Package 'Pursuit'

June 21, 2024

Type Package

Title Projection Pursuit

Version 1.0.5

Date 2024-06-21

Imports graphics, grDevices, MASS, stats

Description Projection pursuit (PP) with 17 methods and grand tour with 3 methods. Being that projection pursuit searches for low-dimensional linear projections in high-dimensional data structures, while grand tour is a technique used to explore multivariate statistical data through animation.

License GPL-3

Encoding UTF-8

NeedsCompilation yes

Author Paulo Cesar Ossani [aut, cre] (<`https://orcid.org/0000-0002-6617-8085`>),
Marcelo Angelo Cirillo [aut] (<`https://orcid.org/0000-0002-2647-439X`>)

Maintainer Paulo Cesar Ossani <`ossanipc@hotmail.com`>

Repository CRAN

Date/Publication 2024-06-21 16:00:02 UTC

Contents

GrandTour *Animation technique Grand Tour.*

Description

Performs the exploration of the data through the technique of animation Grand Tour.

Usage

```
GrandTour(data, method = "Interpolation", title = NA, xlabel = NA,
          ylabel = NA, size = 1.1, grid = TRUE, color = TRUE, linlab = NA,
          class = NA, classcolor = NA, posleg = 2, boxleg = TRUE,
          axesvar = TRUE, axes = TRUE, numrot = 200, choicerot = NA,
          savptc = FALSE, width = 3236, height = 2000, res = 300)
```

Arguments

data Numerical data set.

method Method used for rotations:
"Interpolation" - Interpolation method (default),
"Torus" - Torus method,
"Pseudo" - Pseudo Grand Tour method.

title Titles of the graphics, if not set, assumes the default text.

xlabel Names the X axis, if not set, assumes the default text.

ylabel Names the Y axis, if not set, assumes the default text.

size Size of the points in the graphs.

grid Put grid on graphs (default = TRUE).

color Colored graphics (default = TRUE).

linlab Vector with the labels for the observations.

class Vector with names of data classes.

classcolor Vector with the colors of the classes.

posleg 0 with no caption,
1 for caption in the left upper corner,
2 for caption in the right upper corner (default),
3 for caption in the right lower corner,
4 for caption in the left lower corner.

boxleg Puts the frame in the caption (default = TRUE).

axesvar Puts axes of rotation of the variables (default = TRUE).

axes Plots the X and Y axes (default = TRUE).

numrot Number of rotations (default = 200). If method = "Interpolation", numrot represents the angle of rotation.

`choicerot` Choose specific rotation and display on the screen, or save the image if savptc = TRUE.

`savptc` Saves graphics images to files (default = FALSE).

`width` Graphics images width when savptc = TRUE (defaul = 3236).

`height` Graphics images height when savptc = TRUE (default = 2000).

`res` Nominal resolution in ppi of the graphics images when savptc = TRUE (default = 300).

Value

Graphs with rotations.

`proj.data` Projected data.

`vector.opt` Vector projection.

`method` method used on Grand Tour.

Author(s)

Paulo Cesar Ossani

Marcelo Angelo Cirillo

References

ASIMOV, D. The Grand Tour: A Tool for Viewing Multidimensional data. *SIAM Journal of Scientific and Statistical Computing*, 6(1), 128-143, 1985.

ASIMOV, D.; BUJA, A. The grand tour via geodesic interpolation of 2-frames. in Visual data Exploration and Analysis. *Symposium on Electronic Imaging Science and Technology*, IS&T/SPIE. 1994.

BUJA, A. ; ASIMOV, D. Grand tour methods: An outline. *Computer Science and Statistics*, 17:63-67. 1986.

BUJA, A.; COOK, D.; ASIMOV, D.; HURLEY, C. Computational methods for High-Dimensional Rotations in data Visualization, in C. R. Rao, E. J. Wegman & J. L. Solka, eds, *"Handbook of Statistics: data Mining and Visualization"*, Elsevier/North Holland, http://www.elsevier.com, pp. 391-413. 2005.

HURLEY, C.; BUJA, A. Analyzing high-dimensional data with motion graphics, *SIAM Journal of Scientific and Statistical Computing*, 11 (6), 1193-1211. 1990.

MARTINEZ, W. L., MARTINEZ, A. R., SOLKA, J.; *Exploratory data Analysis with MATLAB*, 2th. ed. New York: Chapman & Hall/CRC, 2010. 499 p.

YOUNG, F. W.; RHEINGANS P. Visualizing structure in high-dimensional multivariate data, *IBM Journal of Research and Development*, 35:97-107, 1991.

YOUNG, F. W.; FALDOWSKI R. A.; McFARLANE M. M. *Multivariate statistical visualization, in Handbook of Statistics*, Vol 9, C. R. Rao (ed.), The Netherlands: Elsevier Science Publishers, 959-998, 1993.

Examples

```
data(iris) # database

res <- GrandTour(iris[,1:4], method = "Torus", title = NA, xlabel = NA, ylabel = NA,
                 color = TRUE, linlab = NA, class = NA, posleg = 2, boxleg = TRUE,
                 axesvar = TRUE, axes = FALSE, numrot = 10, choicerot = NA,
                 savptc = FALSE, width = 3236, height = 2000, res = 300)

print("Projected data:"); res$proj.data
print("Projection vectors:"); res$vector.opt
print("Grand Tour projection method:"); res$method


res <- GrandTour(iris[,1:4], method = "Interpolation", title = NA, xlabel = NA, ylabel = NA,
                 color = TRUE, linlab = NA, posleg = 2, boxleg = FALSE, axesvar = FALSE,
                 axes = FALSE, numrot = 10, choicerot = NA, class = iris[,5],
                 classcolor = c("goldenrod3","gray53","red"),savptc = FALSE,
                 width = 3236, height = 2000, res = 300)

print("Projected data:"); res$proj.data
print("Projection vectors:"); res$vector.opt
print("Grand Tour projection method:"); res$method
```

LocLab *Function for better position of the labels in the graphs.*

Description

Function for better position of the labels in the graphs.

Usage

```
LocLab(x, y = NULL, labels = seq(along = x), cex = 1,
       method = c("SANN", "GA"), allowSmallOverlap = FALSE,
       trace = FALSE, shadotext = FALSE,
       doPlot = TRUE, ...)
```

Arguments

x	Coordinate x
y	Coordinate y
labels	The labels
cex	cex
method	Not used
allowSmallOverlap	Boolean

trace Boolean

shadotext Boolean

doPlot Boolean

... Other arguments passed to or from other methods

Value

See the text of the function.

Plot.PP *Graphics of the Projection Pursuit (PP).*

Description

Graphics of the Projection Pursuit (PP).

Usage

```
Plot.PP(PP, titles = NA, xlabel = NA, ylabel = NA, posleg = 2, boxleg = TRUE,
        size = 1.1, grid = TRUE, color = TRUE, classcolor = NA, linlab = NA,
       axesvar = TRUE, axes = TRUE, savptc = FALSE, width = 3236, height = 2000,
        res = 300, casc = TRUE)
```

Arguments

PP Data of the PP_Optimizer function.

titles Titles of the graphics, if not set, assumes the default text.

xlabel Names the X axis, if not set, assumes the default text.

ylabel Names the Y axis, if not set, assumes the default text.

posleg 0 with no caption,
1 for caption in the left upper corner,
2 for caption in the right upper corner (default),
3 for caption in the right lower corner,
4 for caption in the left lower corner.

boxleg Puts the frame in the caption (default = TRUE).

size Size of the points in the graphs.

grid Put grid on graphs (default = TRUE).

color Colored graphics (default = TRUE).

classcolor Vector with the colors of the classes.

linlab Vector with the labels for the observations.

axesvar Puts axes of rotation of the variables, only when dimproj > 1 (default = TRUE).

axes Plots the X and Y axes (default = TRUE).

`savptc` Saves graphics images to files (default = FALSE).

`width` Graphics images width when savptc = TRUE (defaul = 3236).

`height` Graphics images height when savptc = TRUE (default = 2000).

`res` Nominal resolution in ppi of the graphics images when savptc = TRUE (default = 300).

`casc` Cascade effect in the presentation of the graphics (default = TRUE).

Value

Graph of the evolution of the indices, and graphs whose data were reduced in two dimensions.

Author(s)

Paulo Cesar Ossani

Marcelo Angelo Cirillo

See Also

`PP_Optimizer` and `PP_Index`

Examples

```
data(iris) # dataset

# Example 1 - Without the classes in the data
data <- iris[,1:4]

findex <- "kurtosismax" # index function

dim <- 1 # dimension of data projection

sphere <- TRUE # spherical data

res <- PP_Optimizer(data = data, class = NA, findex = findex,
                    optmethod = "GTSA", dimproj = dim, sphere = sphere,
                    weight = TRUE, lambda = 0.1, r = 1, cooling = 0.9,
                    eps = 1e-3, maxiter = 500, half = 30)

Plot.PP(res, titles = NA, posleg = 1, boxleg = FALSE, color = TRUE,
        linlab = NA, axesvar = TRUE, axes = TRUE, savptc = FALSE,
        width = 3236, height = 2000, res = 300, casc = FALSE)


# Example 2 - With the classes in the data
class <- iris[,5] # data class

res <- PP_Optimizer(data = data, class = class, findex = findex,
                    optmethod = "GTSA", dimproj = dim, sphere = sphere,
                    weight = TRUE, lambda = 0.1, r = 1, cooling = 0.9,
                    eps = 1e-3, maxiter = 500, half = 30)
```

```
tit <- c(NA,"Graph example") # titles for the graphics

Plot.PP(res, titles = tit, posleg = 1, boxleg = FALSE, color = TRUE,
        classcolor = c("blue3","red","goldenrod3"), linlab = NA,
        axesvar = TRUE, axes = TRUE, savptc = FALSE, width = 3236,
        height = 2000, res = 300, casc = FALSE)


# Example 3 - Without the classes in the data, but informing
#             the classes in the plot function
res <- PP_Optimizer(data = data, class = NA, findex = "Moment",
                    optmethod = "GTSA", dimproj = 2, sphere = sphere,
                    weight = TRUE, lambda = 0.1, r = 1, cooling = 0.9,
                    eps = 1e-3, maxiter = 500, half = 30)

lin <- c(rep("a",50),rep("b",50),rep("c",50)) # data class

Plot.PP(res, titles = NA, posleg = 1, boxleg = FALSE, color = TRUE,
        linlab = lin, axesvar = TRUE, axes = TRUE, savptc = FALSE,
        width = 3236, height = 2000, res = 300, casc = FALSE)


# Example 4 - With the classes in the data, but not informed in plot function
class <- iris[,5] # data class

dim <- 2 # dimension of data projection

findex <- "lda" # index function

res <- PP_Optimizer(data = data, class = class, findex = findex,
                    optmethod = "GTSA", dimproj = dim, sphere = sphere,
                    weight = TRUE, lambda = 0.1, r = 1, cooling = 0.9,
                    eps = 1e-3, maxiter = 500, half = 30)

tit <- c("",NA) # titles for the graphics

Plot.PP(res, titles = tit, posleg = 1, boxleg = FALSE, color = TRUE,
        linlab = NA, axesvar = TRUE, axes = TRUE, savptc = FALSE,
        width = 3236, height = 2000, res = 300, casc = FALSE)
```

PP_Index *Function to find the Projection Pursuit indexes (PP).*

Description

Function used to find Projection Pursuit indexes (PP).

Usage

```
PP_Index(data, class = NA, vector.proj = NA,
         findex = "HOLES", dimproj = 2, weight = TRUE,
         lambda = 0.1, r = 1, ck = NA)
```

Arguments

`data` Numeric dataset without class information.

`class` Vector with names of data classes.

`vector.proj` Vector projection.

`findex` Projection index function to be used:
"lda" - LDA index,
"pda" - PDA index,
"lr" - Lr index,
"holes" - Holes index (default),
"cm" - Central Mass index,
"pca" - PCA index,
"friedmantukey" - Friedman Tukey index,
"entropy" - Entropy index,
"legendre" - Legendre index,
"laguerrefourier" - Laguerre Fourier index,
"hermite" - Hermite index,
"naturalhermite" - Natural Hermite index,
"kurtosismax" - Maximum kurtosis index,
"kurtosismin" - Minimum kurtosis index,
"moment" - Moment index,
"mf" - MF index,
"chi" - Chi-square index.

`dimproj` Dimension of data projection (default = 2).

`weight` Used in index LDA, PDA and Lr to weight the calculations for the number of elements in each class (default = TRUE).

`lambda` Used in the PDA index (default = 0.1).

`r` Used in the Lr index (default = 1).

`ck` Internal use of the CHI index function.

Value

`num.class` Number of classes.

`class.names` Class names.

`findex` Projection index function used.

`vector.proj` Projection vectors found.

`index` Projection index found in the process.

Author(s)

Paulo Cesar Ossani

Marcelo Angelo Cirillo

References

OSSANI, P. C.; FIGUEIRA, M. R.; CIRILLO, M. A. Proposition of a new index for projection pursuit in the multiple factor analysis. *Computational and Mathematical Methods*, v. 1, p. 1-18, 2020.

COOK, D., BUJA, A., CABRERA, J.. Projection pursuit indexes based on orthonormal function expansions. *Journal of Computational and Graphical Statistics*, 2(3):225-250, 1993.

COOK, D., BUJA, A., CABRERA, J., HURLEY, C.. Grand tour and projection pursuit, *Journal of Computational and Graphical Statistics*, 4(3), 155-172, 1995.

COOK, D., SWAYNE, D. F.. Interactive and Dynamic Graphics for data Analysis: With R and GGobi. Springer. 2007.

ESPEZUA, S., VILLANUEVA, E., MACIEL, C.D., CARVALHO, A.. A projection pursuit framework for supervised dimension reduction of high dimensional small sample datasets. *Neurocomputing*, 149, 767-776, 2015.

FRIEDMAN, J. H., TUKEY, J. W. A projection pursuit algorithm for exploratory data analysis. *IEEE Transaction on Computers*, 23(9):881-890, 1974.

HASTIE, T., BUJA, A., TIBSHIRANI, R.: Penalized discriminant analysis. *The Annals of Statistics*. 23(1), 73-102 . 1995.

HUBER, P. J.. Projection pursuit. *Annals of Statistics*, 13(2):435-475, 1985.

JONES, M. C., SIBSON, R.. What is projection pursuit, (with discussion), *Journal of the Royal Statistical Society*, Series A 150, 1-36, 1987.

LEE, E. K., COOK, D.. A projection pursuit index for large p small n data. *Statistics and Computing*, 20(3):381-392, 2010.

LEE, E., COOK, D., KLINKE, S., LUMLEY, T.. Projection pursuit for exploratory supervised classification. *Journal of Computational and Graphical Statistics*, 14(4):831-846, 2005.

MARTINEZ, W. L., MARTINEZ, A. R.; *Computational Statistics Handbook with MATLAB*, 2th. ed. New York: Chapman & Hall/CRC, 2007. 794 p.

MARTINEZ, W. L., MARTINEZ, A. R., SOLKA, J.; *Exploratory data Analysis with MATLAB*, 2th. ed. New York: Chapman & Hall/CRC, 2010. 499 p.

PENA, D., PRIETO, F.. Cluster identification using projections. *Journal of the American Statistical Association*, 96(456):1433-1445, 2001.

POSSE, C.. Projection pursuit exploratory data analysis, *Computational Statistics and data Analysis*, 29:669-687, 1995a.

POSSE, C.. Tools for two-dimensional exploratory projection pursuit, *Journal of Computational and Graphical Statistics*, 4:83-100, 1995b.

See Also

`PP_Optimizer` and `Plot.PP`

Examples

```
data(iris) # data set

data <- iris[,1:4]

# Example 1 - Without the classes in the data
ind <- PP_Index(data = data, class = NA, vector.proj = NA,
                findex = "moment", dimproj = 2, weight = TRUE,
                lambda = 0.1, r = 1)

print("Number of classes:"); ind$num.class
print("class Names:"); ind$class.names
print("Projection index function:"); ind$findex
print("Projection vectors:"); ind$vector.proj
print("Projection index:"); ind$index


# Example 2 - With the classes in the data
class <- iris[,5] # data class

findex <- "pda" # index function

sphere <- TRUE # spherical data

res <- PP_Optimizer(data = data, class = class, findex = findex,
                    optmethod = "SA", dimproj = 2, sphere = sphere,
                    weight = TRUE, lambda = 0.1, r = 1, cooling = 0.9,
                    eps = 1e-3, maxiter = 1000, half = 30)

# Comparing the result obtained
if (match(toupper(findex),c("LDA", "PDA", "LR"), nomatch = 0) > 0) {
  if (sphere) {
     data <- apply(predict(prcomp(data)), 2, scale) # spherical data
  }
} else data <- as.matrix(res$proj.data[,1:Dim])

ind <- PP_Index(data = data, class = class, vector.proj = res$vector.opt,
                findex = findex, dimproj = 2, weight = TRUE, lambda = 0.1,
                r = 1)

print("Number of classes:"); ind$num.class
print("class Names:"); ind$class.names
print("Projection index function:"); ind$findex
print("Projection vectors:"); ind$vector.proj
print("Projection index:"); ind$index
print("Optimized Projection index:"); res$index[length(res$index)]
```

PP_Optimizer *Optimization function of the Projection Pursuit index (PP).*

Description

Optimization function of the Projection Pursuit index (PP).

Usage

```
PP_Optimizer(data, class = NA, findex = "HOLES",
             dimproj = 2, sphere = TRUE, optmethod = "GTSA",
             weight = TRUE, lambda = 0.1, r = 1, cooling = 0.9,
             eps = 1e-3, maxiter = 3000, half = 30)
```

Arguments

`data` Numeric dataset without class information.

`class` Vector with names of data classes.

`findex` Projection index function to be used:
"lda" - LDA index,
"pda" - PDA index,
"lr" - Lr index,
"holes" - Holes index (default),
"cm" - Central Mass index,
"pca" - PCA index,
"friedmantukey" - Friedman Tukey index,
"entropy" - Entropy index,
"legendre" - Legendre index,
"laguerrefourier" - Laguerre Fourier index,
"hermite" - Hermite index,
"naturalhermite" - Natural Hermite index,
"kurtosismax" - Maximum kurtosis index,
"kurtosismin" - Minimum kurtosis index,
"moment" - Moment index,
"mf" - MF index,
"chi" - Chi-square index.

`dimproj` Dimension of the data projection (default = 2).

`sphere` Spherical data (default = TRUE).

`optmethod` Optimization method GTSA - Grand Tour Simulated Annealing or SA - Simulated Annealing (default = "GTSA").

`weight` Used in index LDA, PDA and Lr to weight the calculations for the number of elements in each class (default = TRUE).

`lambda` Used in the PDA index (default = 0.1).

`r` Used in the Lr index (default = 1).

`cooling` Cooling rate (default = 0.9).

`eps` Approximation accuracy for cooling (default = 1e-3).

`maxiter` Maximum number of iterations of the algorithm (default = 3000).

`half` Number of steps without incrementing the index, then decreasing the cooling value (default = 30).

Value

`num.class` Number of classes.

`class.names` Class names.

`proj.data` Projected data.

`vector.opt` Projection vectors found.

`index` Vector with the projection indices found in the process, converging to the maximum, or the minimum.

`findex` Projection index function used.

Author(s)

Paulo Cesar Ossani

Marcelo Angelo Cirillo

References

COOK, D., LEE, E. K., BUJA, A., WICKHAM, H.. Grand tours, projection pursuit guided tours and manual controls. In Chen, Chunhouh, Hardle, Wolfgang, Unwin, e Antony (Eds.), *Handbook of data Visualization*, Springer Handbooks of Computational Statistics, chapter III.2, p. 295-314. Springer, 2008.

LEE, E., COOK, D., KLINKE, S., LUMLEY, T.. Projection pursuit for exploratory supervised classification. *Journal of Computational and Graphical Statistics*, 14(4):831-846, 2005.

See Also

`Plot.PP` and `PP_Index`

Examples

```
data(iris) # data set

# Example 1 - Without the classes in the data
data <- iris[,1:4]

class <- NA # data class

findex <- "kurtosismax" # index function

dim <- 1 # dimension of data projection

sphere <- TRUE # spherical data

res <- PP_Optimizer(data = data, class = class, findex = findex,
                    optmethod = "GTSA", dimproj = dim, sphere = sphere,
                    weight = TRUE, lambda = 0.1, r = 1, cooling = 0.9,
                    eps = 1e-3, maxiter = 1000, half = 30)

print("Number of classes:"); res$num.class
print("class Names:"); res$class.names
print("Projection index function:"); res$findex
print("Projected data:"); res$proj.data
print("Projection vectors:"); res$vector.opt
print("Projection index:"); res$index


# Example 2 - With the classes in the data
class <- iris[,5] # classe dos dados

res <- PP_Optimizer(data = data, class = class, findex = findex,
                    optmethod = "GTSA", dimproj = dim, sphere = sphere,
                    weight = TRUE, lambda = 0.1, r = 1, cooling = 0.9,
                    eps = 1e-3, maxiter = 1000, half = 30)

print("Number of classes:"); res$num.class
print("class Names:"); res$class.names
print("Projection index function:"); res$findex
print("Projected data:"); res$proj.data
print("Projection vectors:"); res$vector.opt
print("Projection index:"); res$index
```

`Pursuit` *Projection Pursuit*

Description

Projection pursuit (PP) with 17 methods and grand tour with 3 methods. Being that projection pursuit searches for low-dimensional linear projections in high-dimensional data structures, while grand tour is a technique used to explore multivariate statistical data through animation.

Details

Package:	Pursuit
Type:	Package
Version:	1.0.4
Date:	2023-08-19
License:	GPL(>= 2)
LazyLoad:	yes

Author(s)

Paulo Cesar Ossani and Marcelo Angelo Cirillo.

Maintainer: Paulo Cesar Ossani <ossanipc@hotmail.com>

References

ASIMOV, D. The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal of Scientific and Statistical Computing*, 6(1), 128-143, 1985.

ASIMOV, D.; BUJA, A. The grand tour via geodesic interpolation of 2-frames. in Visual Data Exploration and Analysis. *Symposium on Electronic Imaging Science and Technology*, IS&T/SPIE. 1994.

BUJA, A.; ASIMOV, D. Grand tour methods: An outline. *Computer Science and Statistics*, 17:63-67. 1986.

BUJA, A.; COOK, D.; ASIMOV, D.; HURLEY, C. Computational Methods for High-Dimensional Rotations in Data Visualization, in C. R. Rao, E. J. Wegman & J. L. Solka, eds, *"Handbook of Statistics: Data Mining and Visualization"*, Elsevier/North Holland, http://www.elsevier.com, pp. 391-413. 2005.

COOK, D., LEE, E. K., BUJA, A., WICKHAM, H.. Grand tours, projection pursuit guided tours and manual controls. In Chen, Chunhouh, Hardle, Wolfgang, Unwin, e Antony (Eds.), *Handbook of Data Visualization*, Springer Handbooks of Computational Statistics, chapter III.2, p. 295-314. Springer, 2008.

COOK, D., BUJA, A., CABRERA, J.. Projection pursuit indexes based on orthonormal function expansions. *Journal of Computational and Graphical Statistics*, 2(3):225-250, 1993.

COOK, D., BUJA, A., CABRERA, J., HURLEY, C.. Grand tour and projection pursuit, *Journal of Computational and Graphical Statistics*, 4(3), 155-172, 1995.

COOK, D., SWAYNE, D. F.. Interactive and Dynamic Graphics for Data Analysis: With R and GGobi. Springer. 2007.

ESPEZUA, S., VILLANUEVA, E., MACIEL, C.D., CARVALHO, A.. A projection pursuit framework for supervised dimension reduction of high dimensional small sample datasets. *Neurocomputing*, 149, 767-776, 2015.

FRIEDMAN, J. H., TUKEY, J. W. A projection pursuit algorithm for exploratory data analysis. *IEEE Transaction on Computers*, 23(9):881-890, 1974.

HASTIE, T., BUJA, A., TIBSHIRANI, R.: Penalized discriminant analysis. *The Annals of Statistics*. 23(1), 73-102 . 1995.

HUBER, P. J.. Projection pursuit. *Annals of Statistics*, 13(2):435-475, 1985.

HURLEY, C.; BUJA, A. Analyzing high-dimensional data with motion graphics, *SIAM Journal of Scientific and Statistical Computing*, 11 (6), 1193-1211. 1990.

JONES, M. C., SIBSON, R.. What is projection pursuit, (with discussion), *Journal of the Royal Statistical Society*, Series A 150, 1-36, 1987.

LEE, E., COOK, D., KLINKE, S., LUMLEY, T.. Projection pursuit for exploratory supervised classification. *Journal of Computational and Graphical Statistics*, 14(4):831-846, 2005.

LEE, E. K., COOK, D.. A projection pursuit index for large p small n data. *Statistics and Computing*, 20(3):381-392, 2010.

MARTINEZ, W. L., MARTINEZ, A. R.; *Computational Statistics Handbook with MATLAB*, 2th. ed. New York: Chapman & Hall/CRC, 2007. 794 p.

MARTINEZ, W. L., MARTINEZ, A. R., SOLKA, J.; *Exploratory Data Analysis with MATLAB*, 2th. ed. New York: Chapman & Hall/CRC, 2010. 499 p.

PENA, D., PRIETO, F.. Cluster identification using projections. *Journal of the American Statistical Association*, 96(456):1433-1445, 2001.

POSSE, C.. Projection pursuit exploratory data analysis, *Computational Statistics and Data Analysis*, 29:669-687, 1995a.

POSSE, C.. Tools for two-dimensional exploratory projection pursuit, *Journal of Computational and Graphical Statistics*, 4:83-100, 1995b

YOUNG, F. W.; RHEINGANS P. Visualizing structure in high-dimensional multivariate data, *IBM Journal of Research and Development*, 35:97-107, 1991.

YOUNG, F. W.; FALDOWSKI R. A.; McFARLANE M. M. *Multivariate statistical visualization, in Handbook of Statistics*, Vol 9, C. R. Rao (ed.), The Netherlands: Elsevier Science Publishers, 959-998, 1993.

Index