Package 'OTAcluster'

May 9, 2018

Title Compute Mean Partition and Assess Uncertainty for Clustering Analysis

Version 1.0.0

Author Jia Li [aut],
Beomseok Seo [aut, cre],
Lin Lin [aut]

Maintainer Beomseok Seo <bzs32@psu.edu>

Description Compute mean partition for clustering ensemble by optimal transport alignment(OTA) and provide uncertainty measures for a partition-wise or a cluster-wise assessment, including topological relationships between clustering ensemble, confidence point set(CPS) and cluster alignment and points based(CAP) separability.

Depends R (>= 3.4.1)

License GPL (>= 2)

LazyData true

RoxygenNote 6.0.1.9000

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

OTAcluster-package *Mean Partition and Uncertainty Assessment of Clustering Analysis by Optimal Transport Alignment(OTA)*

Description

OTAcluster is an R package for computing mean partition of ensemble of partitions by optimal transport alignment (OTA) and both partition-wise and cluster-wise uncertainty measures, including topological relationship between clusters in a partition, Confidence Point Set (CPS), Cluster Alignment and Points based (CAP) separability, and Wasserstein distance between partitions. For a quick introduction to OTAcluster see the vignette A quick tour of **OTAcluster**.

Author(s)

Maintainer: Beomseok Seo <bzs32@psu.edu>

Authors:

- Jia Li <jiali@psu.edu>
- Lin Lin <llin@psu.edu>

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

align.control *Align partitions to the reference partition.*

Description

This function aligns partitions to the reference partition by optimal transport alignment(OTA).

Usage

```
align.control(Zb.stack, nb, threshold = 0.8, alpha = 0.1, split = F,
  folder = "align")
```

Arguments

Zb.stack	Stacked cluster label vector. Reference cluster should be on top of ensemble or partitions.
nb	The number of ensemble partitions in Zb.stack.
threshold	Default is 0.8. Threshold for matching statistic. This percentage of overlapping between two clusters is regarded as matching.
alpha	Default is 0.1. Confident level for confidence point set.
split	Logical. If it is TRUE, split is counted as match in confidence set calculation.
folder	Default is "align". The folder name where temporary result files are created.

Value

List of mean partition(repre), distance of mean partition to each partition in ensemble(dist), output(res), weight matrix(wt), topological stability statistics(summary), confidence point set(confset), cluster alignment and point-based(CAP) separability(jaccard_mat).

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

---

align2 *Median partition by Optimal Transport Alignment(OTA)*

Description

This function searches the index of the median partition in ensemble of partitions by optimal transport alignment(OTA).

Usage

```
align2(Zb.stack, nb, folder = "align")
```

Arguments

Zb.stack	Stacked cluster label vector (n times nb length).
nb	The number of ensemble partitions in Zb.stack.
folder	Default is "align". The folder name where temporary result files are created.

Value

Median partition index in ensemble.

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

GenBsSamps *Generates bootstrap samples*

Description

This function generates ensemble of partitions by bootstrap resampling.

Usage

```
GenBsSamps(X, nb, method, stack = T)
```

Arguments

X	Original data matrix.
nb	The number of ensemble partitions (bootstrap samples) to be generated.
method	Baseline clustering methods. User specified function or example methods included in package ("kmeans", "Mclust", "hclust", "dbscan", "PCAreduce", "HMM-VB") can be used. Refer to the Detail.
stack	Default is TRUE. The format of the output ensemble partition. If it is TRUE, a stacked vector is returned. If not, a matrix is returned.

Value

List of ensemble of partitions and detected number of clusters for each partition.

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

GenData *Generates multi-dimensional random partition.*

Description

This function generates multi-dimensional Gaussian random partition for simulation study.

Usage

```
GenData(n, p, C, k = 0, s = 5, dg = 5)
```

Arguments

n	Size of observations.
p	Dimension of data.
C	The number of clusters.
k	Default is 0. The number of small clusters. Proportion of sum of small clusters is set to have 10% of data.
s	Positive number. Default is 5. A partition is generated in (-s,s) p-cells.
dg	Positive number. Default is 5*p. Degree of inverse Wishart distribution, which determine the separability of clusters.

Value

List of n by p data coordinates matrix (X) and n length cluster label vector (m).

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
```

```
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

GenNoiseSamps *Generates perturbed samples by adding noise.*

Description

This function generates ensemble of partitions by adding Gaussian noise.

Usage

```
GenNoiseSamps(X, nb, method, stack = T, pair.dist = NULL)
```

Arguments

X	Original data matrix.
nb	The number of ensemble partitions (perturbed samples) to be generated.
method	Baseline clustering methods. User specified function or example methods included in package ("kmeans", "Mclust", "hclust", "dbscan", "PCAreduce", "HMM-VB") can be used. Refer to the Detail.
stack	Default is TRUE. The format of the output ensemble partition. If it is TRUE, a stacked vector is returned. If not, a matrix is returned.

Value

List of ensemble of partitions and detected number of clusters for each partition.

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
```

```
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

MeanPart *Align ensemble of partitions to the mean partition.*

Description

This function aligns ensemble of partitions to the reference partition and return mean partition and its stability measures.

Usage

```
MeanPart(Zb.stack, nb, idx, threshold = 0.8, alpha = 0.1)
```

Arguments

Zb.stack	Stacked cluster label vector (n times nb length).
nb	The number of ensemble partitions in Zb.stack.
idx	Index of reference partition in ensemble. (idx must be in [1,nb])
threshold	Default is 0.8. Threshold for matching statistic. This percentage of overlapping between two clusters is regarded as matching.
alpha	Default is 0.1. Confident level for confidence point set.

Value

List of mean partition(repre), distance of mean partition to each partition in ensemble(dist), output(res), weight matrix(wt), topological stability statistics(summary), confidence point set(confset), cluster alignment and point-based(CAP) separability(jaccard_mat).

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

plot_part *Visulize a partition on 2 dimensional space*

Description

This function plots a partition on 2 dimensional reduced space.

Usage

```
plot_part(X, K, title = "", xlab = "", ylab = "")
```

Arguments

X	Cordinates matrix of data.
K	Cluster labels of data.

Value

none

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

plot_part2 *Visulize a partition on 2 dimensional space with convex hull*

Description

This function plots a partition on 2 dimensional reduced space with convex hull.

Usage

```
plot_part2(X, K, title = "", xlab = "", ylab = "", legend.title = "")
```

Arguments

X	Cordinates matrix of data.
K	Cluster labels of data.

Value

none

Examples

```
# set OTA package directory and compile C code.
SetOTA(directory)
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

SetOTA *Preset OTA package*

Description

This function compiles C codes required to run OTA alignment.

Usage

```
SetOTA(directory)
```

Arguments

directory	The address of directory for OTA package C files.

Value

None.

Examples

```
SetOTA(directory)
```

WassDist *Wasserstein distance between partitions.*

Description

This function calculate Wasserstein distance between reference partition and ensemble of partitions.

Usage

```
WassDist(Zb.stack, nb)
```

Arguments

Zb.stack	Stacked cluster label vector. Reference cluster should be on top of ensemble or partitions. Wasserstein distances of ensemble of partitions to the reference partition is returned.
nb	The number of ensemble partitions in Zb.stack.

Value

Matrix of number of clusters and Wasserstein distances of ensemble of partitions to the reference partition.

Examples

```
# set OTA package directory and compile C code.
SetOTA("/Users/BSS/Documents/7.Research/Cluster_Stability_2/r-package/package5")
set.seed(2070)
dat = GenData(n=5000,p=2,C=4)
# set parameters for 'kmeans' method in GenBsSamps.
C=4
bs.samps = GenBsSamps(dat$X,100,"kmeans")
# find mean partition and uncertainty statistics.
idx = align2(bs.samps$Z,100)
res = MeanPart(bs.samps$Z, 100, idx)
# plot the result on two dimensional space.
plot.part(dat$X,dat$z)          # ground truth
plot.part(dat$X,kmeans(dat$X,C)$cluster)  # baseline method
plot.part(dat$X,res$repre)      # OTA
# distance between ground truth and each partition
WassDist(c(dat$z,kmeans(dat$X,C)$cluster,res$repre),3)
```

Index