Package ‘PPbigdata’

April 30, 2024

Type Package

Title Projection Pursuit for Big Data Based on Data Nuggets

Version 1.0.0

Date 2024-4-1

Author Yajie Duan [aut, cre],
Javier Cabrera [aut]

Maintainer Yajie Duan <yajieritaduan@gmail.com>

Description Perform 1-dim/2-dim projection pursuit, grand tour and guided tour for big data based on data nuggets.
Reference papers: [1] Beavers et al., (2024) <doi:10.1080/10618600.2024.2341896>.
[2] Duan, Y., Cabrera, J., & Emir, B. (2023). ``A New Projection Pursuit Index for Big Data.'' <doi:10.48550/arXiv.2312.06465>.

Depends R (>= 4.0), stats, datanugget(>= 1.2.4), MASS

Imports dplyr, magrittr, weights, rstiefel, gtools, tourr, mclust,
graphics, grDevices

License GPL-2

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2024-04-30 11:42:50 UTC

R topics documented:

PPbigdata-package *Projection Pursuit for Big Data Based on Data Nuggets*

Description

This package contains functions to perform 1-dim/2-dim projection pursuit for big data based on data nuggets. It includes PP indices for data nuggets, static PP for big data by optimization of PP index, grand tour and guided tour for big data based on data nuggets, and visialization functions for projections and variable loadings.

Author(s)

Yajie Duan, Javier Cabrera

References

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Cherasia, K. E., Cabrera, J., Fernholz, L. T., & Fernholz, R. (2022). Data Nuggets in Supervised Learning. *In Robust and Multivariate Statistical Methods: Festschrift in Honor of David E. Tyler* (pp. 429-449). Cham: Springer International Publishing.

Cook, D., Buja, A., & Cabrera, J. (1993). Projection pursuit indexes based on orthonormal function expansions. Journal of Computational and Graphical Statistics, 2(3), 225-250.

Cook, D., Buja, A., Cabrera, J., & Hurley, C. (1995). Grand tour and projection pursuit. Journal of Computational and Graphical Statistics, 4(3), 155-172.

Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2011). tourr: An R package for exploring multivariate data with projections. Journal of Statistical Software, 40, 1-18.

Cabrera, J., & McDougall, A. (2002). Statistical consulting. Springer Science & Business Media.

Horst, P. (1965). Factor Analysis of Data Matrices. Holt, Rinehart and Winston. Chapter 10.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. Psychometrika, 23(3), 187-200.

See Also

`PPnugg`, `NHnugg`, `HoleNugg`, `plotNugg`, `faProj`, `guidedTourNugg`, `grandTourNugg`, `create.DN`, `refine.DN`

CMNugg *Central Mass Index for projected Big Data based on data nuggets*

Description

This function calculates the value of Central Mass index, a Projection Pursuit index, for projected big data based on data nuggets.

Usage

```
CMNugg(nuggproj,weight)
```

Arguments

`nuggproj`	Projected data nugget centers. Must be a data matrix (of class matrix, or data.frame) or a vector containing only entries of class numeric.
`weight`	Vector of the weight parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nuggproj) or length(nuggproj). Must be of class numeric or integer.

Details

This function calculates the value of Central Mass index, a Projection Pursuit index for projected Big Data based on data nuggets.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

Central Mass index is a kind of Projection Pursuit (PP) index, and larger index values indicate a central mass structure of multivariate data. However, it's computationally hard to calculate the index for big data because of the vector memory limit during calculation. To deal with big data, data nuggets could be used to calculate the index efficiently. In this function, based on the projected data nugget centers with data nugget weights, the Central Mass index is calculated by 1 minus a Hole index based on data nuggets. See `HoleNugg`

Value

A numeric value indicating Central Mass index value of the projected big data based on the data nuggets.

Author(s)

Yajie Duan, Javier Cabrera

References

Che?rdle, W. K., & Unwin, A. (Eds.). (2007). Handbook of data visualization. Springer Science & Business Media.

Cook, D., Buja, A., Cabrera, J., & Hurley, C. (1995). Grand tour and projection pursuit. Journal of Computational and Graphical Statistics, 4(3), 155-172.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

See Also

HoleNugg, NHnugg,create.DN, refine.DN

Examples

```
 require(datanugget)
 require(rstiefel)

 #4-dim small example
 X = cbind.data.frame(rnorm(5*10^3),
                      rnorm(5*10^3,2,1),
                      rnorm(5*10^3,5,2),
                      rnorm(5*10^3))

#raw data is recommended to be scaled firstly to generate data nuggets for Projection Pursuit
 X = as.data.frame(scale(X))

 #create data nuggets
 my.DN = create.DN(x = X,
                   R = 500,
                   delete.percent = .1,
                   DN.num1 = 500,
                   DN.num2 = 250,
                   no.cores = 0,
                   make.pbs = FALSE)


 #refine data nuggets
 my.DN2 = refine.DN(x = X,
                    DN = my.DN,
                    EV.tol = .9,
                    min.nugget.size = 2,
                    max.splits = 5,
                    no.cores = 0,
                    make.pbs = FALSE)

 #get nugget centers, weights, and scales
```

```
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale

#spherize the data nuggets with weights to calculate the PP index
nugg_wsph <- wsph(nugg,weight)$data_wsph

#generate a random orthonormal matrix as a projection matrix to 2-dim space
proj_2d = rustiefel(4, 2)

#project data nugget centers into 2-dim space by the random projection matrix
nuggproj_2d = as.matrix(nugg_wsph)%*%proj_2d

#plot the projected data nuggets
#lighter green represents larger weights
plotNugg(nuggproj_2d, weight)

#calculate the CM index for the projected 2-dim big data
CMNugg(nuggproj_2d,weight)
```

faProj — *Factor rotation for projected Big Data in multi-dimensional space based on data nuggets*

Description

This function performs the factor rotation for projected big data in multi-dimensional space based on data nuggets.

Usage

```
faProj(nugg, weight, wsph_proj = NULL, proj, method = c("varimax","promax"))
```

Arguments

nugg	Data nugget centers obtained from raw data. Must be a data matrix (of class matrix, or data.frame) with at least two columns.
weight	Vector of the weight parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nugg). Must be of class numeric or integer.
wsph_proj	Matrix of size ncol(nugg) by ncol(nugg). It's the sphering/whitening matrix considering nugget weights for the transformation. The projection is on the spherized data nugget centers considering weights, which is obtained by multiplying the centered data nuggets with weights by this sphering/whitening matrix. Default is NULL, which would be obtained by function wsph. Must be a data matrix containing only entries of class numeric.

`proj`	Matrix of size ncol(nugg) by projection dimenstion. It's the orthonormal projection matrix that would be taken on the spherized data nugget centers considering weights, to obtain projected data nuggets. Must be a data matrix containing only entries of class numeric.
`method`	A character indicating the rotation method used for factor analysis. The default method "`varimax`" uses function `varimax` to take rotation; the alternative "`promax`" uses function `promax`. The rotation is taken on the overall transformation matrix for the raw data nuggets, which is a combination of spherization matrix and projection matrix, to back to the original variables.

Details

This function performs the factor rotation for projected big data in multi-dimensional space based on data nuggets.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

After obtaining created and refined data nuggets for big data, data nugget centers needs to be spherized considering nugget weights before conducting projection pursuit. The optimal or interested projection found by projection pursuit would be taken on the spherized nugget centers. This function conducts the factor analysis for the projected data nugget centers. The default rotation method "`varimax`" uses function `varimax` to take rotation; the alternative "`promax`" uses function `promax`. The rotation is taken on the overall transformation matrix for the raw data nuggets, which is a combination of spherization matrix and projection matrix, to back to the original variables.

Value

A list containing the following components:

`nuggproj_rotat`	The rotated projected data nugget centers after conducting factor ratation. It's obtained by multiplying the centered data nuggets `nugg_wcen` with the rotated matrix `loadings`.
`loadings`	A matrix of loadings for original variables, one column for each projection direction. It's the rotated transformation matrix to obtain updated projected data nugget centers.
`nugg_wcen`	The centered data nugget centers that has a zero weighted mean for each column considering nugget weights. It's obtained by extracting the weighted mean from the original data nugget centers.

Author(s)

Yajie Duan, Javier Cabrera

References

Cook, D., Buja, A., & Cabrera, J. (1993). Projection pursuit indexes based on orthonormal function expansions. Journal of Computational and Graphical Statistics, 2(3), 225-250.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

Hendrickson, A. E., & White, P. O. (1964). Promax: A quick method for rotation to oblique simple structure. British journal of statistical psychology, 17(1), 65-70.

Horst, P. (1965). Factor Analysis of Data Matrices. Holt, Rinehart and Winston. Chapter 10.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. Psychometrika, 23(3), 187-200.

See Also

PPnugg, NHnugg,create.DN, refine.DN

Examples

```
 require(datanugget)
 require(rstiefel)

 #4-dim small example with cluster stuctures in V3 and V4
 X = cbind.data.frame(V1 = rnorm(5*10^3,mean = 5,sd = 2),
                      V2 = rnorm(5*10^3,mean = 5,sd = 1),
                      V3 = c(rnorm(3*10^3,sd = 0.3),
                             rnorm(2*10^3,mean = 2, sd = 0.3)),
                      V4 = c(rnorm(1*10^3,mean = -8, sd = 1),
                             rnorm(3*10^3,mean = 0,sd = 1),
                             rnorm(1*10^3,mean = 7, sd = 1.5)))

#raw data is recommended to be scaled firstly to generate data nuggets for Projection Pursuit
 X = as.data.frame(scale(X))

 #create data nuggets
 my.DN = create.DN(x = X,
                  R = 500,
                  delete.percent = .1,
                  DN.num1 = 500,
                  DN.num2 = 250,
                  no.cores = 2,
                  make.pbs = FALSE)


 #refine data nuggets
 my.DN2 = refine.DN(x = X,
                    DN = my.DN,
                    EV.tol = .9,
                    min.nugget.size = 2,
                    max.splits = 5,
                    no.cores = 2,
                    make.pbs = FALSE)

 #get nugget centers, weights, and scales
 nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
 weight = my.DN2$`Data Nuggets`$Weight
 scale = my.DN2$`Data Nuggets`$Scale

 #spherize data nugget centers considering weightsn to conduct Projection Pursuit
 wsph.res = wsph(nugg,weight)
 nugg_wsph = wsph.res$data_wsph
 wsph_proj = wsph.res$wsph_proj

 #conduct the same spherization projection on the standardized raw data
 X_cen = X- as.matrix(rep(1,nrow(X)))%*%wsph.res$wmean
 X_sph = as.matrix(X_cen)%*%wsph_proj

 #conduct Projection Pursuit in 2-dim by optimizing Natural Hermite index
 res = PPnuggOptim(NHnugg, nugg_wsph, dimproj = 2, weight = weight, scale = scale)

 #optimal projection matrix obtained
 proj_opt = res$proj.opt

 #plot projected data nuggets
 plotNugg(nugg_wsph%*%proj_opt,weight,qt = 0.8)

 #conduct varimax rotation for projection
 fa = faProj(nugg,weight,proj = proj_opt)

 #obtain rotated projected data nuggets and
 #corresponding loadings of original variables
 nuggproj_rotat = fa$nuggproj_rotat
 loadings = fa$loadings

 #plot rotated projected data nuggets after varimax rotation
 plotNugg(nuggproj_rotat,weight,qt = 0.8)

 #plot corresponding projected raw big data after factor roation
 X_proj = as.matrix(X_cen)%*%loadings
 plot(X_proj,cex = 0.5)

 #plot loadings of original variables
 #V3 and V4 have large loadings, same as the simulation setting.
 plotLoadings(loadings)
```

grandTourNugg *1-dim/2-dim Grand Tour for Big Data based on Data Nuggets*

Description

This function performs a 1-dim/2-dim grand tour path for big data based on constructed data nuggets. The grand tour finds projections at random.

Usage

```
grandTourNugg(nugg, weight, dim, qt = 0.8,...)
```

Arguments

`nugg` Data nugget centers obtained from raw data. Must be a data matrix (of class matrix, or data.frame) with at least two columns.

`weight` Vector of the weight parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nugg). Must be of class numeric or integer.

`dim` A numerical value indicating the target dimensionality for the tour. It's either 1 or 2.

`qt` For projected plots of 2-dim tour, a scalar with value in `[0,1]` indicating the probability used to obtain a sample quantile of the data nugget weights as the maximal value to transform weights to colors for the plot. Defaults to be 0.8. See `plotNugg`.

`...` Other arguments sent to `animate`, `animate_xy`, `display_dist`, `display_xy`, `wtd.hist`.

Details

This function performs a 1-dim/2-dim grand tour path for big data based on constructed data nuggets. The grand tour finds projections randomly.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

Based on the data nuggets from big data, a grand tour is performed to explore the multivariate data. It walks randomly to discover 1-dim/2-dim projections. This function for data nuggets is based on functions about grand tour in the package `tourr`. See details in `grand_tour`, `animate`, `animate_xy`, `display_dist`, and `display_xy`. For 2-dim grand tour, the projected data nugget centers are plotted with colors based on their weights where lighter green represents larger weights. For 1-dim grand tour, a weighted density histgram of 1-dim projected data nugget centers is plotted considering the data nugget weights. See details in `wtd.hist`. The loadings of each variable for projections are also shown at each step.

Value

A list containing the bases, index values, and other information during the tour.

Author(s)

Yajie Duan, Javier Cabrera

References

Cook, D., Buja, A., Cabrera, J., & Hurley, C. (1995). Grand tour and projection pursuit. Journal of Computational and Graphical Statistics, 4(3), 155-172.

Cook, D., Buja, A., & Cabrera, J. (1993). Projection pursuit indexes based on orthonormal function expansions. Journal of Computational and Graphical Statistics, 2(3), 225-250.

Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2011). tourr: An R package for exploring multivariate data with projections. Journal of Statistical Software, 40, 1-18.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

See Also

PPnugg, NHnugg,create.DN, refine.DN, guided_tour, animate

Examples

```
require(datanugget)

#4-dim small example with cluster stuctures in V3 and V4
X = cbind.data.frame(V1 = rnorm(5*10^3,mean = 5,sd = 2),
                     V2 = rnorm(5*10^3,mean = 5,sd = 1),
                     V3 = c(rnorm(3*10^3,sd = 0.3),
                            rnorm(2*10^3,mean = 2, sd = 0.3)),
                     V4 = c(rnorm(1*10^3,mean = -8, sd = 1),
                            rnorm(3*10^3,mean = 0,sd = 1),
                            rnorm(1*10^3,mean = 7, sd = 1.5)))

#raw data is recommended to be scaled firstly to generate data nuggets for Projection Pursuit
X = as.data.frame(scale(X))

#create data nuggets
my.DN = create.DN(x = X,
                  R = 500,
                  delete.percent = .1,
                  DN.num1 = 500,
                  DN.num2 = 250,
                  no.cores = 0,
                  make.pbs = FALSE)


#refine data nuggets
my.DN2 = refine.DN(x = X,
                    DN = my.DN,
                    EV.tol = .9,
                    min.nugget.size = 2,
                    max.splits = 5,
                    no.cores = 0,
                    make.pbs = FALSE)

#get nugget centers, weights, and scales
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale

#2-dim grand tour based on data nuggets
grandTourNugg(nugg,weight,dim = 2,cex = 0.5)

#1-dim grand tour based on data nuggets
grandTourNugg(nugg,weight,dim = 1,density_max = 4.5)
```

`guidedTourNugg` *1-dim/2-dim Guided Tour for Big Data based on Data Nuggets*

Description

This function performs a 1-dim/2-dim guided tour path for big data based on constructed data nuggets with their weights and scales. The guided tour tries to find a projection with a higher value of PP index than the current projection.

Usage

```
guidedTourNugg(nugg, weight, scale, dim, index = c("NH","Hole","CM"), qt = 0.8,...)
```

Arguments

`nugg` Data nugget centers obtained from raw data. Must be a data matrix (of class matrix, or data.frame) with at least two columns.

`weight` Vector of the weight parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nugg). Must be of class numeric or integer.

`scale` Vector of the scale parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nuggproj) for 2-dim projection/length(nuggproj) for 1-dim projection. Must be of class numeric or integer.

`dim` A numerical value indicating the target dimensionality for the tour. It's either 1 or 2.

`index` A character indicating the PP index function to be used to guide the tour: "NH" - Natural Hermite Index for data nuggets "Hole" - Hole Index for data nuggets "CM" - Central Mass Index for data nuggets `qt` For projected plots of 2-dim tour, a scalar with value in `[0,1]` indicating the probability used to obtain a sample quantile of the data nugget weights as the maximal value to transform weights to colors for the plot. Defaults to be 0.8. See `plotNugg`.

`...` Other arguments sent to `NHnugg`, `guided_tour`, `animate`, `animate_xy`, `display_dist`, `display_xy`, `wtd.hist`.

Details

This function performs a 1-dim/2-dim guided tour path for big data based on constructed data nuggets with their weights and scales. The guided tour tries to find a projection with a higher value of PP index than the current projection.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

Based on the data nuggets from big data, a projection pursuit guided tour is performed to explore the multivariate data. Unlike walking randomly to discover projections in grand tour, the guided tour selects the next target basis by optimizing a projection pursuit index function defining interesting projections. Here the considered choices of PP indices include Nature Hermite index, Hole index and CM index for big data based on data nuggets. See details in `NHnugg`, `HoleNugg`, and `CMNugg`.

This function for data nuggets is based on functions about guided tour in the package `tourr`. See details in `guided_tour`, `animate`, `animate_xy`, `display_dist`, and `display_xy`. For 2-dim guided tour, the projected data nugget centers are plotted with colors based on their weights where lighter green represents larger weights. For 1-dim guided tour, a weighted density histgram of 1-dim projected data nugget centers is plotted considering the data nugget weights. See details in `wtd.hist`. The loadings of each variable for projections are also shown at each step.

Value

A list containing the bases, index values, and other information during the tour.

Author(s)

Yajie Duan, Javier Cabrera

References

Cook, D., Buja, A., Cabrera, J., & Hurley, C. (1995). Grand tour and projection pursuit. Journal of Computational and Graphical Statistics, 4(3), 155-172.

Cook, D., Buja, A., & Cabrera, J. (1993). Projection pursuit indexes based on orthonormal function expansions. Journal of Computational and Graphical Statistics, 2(3), 225-250.

Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2011). tourr: An R package for exploring multivariate data with projections. Journal of Statistical Software, 40, 1-18.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

See Also

PPnugg, NHnugg,create.DN, refine.DN, guided_tour, animate

Examples

```
require(datanugget)

#4-dim small example with cluster stuctures in V3 and V4
X = cbind.data.frame(V1 = rnorm(5*10^3,mean = 5,sd = 2),
                     V2 = rnorm(5*10^3,mean = 5,sd = 1),
                     V3 = c(rnorm(3*10^3,sd = 0.3),
                            rnorm(2*10^3,mean = 2, sd = 0.3)),
                     V4 = c(rnorm(1*10^3,mean = -8, sd = 1),
                            rnorm(3*10^3,mean = 0,sd = 1),
                            rnorm(1*10^3,mean = 7, sd = 1.5)))

#raw data is recommended to be scaled firstly to generate data nuggets for Projection Pursuit
X = as.data.frame(scale(X))

#create data nuggets
my.DN = create.DN(x = X,
                  R = 500,
                  delete.percent = .1,
                  DN.num1 = 500,
                  DN.num2 = 250,
                  no.cores = 0,
                  make.pbs = FALSE)


#refine data nuggets
my.DN2 = refine.DN(x = X,
                   DN = my.DN,
                   EV.tol = .9,
                   min.nugget.size = 2,
                   max.splits = 5,
                   no.cores = 0,
                   make.pbs = FALSE)

#get nugget centers, weights, and scales
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale

#2-dim guided tour by Natural Hermite Index based on data nuggets
guidedTourNugg(nugg,weight,scale,dim = 2,index = "NH",cex = 0.5,max.tries = 15)

#1-dim guided tour by Hole Index based on data nuggets
guidedTourNugg(nugg,weight,scale,dim = 1,index = "Hole",density_max = 4.5)
```

HoleNugg *Hole Index for projected Big Data based on data nuggets*

Description

This function calculates the value of Hole index, a Projection Pursuit index, for projected big data based on data nuggets.

Usage

```
HoleNugg(nuggproj,weight)
```

Arguments

nuggproj	Projected data nugget centers. Must be a data matrix (of class matrix, or data.frame) or a vector containing only entries of class numeric.
weight	Vector of the weight parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nuggproj) or length(nuggproj). Must be of class numeric or integer.

Details

This function calculates the value of Hole index, a Projection Pursuit index for projected Big Data based on data nuggets.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

Hole index is a kind of Projection Pursuit (PP) index, and larger index values indicate a hole structure of multivariate data. However, it's computationally hard to calculate the index for big data because of the vector memory limit during calculation. To deal with big data, data nuggets could be used to calculate the index efficiently. In this function, based on the projected data nugget centers with data nugget weights, the Hole index is calculated via a weighted version of the original Hole index formula.

Value

A numeric value indicating Hole index value of the projected big data based on the data nuggets.

Author(s)

Yajie Duan, Javier Cabrera

References

Chen, C. H., Hardle, W. K., & Unwin, A. (Eds.). (2007). Handbook of data visualization. Springer Science & Business Media.

Cook, D., Buja, A., Cabrera, J., & Hurley, C. (1995). Grand tour and projection pursuit. Journal of Computational and Graphical Statistics, 4(3), 155-172.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

See Also

`CMNugg`, `NHnugg`,`create.DN`, `refine.DN`

Examples

```
require(datanugget)
require(rstiefel)

#4-dim small example
X = cbind.data.frame(rnorm(5*10^3),
                     rnorm(5*10^3,2,1),
                     rnorm(5*10^3,5,2),
                     rnorm(5*10^3))

#raw data is recommended to be scaled firstly to generate data nuggets for Projection Pursuit
X = as.data.frame(scale(X))

#create data nuggets
my.DN = create.DN(x = X,
                  R = 500,
                  delete.percent = .1,
                  DN.num1 = 500,
                  DN.num2 = 250,
                  no.cores = 0,
                  make.pbs = FALSE)


#refine data nuggets
my.DN2 = refine.DN(x = X,
                   DN = my.DN,
                   EV.tol = .9,
                   min.nugget.size = 2,
                   max.splits = 5,
                   no.cores = 0,
                   make.pbs = FALSE)

#get nugget centers, weights, and scales
```

```
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale

#spherize the data nuggets with weights to calculate the PP index
nugg_wsph <- wsph(nugg,weight)$data_wsph

#generate a random orthonormal matrix as a projection matrix to 2-dim space
proj_2d = rustiefel(4, 2)

#project data nugget centers into 2-dim space by the random projection matrix
nuggproj_2d = as.matrix(nugg_wsph)%*%proj_2d

#plot the projected data nuggets
#lighter green represents larger weights
plotNugg(nuggproj_2d, weight)

#calculate the Hole index for the projected 2-dim big data
HoleNugg(nuggproj_2d,weight)
```

NHnugg *Natural Hermite Index for projected 1-dim/2-dim Big Data based on data nuggets*

Description

This function calculates the value of Nature Hermite index, a Projection Pursuit index proposed by Cook(1993) for projected 1-dim/2-dim big data based on data nuggets.

Usage

```
NHnugg(nuggproj, weight, scale,
       bandwidth = NULL, gridn = 300,lims = NULL, gridnAd = TRUE)
```

Arguments

`nuggproj` Projected data nugget centers in 1-dim/2-dim space. Must be a data matrix (of class matrix, or data.frame) with two columns or a vector containing only entries of class numeric.

`weight` Vector of the weight parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nuggproj) for 2-dim projection/length(nuggproj) for 1-dim projection. Must be of class numeric or integer.

`scale` Vector of the scale parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nuggproj) for 2-dim projection/length(nuggproj) for 1-dim projection. Must be of class numeric or integer.

`bandwidth` Bandwidth in each direction that would be combined with data nuggets scales as the final bandwith for kernal density estimation of projected data nuggets. Defaults to normal reference bandwidth considering the weights. Can be scalar or a length-2 numeric vector. For 2-dim projection, a scalar value will be applied on both directions.

`gridn` Number of grid points in each direction used for kernel density estimation of projected data. Can be scalar or a length-2 integer vector.

`lims` The limits of each direction used for kernel density estimation of projected data. Must be a length-4 numeric vector as (xlow, xupper, ylow, yupper) for 2-dim projected data, or a length-2 numeric vector as (xlow, xupper) for 1-dim projected data. If NULL, defaulting to the range of each direction.

`gridnAd` logical; if `TRUE` (default) and `gridn` is a scalar, for 2-dim projected data `rawproj`, `gridn` is used for x-direction, and the number of grid points in y-direction is adjusted by the limits of both directions, i.e., `round(gridn*diff(lims[3:4])/diff(lims[1:2]))`. Ignorable when `gridn` is a length-2 integer vector or projected data `rawproj` is 1-dim.

Details

This function calculates the value of Nature Hermite index, a Projection Pursuit index proposed by Cook(1993) for projected 1-dim/2-dim Big Data based on data nuggets.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

Natural Hermite index is one kind of Projection Pursuit (PP) index, and it measures the distance between the density of projected data and the standard normal. Larger index values indicate a hidden structure of multivariate data, such as clustersing, outliers or other non-linear structures. However, it's computationally hard to calculate the index for big data because of the issue about density estimation of projected big data. A new PP index for big data was proposed by Duan(2023), which is based on the Natural Hermite index and data nuggets.

In this function, the PP index value for projected 1-dim/2-dim big data is calculated based on created and refined data nuggets. Data nuggets are firstly created and refined for the big data. For Natural Hermite index, the data nugget centers need to be spherized considering nugget weights before projection. The projection is taken on the spherized data nugget centers to obtain projected data nuggets. The density values of projected big data are firstly estimated by `nuggKDE`. Based on it, the Natural Hermite index value is calculated via numerical integral by summation.

Value

A numeric value indicating Nature Hermite index value of the projected big data based on the data nuggets.

Author(s)

Yajie Duan, Javier Cabrera

References

Cook, D., Buja, A., & Cabrera, J. (1993). Projection pursuit indexes based on orthonormal function expansions. Journal of Computational and Graphical Statistics, 2(3), 225-250.

Cook, D., Buja, A., Cabrera, J., & Hurley, C. (1995). Grand tour and projection pursuit. Journal of Computational and Graphical Statistics, 4(3), 155-172.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

See Also

nuggKDE,create.DN, refine.DN

Examples

```
 require(datanugget)
 require(rstiefel)

 #4-dim small example
 X = cbind.data.frame(rnorm(5*10^3),
                      rnorm(5*10^3,2,1),
                      rnorm(5*10^3,5,2),
                      rnorm(5*10^3))

#raw data is recommended to be scaled firstly to generate data nuggets for Projection Pursuit
 X = as.data.frame(scale(X))

 #create data nuggets
 my.DN = create.DN(x = X,
                   R = 500,
                   delete.percent = .1,
                   DN.num1 = 500,
                   DN.num2 = 250,
                   no.cores = 0,
                   make.pbs = FALSE)


 #refine data nuggets
 my.DN2 = refine.DN(x = X,
                    DN = my.DN,
                    EV.tol = .9,
                    min.nugget.size = 2,
                    max.splits = 5,
                    no.cores = 0,
                    make.pbs = FALSE)

 #get nugget centers, weights, and scales
```

```
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale

#spherize the data nuggets with weights to calculate the PP index
nugg_wsph <- wsph(nugg,weight)$data_wsph

#generate a random orthonormal matrix as a projection matrix to 2-dim space
proj_2d = rustiefel(4, 2)

#project data nugget centers into 2-dim space by the random projection matrix
nuggproj_2d = as.matrix(nugg_wsph)%*%proj_2d

#plot the projected data nuggets
#lighter green represents larger weights
plotNugg(nuggproj_2d, weight)

#calculate the Natural Hermite index for the projected 2-dim big data
NHnugg(nuggproj_2d,weight,scale)
```

---

nuggKDE — *Density Estimation for projected 1-dim/2-dim big data based on data nuggets*

Description

This function estimates the density function of projected 1-dim/2-dim big data based on data nuggets and kernal density esimation.

Usage

```
nuggKDE(nuggproj, weight, scale, h = NULL, gridn = 300,lims = NULL, gridnAd = TRUE)
```

Arguments

`nuggproj` Projected data nugget centers in 1-dim/2-dim space. Must be a data matrix (of class matrix, or data.frame) with two columns or a vector containing only entries of class numeric.

`weight` Vector of the weight parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nuggproj) for 2-dim projection/length(nuggproj) for 1-dim projection. Must be of class numeric or integer.

`scale` Vector of the scale parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nuggproj) for 2-dim projection/length(nuggproj) for 1-dim projection. Must be of class numeric or integer.

`h` Bandwidth in each direction that would be combined with data nuggets scales as the final bandwith for kernal density estimation of projected data nuggets. Defaults to normal reference bandwidth considering the nugget weights, i.e., bandwidth.nrd(rep(dat,weight)). Can be scalar or a length-2 numeric vector. For 2-dim projection, a scalar value will be applied on both directions.

gridn
: Number of grid points in each direction used for kernel density estimation of projected data. Can be scalar or a length-2 integer vector.

lims
: The limits of each direction used for kernel density estimation of projected data. Must be a length-4 numeric vector as (xlow, xupper, ylow, yupper) for 2-dim projected data, or a length-2 numeric vector as (xlow, xupper) for 1-dim projected data. If NULL, defaulting to the range of each direction.

gridnAd
: logical; if TRUE (default) and gridn is a scalar, for 2-dim projected data rawproj, gridn is used for x-direction, and the number of grid points in y-direction is adjusted by the limits of both directions, i.e., round(gridn*diff(lims[3:4])/diff(lims[1:2])). Ignorable when gridn is a length-2 integer vector or projected data rawproj is 1-dim.

Details

This function calculates the estimated density values of projected 1-dim/2-dim big based on data nuggets.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions create.DN or refine.DN in the package datanugget.

Based on the created and refined data nuggets, the density of projected 1-dim/2-dim big data could be estimated via a revised version of kernal density estimation considering the data nugget centers, weightes and scales. For the estimation, the normal kernal is used with a bandwidth being a combination of pre-specified bandwidth and scales of data nuggets. By default, the pre-specified bandwidth in each direction is a normal reference bandwidth considering the nugget weights, i.e., bandwidth.nrd(rep(dat,weight)).

Value

A list containing the following components:

x
: The coordinates of the grid points on x-direction.

y
: For 2-dim projection, the coordinates of the grid points on y-direction. Non-existing for 1-dim projection.

z
: For 2-dim projection, a length(x) by length(y) martix of the estimated density. For 1-dim projection, a vector of length length(x) of the estimated density values.

Author(s)

Yajie Duan, Javier Cabrera

References

Duan, Y., Cabrera, J. & Emir, B. A New Projection Pursuit Index for Big Data. Under revision.

Cherasia, K. E., Cabrera, J., Fernholz, L. T., & Fernholz, R. (2022). Data Nuggets in Supervised Learning. In Robust and Multivariate Statistical Methods: Festschrift in Honor of David E. Tyler (pp. 429-449). Cham: Springer International Publishing.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

See Also

NHnugg,create.DN, refine.DN

Examples

```
require(datanugget)
require(rstiefel)

#4-dim small example
X = cbind.data.frame(rnorm(5*10^3),
                     rnorm(5*10^3),
                     rnorm(5*10^3),
                     rnorm(5*10^3))


#create data nuggets
my.DN = create.DN(x = X,
                  R = 500,
                  delete.percent = .1,
                  DN.num1 = 500,
                  DN.num2 = 250,
                  no.cores = 0,
                  make.pbs = FALSE)


#refine data nuggets
my.DN2 = refine.DN(x = X,
                   DN = my.DN,
                   EV.tol = .9,
                   min.nugget.size = 2,
                   max.splits = 5,
                   no.cores = 0,
                   make.pbs = FALSE)

#get nugget centers, weights, and scales
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale

#generate a random orthonormal matrix as a projection matrix to 2-dim space
```

```
proj_2d = rustiefel(4, 2)

#project data nugget centers into 2-dim space by the random projection matrix
nuggproj_2d = as.matrix(nugg)%*%proj_2d

#plot the projected data nuggets
#lighter green represents larger weights
plotNugg(nuggproj_2d, weight)

#project raw large data into 2-dim space using the same projection matrix
rawproj_2d = as.matrix(X)%*%proj_2d

#plot projected raw large dataset
plot(rawproj_2d)

#estimated density for 2-dim projected data based on the data nuggets
est_nugg = nuggKDE(nuggproj_2d, weight, scale)
#plot the estimated density values
image(est_nugg)
```

plotLoadings *Plot of variable loadings for 1-dim/2-dim projection*

Description

Draw a loading plot for 1-dim/2-dim projection

Usage

```
plotLoadings(loadings, label = NULL,pch = 16,main = "Loadings",
              xlab = NULL,ylab = NULL, xlim = NULL, ylim = NULL,
            textpoints = loadings, textcex = 1, textpos = 2, textcol = NULL, ...)
```

Arguments

`loadings` Loadings of original variables for projection in 1-dim/2-dim space, one column for each projection direction. Must be a data matrix (of class matrix, or data.frame) with one or two columns or a vector containing only entries of class numeric.

`label` A character vector specifying the text to be written for the variables. Default is NULL, which uses the "V1, V2,..." as the labels of variables. If not NULL, the vector length should be the number of variables, i.e., nrow(loadings).

`pch` For 2-dim projection loadings, either an integer specifying a symbol or a single character to be used as the default in plotting points. See `points` for possible values and their interpretation. Default is 16.

`main` The title for the loading plot.

`xlab` The title for the x axis.

`ylab` The title for the y axis.

`xlim` The scale limits for the x axis.

`ylim` The scale limits for the y axis.

`textpoints` For 2-dim projection loadings, the x and y coordinates of the variable label text displayed on the loading plot. Must be a data matrix (of class matrix, or data.frame) with two columns. Default value is the 2-dim projection loading matrix.

`textcex` For 2-dim projection loadings, numeric character expansion factor for the variable label text displayed on the loading plot. Default is 1. See `text`.

`textpos` For 2-dim projection loadings, a position specifier for the variable label text displayed on the loading plot. Default is 2, indicating the left of the coordinates. See `text`.

`textcol` For 2-dim projection loadings, the color font to be used for the variable label text displayed on the loading plot. See `text`.

`...` other arguments and graphical parameters passed to `plot` for 2-dim projection, or `barplot` for 1-dim projection.

Details

This function draws a loading plot for 1-dim/2-dim projection. It plot a barplot for 1-dim projection loadings and a scatterplot with variable label text for 2-dim.

Value

No return value, called for plotting.

Author(s)

Yajie Duan, Javier Cabrera

References

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

Cherasia, K. E., Cabrera, J., Fernholz, L. T., & Fernholz, R. (2022). Data Nuggets in Supervised Learning. *In Robust and Multivariate Statistical Methods: Festschrift in Honor of David E. Tyler* (pp. 429-449). Cham: Springer International Publishing.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

See Also

`PPnugg`, `faProj`

Examples

```
require(datanugget)
require(rstiefel)

#4-d small example with visualization
X = rbind.data.frame(matrix(rnorm(5*10^3, sd = 0.3), ncol = 4),
          matrix(rnorm(5*10^3, mean = 1, sd = 0.3), ncol = 4))


#create data nuggets
my.DN = create.DN(x = X,
                  R = 500,
                  delete.percent = .1,
                  DN.num1 = 500,
                  DN.num2 = 250,
                  no.cores = 0,
                  make.pbs = FALSE)


#refine data nuggets
my.DN2 = refine.DN(x = X,
                   DN = my.DN,
                   EV.tol = .9,
                   min.nugget.size = 2,
                   max.splits = 5,
                   no.cores = 0,
                   make.pbs = FALSE)

#get nugget centers, weights, and scales
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale


#generate a random projection matrix to 2-dim space
proj_2d = rustiefel(4, 2)

#project data nugget centers into 2-dim space by the random projection matrix
nuggproj_2d = as.matrix(nugg)%*%proj_2d

#plot data nuggets in 2-dim space
plotNugg(nuggproj_2d,weight)

#plot loadings for the variables
plotLoadings(proj_2d)


#generate a random projection vector to 1-dim space
proj_1d = rustiefel(4, 1)

#project data nugget centers into 1-dim space by the random projection vector
nuggproj_1d = as.matrix(nugg)%*%proj_1d
```

```
#plot the weighted histogram for 1-dim projected data nuggets
plotNugg(nuggproj_1d,weight,hist = TRUE)

#plot loadings for the variables
plotLoadings(proj_1d)
```

plotNugg *Plot of projected 1-dim/2-dim data nuggets*

Description

Draw a scatterplot/stripchart/weighted histogram of projected 1-dim/2-dim data nuggets considering the weights of data nuggets.

Usage

```
plotNugg(nuggproj, weight,
         qt = 0.8,pch = 16,cex = 0.5,
         hist = FALSE, jitter = 0.1,freq = TRUE, breaks = 30,
         main = NULL, xlab = NULL, ylab = NULL,...)
```

Arguments

`nuggproj` Projected data nugget centers in 1-dim/2-dim space. Must be a data matrix (of class matrix, or data.frame) with two columns or a vector containing only entries of class numeric.

`weight` Vector of the weight parameter for each data nugget. Its length should be the same as the number of data nuggets, i.e., nrow(nuggproj) for 2-dim projection/length(nuggproj) for 1-dim projection. Must be of class numeric or integer.

`qt` A scalar with value in `[0,1]` indicating the probability used to obtain a sample quantile of the data nugget weights as the maximal value to transform weights to colors for the plot. Defaults to be 0.8.

`pch` Either an integer specifying a symbol or a single character to be used as the default in plotting points. See `points` for possible values and their interpretation.

`cex` A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default. See `par`.

`hist` logical; If TRUE, a weighted histogram is plotted for 1-dim projected data nuggets considering the nugget weights. Otherwise, a stripchart of the points jittered is plotted for 1-dim projection with colors indicating nugget weights. Defalut to be FALSE. Ignorable for 2-dim projection.

`jitter` If `hist = FALSE`(default), the amount of jittering applied to the stripchart of 1-dim projected data nuggets. Ignorable for 2-dim projection and 1-dim projection with `hist = TRUE`.

`freq` If `hist = TRUE`, a logical value indicating whether to plot frequencies. or probability densities for the weighted histogram of 1-dim projected data nuggets. Ignorable for 2-dim projection and 1-dim projection with `hist = FALSE`.

`breaks` If `hist = TRUE`, the `breaks` argument for the weighted histogram of 1-dim projected data nuggets. See details in `wtd.hist`. Ignorable for 2-dim projection and 1-dim projection with `hist = FALSE`.

`main` an overall title for the plot: see `title`.

`xlab` a title for the x axis: see `title`.

`ylab` a title for the y axis: see `title`.

`...` further arguments and graphical parameters passed to `plot` for 2-dim projection, or `stripchart` or `wtd.hist` for 1-dim projection.

Details

This function plots a scatterplot/stripchart/weighted histogram of projected 1-dim/2-dim data nuggets considering the weights of data nuggets.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

Based on the created and refined data nuggets, the projected 1-dim/2-dim data nuggets are plotted via a scattorplot (2d) or a stripchart with points jittered (1d) of data nugget centers, coloring with data nugget weights where lighter green represents larger weights. If `hist = TRUE`, a weighted histgram of 1-dim projected data nugget centers is plotted considering the data nugget weights.

Value

If `hist = TRUE`, an object of class `histogram`. See `wtd.hist`. Otherwise, no returned values.

Author(s)

Yajie Duan, Javier Cabrera

References

Cherasia, K. E., Cabrera, J., Fernholz, L. T., & Fernholz, R. (2022). Data Nuggets in Supervised Learning. *In Robust and Multivariate Statistical Methods: Festschrift in Honor of David E. Tyler* (pp. 429-449). Cham: Springer International Publishing.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

See Also

`datanugget-package`, `create.DN`, `refine.DN`,`wtd.hist`, `stripchart`

Examples

```
require(datanugget)
require(rstiefel)

#2-d small example with visualization
X = rbind.data.frame(matrix(rnorm(10^4, sd = 0.3), ncol = 2),
          matrix(rnorm(10^4, mean = 1, sd = 0.3), ncol = 2))


#create data nuggets
my.DN = create.DN(x = X,
                  R = 500,
                  delete.percent = .1,
                  DN.num1 = 500,
                  DN.num2 = 250,
                  no.cores = 0,
                  make.pbs = FALSE)


#refine data nuggets
my.DN2 = refine.DN(x = X,
                   DN = my.DN,
                   EV.tol = .9,
                   min.nugget.size = 2,
                   max.splits = 5,
                   no.cores = 0,
                   make.pbs = FALSE)

#get nugget centers, weights, and scales
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale

#plot raw large dataset
plot(X)

#plot data nuggets in 2-dim space
plotNugg(nugg,weight)

#generate a random projection vector to 1-dim space
proj_1d = rustiefel(2, 1)

#project data nugget centers into 1-dim space by the random projection vector
nuggproj_1d = as.matrix(nugg)%*%proj_1d

#plot the stripchart for 1-dim projected data nuggets
plotNugg(nuggproj_1d,weight)
#plot the weighted histogram for 1-dim projected data nuggets
plotNugg(nuggproj_1d,weight,hist = TRUE)
```

PPnugg *Projection Pursuit for Big Data based on data nuggets*

Description

This function performs 1-dim/2-dim projection pursuit (PP) for big data based on data nuggets.

Usage

```
PPnugg(data, index = c("NH","Hole","CM"), dim, h = NULL, fa = TRUE, den = TRUE,
       R = 5000, DN.num1 = 10^4, DN.num2 = 2000, max.splits = 5, seed_nugg = 5,
       cooling = 0.9, tempMin = 1e-3, maxiter = 2000, tol = 1e-6, seed_opt = 3,
       initP = NULL,qt = 0.8,label = colnames(data), ...)
```

Arguments

`data` A big data matrix (of class matrix, or data.frame) containing only entries of class numeric. The data size is large in terms of the number of observations, i.e., nrow(data).

`index` A character indicating the PP index function to be optimized: "NH" - Natural Hermite Index for data nuggets "Hole" - Hole Index for data nuggets "CM" - Central Mass Index for data nuggets

`dim` A numerical value indicating the target dimensionality for the projection. It's either 1 or 2.

`h` If `index == "NH"`, a scalar or a length-2 numeric vector indicating the bandwidth used in the calculation of Natural Hermite index for data nuggets. Defaults to NULL, which uses normal reference bandwidth considering the nugget weights. See details in `NHnugg`.

`fa` Logical value indicating whether to perform factor rotation after obtaining the optimal projection. Default is TRUE. See details in `faProj`.

`den` Logical value indicating whether to estimate the density function of the optimal projected data based on data nuggets. Default is TRUE. See details in `nuggKDE`.

`R` The number of observations to sample from the data matrix when creating the initial data nugget centers. Must be of class numeric within [100,10000]. Default is 5000. See details in `create.DN`.

`DN.num1` The number of initial data nugget centers to create. Must be of class numeric. Default is 10^4. See details in `create.DN`.

`DN.num2` The number of data nuggets to create. Must be of class numeric. Default is 2000. See details in `create.DN`.

`max.splits` A numeric value indicating the maximum amount of attempts that will be made to split data nuggets during the refining of data nuggets. Default is 5. See details in `refine.DN`.

`seed_nugg` A numeric value indicating the random seed for replication of data nugget creation and refining. Default is 5. See details in `create.DN`, and `refine.DN`.

`cooling` The cooling factor for optimization of the PP index. Default is 0.9. See details in `PPnuggOptim`.

`tempMin` The minimal temperature parameter to stop the optimization of the PP index. It's a numeric value between (0,1). Default is 1e-3. See details in `PPnuggOptim`.

`maxiter` The maximal number of iterations for optimization of the PP index. Default is 2000. See details in `PPnuggOptim`.

`tol` The tolerance parameter for the PP index value during optimization. Default is 1e-6. See details in `PPnuggOptim`.

`seed_opt` A positive integer value indicating the seed for optimization of the PP index. Default is 3.

`initP` The initial projection matrix to start the optimization of PP index. Must be a data matrix (of class matrix, or data.frame) with the dimension of ncol(data)*dim. Default is NULL, which uses a matrix of orthogonal initialization. See details in `PPnuggOptim`.

`qt` A scalar with value in `[0,1]` used in the plot of projected data nuggets. See details in `plotNugg`.

`label` A character vector specifying the text to be written for the variables on the loading plot. Defaults to the column names of the bia data set. See details in `plotLoadings`.

`...` Other arguments sent to `create.DN`, `refine.DN`, `PPnuggOptim`, `NHnugg`, `faProj`, `plotNugg`, `plotLoadings`, and `nuggKDE`.

Details

This function performs 1-dim/2-dim projection pursuit (PP) for big data based on data nuggets.

Projection Pursuit (PP) is a tool for high-dim data to find low-dim projections indicating hidden structures such as clusters, outliers, and other non-linear structures. The interesting low dimensional projections of high-dimensional data could be found by optimizing PP index function. PP Index function numerically measures features of low-dimensional projections. Higher values of PP indices correspond to more interesting structure, such as point mass, holes, clusters, and other non-linear structures.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

The projection pursuit for big data with a large number of observations could be performed based on data nuggets. Before creating data nuggets for big data, the raw data is recommended to be standardized first for Projection Pursuit, which is performed by `scale` in the function. After obtaining created and refined data nuggets for big data, data nugget centers needs to be spherized considering nugget weights before conducting projection pursuit. The optimal or interested projection found by projection pursuit would be taken on the spherized nugget centers. The optimization of PP index is performed by `PPnuggOptim` for 1-dim/2-dim projection using grand tour simulated annealing method. The PP index for data nuggets could be Natural Hermite index, Hole index and CM index. See details in `NHnugg`, `HoleNugg`, and `CMNugg`.

After obtaining the optimal 1-dim/2-dim projection by maximizing PP index for data nuggets, a factor analysis could be performed on the projection to back to the original variables. See details in faProj. The rotation is taken on the overall transformation matrix for the raw data nuggets, which is a combination of spherization matrix and optimial projection matrix, to back to the original variables.

The density of projected data can be estimated, which is performed by nuggKDE in the function. Moreover, the same centering, spherization and the optimal projection found for the data nuggets is also performed on the standardized raw data to output the projected raw big data found by Projection Pursuit based on data nuggets, i.e., dataproj. Based on the 1-dim/2-dim projection found, the hidden structures such as clusters, outliers, and other non-linear structures inside the big data set could be explored.

Value

A list containing the following components:

DN An object of class datanugget obtained from the standardized raw big data after data nugget creation and refining. It contains a data frame including data nugget centers, weights, and scales, and a vector of length nrow(data) indicating the data nugget assignments of each observation in the raw data.

nuggproj If fa == TRUE, the rotated projected data nugget centers after conducting factor ratation on the optimal projection found. If fa == FALSE, the projected data nugget centers under the optimal projection found.

loadings A matrix of loadings for original variables, one column for each projection direction. If fa == TRUE, it's the rotated transformation matrix to obtain the projected data nugget centers after factor rotation. If fa == FALSE, it's an overall transformation matrix for the centered data nuggets, which is a combination of spherization matrix and the optimal projection matrix found, to back to the original variables. In either case, nuggproj can be obtained by multiplying the centered data nuggets nugg_wcen with this loading matrix loadings.

nugg_wcen The centered data nugget centers that has a zero weighted mean for each column considering nugget weights. It's obtained by extracting the weighted mean from the original data nugget centers.

dataproj The corresponding projected raw data. It's obtained by performing the same projection found on the standardized raw big data, i.e., dataproj = data_cen %*% loadings.

data_cen The standardized raw data centered by the weighted mean of data nugget centers.

index.opt The optimal PP index value found.

density If den == TRUE, a list containing the estimated density for projected data. See details in nuggKDE.

Author(s)

Yajie Duan, Javier Cabrera

References

Cook, D., Buja, A., & Cabrera, J. (1993). Projection pursuit indexes based on orthonormal function expansions. Journal of Computational and Graphical Statistics, 2(3), 225-250.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

Cabrera, J., & McDougall, A. (2002). Statistical consulting. Springer Science & Business Media.

Horst, P. (1965). Factor Analysis of Data Matrices. Holt, Rinehart and Winston. Chapter 10.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. Psychometrika, 23(3), 187-200.

See Also

`PPnuggOptim`, `NHnugg`, `HoleNugg`, `create.DN`, `refine.DN`

Examples

```
require(datanugget)

#4-dim small example with cluster stuctures in V3 and V4
X = cbind.data.frame(V1 = rnorm(5*10^4,mean = 5,sd = 2),
                     V2 = rnorm(5*10^4,mean = 5,sd = 1),
                     V3 = c(rnorm(3*10^4,sd = 0.3),
                            rnorm(2*10^4,mean = 2, sd = 0.3)),
                     V4 = c(rnorm(1*10^4,mean = -8, sd = 1),
                            rnorm(3*10^4,mean = 0,sd = 1),
                            rnorm(1*10^4,mean = 7, sd = 1.5)))


#perform 2-dim Projection Pursuit for the big data
#based on Hole index for data nuggets
res = PPnugg(X, index = "Hole", dim = 2, R = 5000, DN.num1 = 1*10^4, DN.num2 = 2000,
no.cores = 2, tempMin = 0.05, maxiter = 1000, tol = 1e-4)

#data nuggets created and refined from the standardized raw data
nugg = res$DN$`Data Nuggets`

#data nugget assignments of each observation in the raw data
nugg_assign = res$DN$`Data Nugget Assignments`

#plot projected data nuggets
plotNugg(res$nuggproj,nugg$Weight,qt = 0.8)

#plot the corresponding projected raw big data
plot(res$dataproj,cex = 0.5,main = "Projected Raw Data")
```

```
#plot the estimated density of the projected data
image(res$density)

#plot loadings of original variables
#V3 and V4 have large loadings, same as the simulation setting.
plotLoadings(res$loadings)



#perform 1-dim Projection Pursuit for the big data
#based on Natural Hermite index for data nuggets
res = PPnugg(X, index = "NH", dim = 1, R = 5000, DN.num1 = 1*10^4, DN.num2 = 2000,
no.cores = 2, tempMin = 0.05, maxiter = 1000, tol = 1e-5)

#data nuggets created and refined from the standardized raw data
nugg = res$DN$`Data Nuggets`

#data nugget assignments of each observation in the raw data
nugg_assign = res$DN$`Data Nugget Assignments`

#plot projected data nuggets
plotNugg(res$nuggproj,nugg$Weight,qt = 0.8,hist = TRUE)

#plot the corresponding projected raw big data
hist(res$dataproj,breaks = 100)
```

PPnuggOptim *Optimize Projection Pursuit index for Big Data based on data nuggets*

Description

Optimize PP index for 1-dim/2-dim projection for big data based on data nuggets, using grand tour simulated annealing optimizationg method.

Usage

```
PPnuggOptim(FUN, nugg_wsph, dimproj, tempInit = 1, cooling = 0.9, eps = 1e-3,
            tempMin = 0.01, maxiter = 1000, half = 10, tol = 1e-5, maxc = 15,
            seed = 3, initP = NULL, ...)
```

Arguments

`FUN` The index function for data nuggets to optimize.

`nugg_wsph` The data nugget centers spherized with nugget weights. Must be a data matrix (of class matrix, or data.frame) with at least two columns. See `wsph` for spherization with nugget weights.

`dimproj` A numerical value indicating the dimension of the data projection. It's either 1 or 2.

`tempInit` The initial temperature parameter for optimization, i.e., grand tour simulated annealing method. It's a numeric value between (0,1).

`cooling` The cooling factor for optimization. Rate by which the temperature is reduced from one cycle to the next, i.e., the new temperature = cooling * old temperature. Default is 0.9.

`eps` The approximation accuracy for cooling used in the interpolation between current projection and the target one. Default is 1e-3.

`tempMin` The minimal temperature parameter to stop the optimization. It's a numeric value between (0,1). Default is 0.01.

`maxiter` The maximal number of iterations for the optimization. Default is 1000.

`half` The number of steps without incrementing the index before decreasing the temperature parameter by multiplying the cooling factor. Default is 10.

`tol` The tolerance parameter for the index value during optimization. The increase of index value smaller than the tolerance would be ignored. Default is 1e-5.

`maxc` The maximal number of temperature changes without incrementing the index value. The algorithm would stop if the number of temperature changes without index value increasing exceeds maxc paramter. Default is 15.

`seed` A positive integer value indicating the seed for the optimization. Default is 3.

`initP` The initial projection matrix to start the optimization. Must be a data matrix (of class matrix, or data.frame) with the dimension of ncol(nugg_wsph)*dimproj. Default is NULL, which uses a matrix of orthogonal initialization.

`...` Other arguments passed to the index function FUN.

Details

This function performs the optimization PP index for 1-dim/2-dim projection for big data based on data nuggets, using grand tour simulated annealing optimizationg method.

Data nuggets are a representative sample meant to summarize Big Data by reducing a large dataset to a much smaller dataset by eliminating redundant points while also preserving the peripheries of the dataset. Each data nugget is defined by a center (location), weight (importance), and scale (internal variability). Data nuggets for a large dataset could be created and refined by functions `create.DN` or `refine.DN` in the package `datanugget`.

After obtaining created and refined data nuggets for big data, data nugget centers needs to be spherized considering nugget weights before conducting projection pursuit. The optimal or most interested projection could be found by optimization of the projection pursuit index based on data nuggets. This function optimizes the PP index function for data nuggets by GTSA, i.e., the grand tour simulated annealing optimizationg method. The optimization starts with an initial projection and a initial temperature parameter. A target projection would be generated from the current projection plus the temperature times the initial base, from which a projection matrix is generated through interpolation between current and target projections. If the number of steps without incrementing the index exceeds the parameter half, the temperature parameter is decreased by multiplying the cooling factor. The increase of index value smaller than the tolerance would be ignored. The optimization would stop if either there are a maximal number of iterations, or the temperature has decreased to the minimal value, or there are a maximal number of temperature changes without incrementing the index value.

Value

A list containing the following components:

`proj.nugg` The projected data nugget centers under the optimal projection found.

`proj.opt` The optimal projection matrix found.

`index` A vector with the PP index values found in the optimization process.

Author(s)

Yajie Duan, Javier Cabrera

References

Cook, D., Buja, A., & Cabrera, J. (1993). Projection pursuit indexes based on orthonormal function expansions. Journal of Computational and Graphical Statistics, 2(3), 225-250.

Beavers, T. E., Cheng, G., Duan, Y., Cabrera, J., Lubomirski, M., Amaratunga, D., & Teigler, J. E. (2024). Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure. Journal of Computational and Graphical Statistics, (just-accepted), 1-21.

Duan, Y., Cabrera, J., & Emir, B. (2023). A New Projection Pursuit Index for Big Data. ArXiv:2312.06465. https://doi.org/10.48550/arXiv.2312.06465

See Also

`PPnugg`, `NHnugg`,`create.DN`, `refine.DN`

Examples

```
require(datanugget)

#4-dim small example with cluster stuctures in V3 and V4
X = cbind.data.frame(V1 = rnorm(5*10^3,mean = 5,sd = 2),
                     V2 = rnorm(5*10^3,mean = 5,sd = 1),
                     V3 = c(rnorm(3*10^3,sd = 0.3),
                            rnorm(2*10^3,mean = 2, sd = 0.3)),
                     V4 = c(rnorm(1*10^3,mean = -8, sd = 1),
                            rnorm(3*10^3,mean = 0,sd = 1),
                            rnorm(1*10^3,mean = 7, sd = 1.5)))

#raw data is recommended to be scaled firstly to generate data nuggets for Projection Pursuit
X = as.data.frame(scale(X))

#create data nuggets
my.DN = create.DN(x = X,
                  R = 500,
                   delete.percent = .1,
                   DN.num1 = 500,
                   DN.num2 = 250,
                   no.cores = 2,
                   make.pbs = FALSE)


#refine data nuggets
my.DN2 = refine.DN(x = X,
                   DN = my.DN,
                   EV.tol = .9,
                   min.nugget.size = 2,
                   max.splits = 5,
                   no.cores = 2,
                   make.pbs = FALSE)

#get nugget centers, weights, and scales
nugg = my.DN2$`Data Nuggets`[,2:(ncol(X)+1)]
weight = my.DN2$`Data Nuggets`$Weight
scale = my.DN2$`Data Nuggets`$Scale

#spherize data nugget centers considering weights to conduct Projection Pursuit
wsph.res = wsph(nugg,weight)
nugg_wsph = wsph.res$data_wsph
wsph_proj = wsph.res$wsph_proj

#conduct Projection Pursuit in 2-dim by optimizing Natural Hermite index
res = PPnuggOptim(NHnugg, nugg_wsph, dimproj = 2, weight = weight, scale = scale,
      tempMin = 0.05, maxiter = 1000, tol = 1e-5)

#plot projected data nuggets
plotNugg(nugg_wsph%*%res$proj.opt,weight,qt = 0.8)


#conduct Projection Pursuit in 1-dim by optimizing Hole index
res = PPnuggOptim(HoleNugg, nugg_wsph, dimproj = 1, weight = weight,
      tempMin = 0.05, maxiter = 1000, tol = 1e-5)

#plot projected data nuggets
plotNugg(nugg_wsph%*%res$proj.opt,weight)
```

wsph *Spherize/Whiten data with observational weights*

Description

This function performs PCA sphering/whitening transformation on data with observational weights.

Usage

```
wsph(data,weight)
```

Arguments

`data`	A data matrix (of class matrix, or data.frame) containing only entries of class numeric.
`weight`	Vector of length nrow(data) of weights for each observation in the dataset. Must be of class numeric or integer or table. If NULL, the default value is a vector of 1 with length nrow(data), i.e., weights equal 1 for all observations.

Details

This function performs PCA sphering/whitening transformation on data with observational weights. Specifically, weighted sample mean and weighted sample covariance matrix are firstly calculated. Next, data are centered with weights, and spectral decomposition of the weighted covariance matrix is conducted to obtain its eigenvalues and eigenvectors. Based on them, the PCA sphering/whitening transformation is performed to obtain a spherized data matrix considering the observational weights. The spherized data matrix has a zero weighted mean for each column, and a weighted covariance that equals identity matrix.

Value

A list containing the following components:

`data_wsph`	The spherized data matrix that has a zero weighted mean for each column, and a weighted covariance that equals identity matrix.
`data_wcen`	The centered data matrix that has a zero weighted mean for each column. It's obtained by extracting the weighted sample mean from the original data matrix.
`wmean`	Vector of length ncol(data). It's the weighted sample mean of the original data matrix.
`wcov`	Matrix of size ncol(data) by ncol(data). It's the weighted sample covariance matrix of the original data matrix.
`wsph_proj`	Matrix of size ncol(data) by ncol(data). It's the sphering/whitening matrix for the transformation. The spherized data matrix `data_wsph` is obtained by multiplying the centered data matrix with weights `data_wcen` with this sphering/whitening matrix.

Author(s)

Yajie Duan, Javier Cabrera

References

Cabrera, J., & McDougall, A. (2002). Statistical consulting. Springer Science & Business Media.

Examples

```
dataset = matrix(rnorm(300),100,3)

# assign random weights to observations
weight = sample(1:20,100,replace = TRUE)

# spherize the dataset with observational weights
res = wsph(dataset,weight)

# spherized data matrix considering the observation weights
res$data_wsph

# The spherzied data matrix has a zero weighted sample mean for each column
1/sum(weight)*t(as.matrix(weight))%*%as.matrix(res$data_wsph)

# The spherzied data matrix has a weighted covariance that equals identity matrix
1/sum(weight)*t(as.matrix(res$data_wsph))%*%diag(weight)%*%as.matrix(res$data_wsph)
```

Index