Package 'datanugget'

January 24, 2020

Type Package

Title Create, Refine, and Cluster Data Nuggets

Version 1.0.0

Date 2020-1-16

Author Traymon Beavers [aut, cre],
Javier Cabrera [aut],
Mariusz Lubomirski [aut]

Maintainer Traymon Beavers <tray.beavers@gmail.com>

Description Creating, refining, and clustering data nuggets.
Data nuggets reduce a large dataset into a small collection of nuggets of data, each containing a center (location), weight (importance), and scale (variability) parameter. Data nugget centers are created by choosing observations in the dataset which are as equally spaced apart as possible. Data nugget weights are created by counting the number observations closest to a given data nugget's center. We then say the data nugget 'contains' these observations and the data nugget center is recalculated as the mean of these observations. Data nugget scales are created by calculating the trace of the covariance matrix of the observations contained within a data nugget divided by the dimension of the dataset. Data nuggets are refined by 'splitting' data nuggets which have scales or shapes (defined as the ratio of the two largest eigenvalues of the covariance matrix of the observations contained within the data nugget) deemed too large. Data nuggets are clustered by using a weighted form of k-means clustering which uses both the centers and weights of data nuggets to optimize the clustering assignments.

Depends R (>= 3.5.0), doSNOW (>= 1.0.16), foreach (>= 1.4.4), parallel (>= 3.5.0)

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-24 17:30:08 UTC

R topics documented:

`datanugget-package` *Data Nuggets*

Description

This package contains functions to create, refine, and cluster data nuggets which serve as representative samples of large datasets. The functions which perform these processes are create.DN, refine.DN, and WKmeans, respectively.

Author(s)

Traymon Beavers, Javier Cabrera, Mariusz Lubomirski

References

Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure (Submitted for Publication, 2019)

`create.DN` *Create Data Nuggets*

Description

This function draws a random sample of observations from a large dataset and creates data nuggets, a type of representative sample of the dataset, using a specified distance metric.

Usage

```
create.DN(x,
          RS.num = 2.5*(10^5),
          DN.num1 = 10^4,
          DN.num2 = 2000,
          dist.metric = "euclidean",
          seed = 291102,
          no.cores = (detectCores() - 1),
          make.pbs = TRUE)
```

Arguments

x	A data matrix (of class matrix, data.frame, or data.table) containing only entries of class numeric.
RS.num	The number of observations to sample from the data matrix. Must be of class numeric.
DN.num1	The number of initial data nugget centers to create. Must be of class numeric.
DN.num2	The number of data nuggets to create. Must be of class numeric.
dist.metric	The distance metric used to create the initial centers of data nuggets. Must be 'euclidean' or 'manhattan'.
seed	Random seed for replication. Must be of class numeric.
no.cores	Number of cores used for parallel processing. If '0' then parallel processing is not used. Must be of class numeric.
make.pbs	Print progress bars? Must be TRUE or FALSE.

Details

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). This function creates data nuggets using Algorithm 1 provided in the reference.

Value

An object of class datanugget:

Data Nuggets	DN.num by (ncol(x)+3) data frame containing the information for the data nuggets created (index, center, weight, scale).
Data Nugget Assignments	Vector of length nrow(x) containing the data nugget assignment of each observation in x.

Author(s)

Traymon Beavers, Javier Cabrera, Mariusz Lubomirski

References

Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure (Submitted for Publication, 2019)

Examples

```
## small example
X = cbind.data.frame(rnorm(10^4),
                     rnorm(10^4),
                     rnorm(10^4))

suppressMessages({

  my.DN = create.DN(x = X,
                    RS.num = 10^3,
                    DN.num1 = 500,
                    DN.num2 = 250,
                    no.cores = 0,
                    make.pbs = FALSE)

})

my.DN$`Data Nuggets`
my.DN$`Data Nugget Assignments`



## large example
X = cbind.data.frame(rnorm(10^6),
                     rnorm(10^6),
                     rnorm(10^6),
                     rnorm(10^6),
                     rnorm(10^6))

my.DN = create.DN(x = X,
                  RS.num = 10^5,
                  DN.num1 = 10^4,
                  DN.num2 = 2000)

my.DN$`Data Nuggets`
my.DN$`Data Nugget Assignments`
```

`create.DNcenters` *Create Data Nugget Centers*

Description

This function creates the centers of data nuggets from a random sample.

Usage

```
create.DNcenters(RS,
                 DN.num,
                 dist.metric,
                 make.pb = FALSE)
```

Arguments

RS	A data matrix (data frame, data table, matrix, etc) containing only entries of class numeric.
DN.num	The number of data nuggets to create. Must be of class numeric.
dist.metric	The distance metric used to create the initial centers of data nuggets. Must be 'euclidean' or 'manhattan'.
make.pb	Print progress bar? Must be TRUE or FALSE.

Details

This function is used for reducing a random sample to data nugget centers in the create.DN function. NOTE THAT THIS FUNCTION IS NOT DESIGNED FOR USE OUTSIDE OF THE create.DN FUNCTION.

Value

DN.data	DN.num by (ncol(RS)) data frame containing the data nugget centers.

Author(s)

Traymon Beavers, Javier Cabrera, Mariusz Lubomirski

References

Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure (Submitted for Publication, 2019)

refine.DN *Refine Data Nuggets*

Description

This function refines the data nuggets found in an object of class datanugget created using the create.DN function.

Usage

```
refine.DN(x,
          DN,
          scale.tol = .9,
          shape.tol = .9,
          min.nugget.size = 2,
          max.nuggets = 10000,
          scale.max.splits = 5,
          shape.max.splits = 5,
          seed = 291102,
          no.cores = (detectCores() - 1),
          make.pbs = TRUE)
```

Arguments

x	A data matrix (data frame, data table, matrix, etc.) containing only entries of class numeric.
DN	An object of class data nugget created using the create.DN function.
scale.tol	A value designating the percentile for finding the corresponding quantile that will designate how large the data nugget scales can be before it must be split. Must be of class numeric and within (0,1).
shape.tol	A value designating the percentile for finding the corresponding quantile that will designate how large the ratio of the two largest eigenvalues of the covariance matrix of a data nugget can be before it must be split. Must be of class numeric and within (0,1).
min.nugget.size	A value designating the minimum amount of observations a data nugget created from a split must contain. Must be of class numeric and greater than 1.
max.nuggets	A value designating the maximum amount of data nuggets that will be created before the algorithm breaks. Must be of class numeric and greater than the number of data nuggets in argument DN.
scale.max.splits	A value designating the maximum amount of attempts that will be made to split data nuggets according to their scale before the algorithm breaks. Must be of class numeric.

`shape.max.splits`
: A value designating the maximum amount of attempts that will be made to split data nuggets according to their shape before the algorithm breaks. Must be of class numeric.

`seed`
: Random seed for replication. Must be of class numeric.

`no.cores`
: Number of cores used for parallel processing. If '0' then parallel processing is not used. Must be of class numeric.

`make.pbs`
: Print progress bars? Must be TRUE or FALSE.

Details

Data nuggets can be refined by attempting to make all of the data nugget scales as small as possible and their shapes as spherical as possible. This is achieved by designating a scale tolerance (scale.tol) and a shape tolerance (shape.tol) which is used to give a lower threshold for a data nugget's scale and deviation from sphericity, respectively.

If a data nugget has a scale greater than the quantile associated with the percentile given by scale.tol, this data nugget is split into two smaller data nuggets using K-means clustering. Likewise, if the two largest eigenvalues of a data nugget's covariance matrix have a ratio greater than the quantile associated with the percentile given by shape.tol, this data nugget is split into two smaller data nuggets using K-means clustering.

However, if either of the two data nuggets created by this split have less than the designated minimum data nugget size (min.nugget.size), then the split is cancelled and the data nugget remains as is. This function refines data nuggets using Algorithm 2 provided in the reference.

Value

An object of class datanugget:

`Data Nuggets`
: DN.num by (ncol(x)+3) data frame containing the information for the data nuggets created (index, center, weight, scale).

`Data Nugget Assignments`
: Vector of length nrow(x) containing the data nugget assignment of each observation in x.

Author(s)

Traymon Beavers, Javier Cabrera, Mariusz Lubomirski

References

Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure (Submitted for Publication, 2019)

Examples

```
## small example
X = cbind.data.frame(rnorm(10^4),
```

```
                    rnorm(10^4),
                    rnorm(10^4))

suppressMessages({

  my.DN = create.DN(x = X,
                    RS.num = 10^3,
                    DN.num1 = 500,
                    DN.num2 = 250,
                    no.cores = 0,
                    make.pbs = FALSE)

  my.DN2 = refine.DN(x = X,
                     DN = my.DN,
                     scale.tol = .9,
                     shape.tol = .9,
                     min.nugget.size = 2,
                     max.nuggets = 1000,
                     scale.max.splits = 5,
                     shape.max.splits = 5,
                     no.cores = 0,
                     make.pbs = FALSE)

})

my.DN2$`Data Nuggets`
my.DN2$`Data Nugget Assignments`



  ## large example
  X = cbind.data.frame(rnorm(10^6),
                       rnorm(10^6),
                       rnorm(10^6),
                       rnorm(10^6),
                       rnorm(10^6))

  my.DN = create.DN(x = X,
                    RS.num = 10^5,
                    DN.num1 = 10^4,
                    DN.num2 = 2000)

  my.DN$`Data Nuggets`
  my.DN$`Data Nugget Assignments`

  my.DN2 = refine.DN(x = X,
                DN = my.DN,
                scale.tol = .9,
                shape.tol = .9,
                min.nugget.size = 2,
                max.nuggets = 10000,
                scale.max.splits = 5,
                shape.max.splits = 5)
```

```
my.DN2$`Data Nuggets`
my.DN2$`Data Nugget Assignments`
```

WKmeans *Weighted K-means Clustering of Data Nuggets*

Description

This function clusters data nuggets using a form of weighted K-means clustering.

Usage

```
WKmeans(dataset,
        k,
        cl.centers = NULL,
        obs.weights,
        num.init = 1,
        max.iterations = 10,
        print.progress = TRUE,
        seed = 291102,
        reassign.prop = .25)
```

Arguments

dataset	A data matrix (data frame, data table, matrix, etc) containing only entries of class numeric (i.e. matrix of data nugget centers).
k	Number of desired clusters. Must be of class numeric.
cl.centers	Chosen cluster centers. If not NULL, must be a k by ncol(dataset) matrix containing only entries of class numeric.
obs.weights	Vector of length nrow(dataset) of weights for each observation in the dataset. Must be of class numeric.
num.init	Number of initial clusters to attempt. Ignored if cl.centers is not NULL. Must be of class numeric.
max.iterations	Maximum number of iterations attempted for convergence before quitting. Must be of class numeric.
print.progress	Print progress of algorithm? Must be TRUE or FALSE.
seed	Random seed for replication. Must be of class numeric.
reassign.prop	Proportion of data to attempt to reassign during each iteration. Must be of class numeric and within (0,1].

Details

Weighted K-means clustering can be used as an unsupervised learning technique to cluster observations contained in datasets that also have a measure of importance (e.g. weight) associated with them. In the case of data nuggets, this is the weight parameter associated with the data nuggets, so the centers of data nuggets are clustered using their weight parameters. The objective of the algorithm which performs this method of clustering is to minimize the weighted within cluster sum of squares (WWCSS). This function clusters data nuggets using Algorithm 3 provided in the reference.

Note that although this method was designed for use with data nuggets, there is no obvious reason to suggest that it cannot be used to perform clustering for other datasets which also have some weighting scheme.

Value

`Cluster Assignments`
: Vector of length nrow(dataset) containing the cluster assignment for each observation.

`Cluster Centers`
: k by ncol(dataset) matrix containing the cluster centers for each cluster.

`Weighted WCSS`
: List containing the individual WWCSS for each cluster and the combined sum of all individual WWCSS's.

Author(s)

Traymon Beavers, Javier Cabrera, Mariusz Lubomirski

References

Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure (Submitted for Publication, 2019)

Examples

```
## small example
X = cbind.data.frame(rnorm(10^4),
                     rnorm(10^4),
                     rnorm(10^4))

suppressMessages({

  my.DN = create.DN(x = X,
                    RS.num = 10^3,
                    DN.num1 = 500,
                    DN.num2 = 250,
                    no.cores = 0,
                    make.pbs = FALSE)

  my.DN2 = refine.DN(x = X,
                     DN = my.DN,
                     scale.tol = .9,
```

```
                      shape.tol = .9,
                      min.nugget.size = 2,
                      max.nuggets = 1000,
                      scale.max.splits = 5,
                      shape.max.splits = 5,
                      no.cores = 0,
                      make.pbs = FALSE)


  DN.clus = WKmeans(dataset = my.DN2$`Data Nuggets`[, c("Center1",
                                                        "Center2",
                                                        "Center3")],
                    k = 3,
                    obs.weights = my.DN2$`Data Nuggets`[, "Weight"],
                    num.init = 1,
                    max.iterations = 3,
                    reassign.prop = .33,
                    print.progress = FALSE)

})

DN.clus$`Cluster Assignments`
DN.clus$`Cluster Centers`
DN.clus$`Weighted WCSS`



  ## large example
  X = cbind.data.frame(rnorm(10^6),
                       rnorm(10^6),
                       rnorm(10^6),
                       rnorm(10^6),
                       rnorm(10^6))

  my.DN = create.DN(x = X,
                    RS.num = 10^5,
                    DN.num1 = 10^4,
                    DN.num2 = 2000)

  my.DN$`Data Nuggets`
  my.DN$`Data Nugget Assignments`

  my.DN2 = refine.DN(x = X,
                     DN = my.DN,
                     scale.tol = .9,
                     shape.tol = .9,
                     min.nugget.size = 2,
                     max.nuggets = 10000,
                     scale.max.splits = 5,
                     shape.max.splits = 5)

  my.DN2$`Data Nuggets`
  my.DN2$`Data Nugget Assignments`
```

```
DN.clus = WKmeans(dataset = my.DN2$`Data Nuggets`[, c("Center1",
                                                      "Center2",
                                                      "Center3")],
                  k = 3,
                  obs.weights = my.DN2$`Data Nuggets`[, "Weight"],
                  num.init = 1,
                  max.iterations = 3,
                  reassign.prop = .33)

DN.clus$`Cluster Assignments`
DN.clus$`Cluster Centers`
DN.clus$`Weighted WCSS`
```

WWCSS *Weighted Within Cluster Sum of Squares*

Description

This function computes the weighted within cluster sum of squares (WWCSS) for a set of cluster assignments provided to a dataset with some weighting scheme.

Usage

```
WWCSS(x,
      k,
      P)
```

Arguments

x	A data matrix (data frame, data table, matrix, etc) containing only entries of class numeric (i.e. matrix of data nugget centers). Must contain the location, weight, and cluster assignment for each observation.
k	The number of possible clusters. Must be of class numeric.
P	The number of columns from the original dataset before clustering and without weight variable. Must be of class numeric.

Details

The WWCSS is used for optimizing the cluster assignments in the WKmeans function. NOTE THAT THIS FUNCTION IS NOT DESIGNED FOR USE OUTSIDE OF THE WKmeans FUNCTION.

Value

`output`	Vector of individual WWCSS's for each cluster
`sum.output`	Combined sum of all individual WWCSS's.

Author(s)

Traymon Beavers, Javier Cabrera, Mariusz Lubomirski

References

Data Nuggets: A Method for Reducing Big Data While Preserving Data Structure (Submitted for Publication, 2019)

Index