, -5]
Class <- iris$Species

# Unscented Transformation approximation (default)
pp1 <- ppgmmga(data = X, d = 2, approx = "UT", seed = 123)
summary(pp1, check = TRUE)
plot(ppgmmga(data = X, d = 2, approx = "VAR"))
summary(ppgmmga(data = X, d = 2, approx = "SOTE"))
plot(ppgmmga(data = X, d = 2, approx = "SOTE"))

## 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**

```r
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:

- **modelName** 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 improvement in the best value of fitness function. `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 \texttt{ga\_Crossover} for details. By default \texttt{crossover = gareal\_laCrossover}.

mutation An R function performing the mutation genetic operator. See \texttt{ga\_Mutation} for details. By default \texttt{mutation = gareal\_raMutation}.

parallel A logical value specifying whether or not GA should be run in parallel. By default \texttt{parallel = FALSE}.

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

migrationRate A value specifying the fraction of migration between islands. By default \texttt{migrationRate = 0.1}.

migrationInterval An integer values specifying the number of generations to run before each migration. By default \texttt{migrationInterval = 10}.

optim A logical value specifying whether or not a local search should be performed. By default \texttt{optim = TRUE}.

optimPoptim A value specifying the probability a local search is performed at each GA generation. By default \texttt{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 \texttt{optimPressel = 0.5}.

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

optimMaxit An integer value specifying the number of iterations for the local search algorithm. By default \texttt{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 \texttt{orth = "QR"}.

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

Author(s)

Serafini A. <srfalessio@gmail.com>

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

References


See Also

ppgmmga

Examples

```r
## Not run:
papgmmga.options()

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

# Change (globally) an option
papgmmga.options("popSize" = 10)
papgmmga.options("popSize")

## End(Not run)
```

summary.papgmmga

Summary method for objects of class 'ppgmmga'.

Usage

```r
## S3 method for class 'ppgmmga'
summary(object, check = FALSE, ...)

## S3 method for class 'summary.papgmmga'
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 set to FALSE.
- **x**: An object of class `summary.papgmmga`.
- **digits**: The number of significant digits.
- **...**: Further arguments passed to or from other methods.
Value

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

If the optional argument `check = TRUE` then the value of approximated negentropy is compared to the Monte Carlo negentropy calculated for the same optimal projection basis selected by the algorithm. Since the Monte Carlo approximation is the only one guarantee to converge to the "true" negentropy, it allows to check if the value returned by the employed approximation is closed 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

`ppgmga`
Index

*Topic `plot.ppgmmga`, 3
*Topic `plot.ppgmmga`, 3
*Topic `classes`
  `ppgmmga-class`, 7
*Topic `options`
  `ppgmmga.options`, 8

densityMclust, 5, 7, 9

ga, 5, 7, 9
`ga_Crossover`, 9
`ga_Mutation`, 9
`ga_Selection`, 8
gaisl, 5, 9
ggplot, 3

mclust.options, 8
mclustModelNames, 8

nclass.Sturges, 3

optim, 9

plot.ppgmmga, 2, 3, 6, 7
`ppgmmga`, 2–4, 4, 7, 8, 10, 11
`ppgmmga-class`, 7
`ppgmmga-package`, 2
`ppgmmga.options`, 2, 6, 8
print.ppgmmga(ppgmmga), 4
print.summary.ppgmmga
  (summary.ppgmmga), 10

summary.ppgmmga, 2, 6, 7, 10