Package 'UNPaC'

April 13, 2020

Title Non-Parametric Cluster Significance Testing with Reference to a Unimodal Null Distribution

Version 1.1.0

Date 2020-03-18

Author Erika S. Helgeson, David Vock, and Eric Bair

Maintainer Erika S. Helgeson <helge@umn.edu>

Description

Assess the significance of identified clusters and estimates the true number of clusters by comparing the explained variation due to the clustering from the original data to that produced by clustering a unimodal reference distribution which preserves the covariance structure in the data. The reference distribution is generated using kernel density estimation and a Gaussian copula framework. A dimension reduction strategy and sparse covariance estimation optimize this method for the high-dimensional, low-sample size setting. This method is similar to them method described in Helgeson and Bair (2016) <arXiv:1610.01424> except a Gaussian copula approach is used to account for feature correlation.

Depends R (>= 3.6.0)

Imports huge, PDSCE

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-13 14:30:02 UTC

R topics documented:

UNPaC_Copula *Unimodal Non-Parametric Cluster (UNPaC) Significance Test*

Description

The UnPAC test assesses the significance of clusters by comparing the cluster index (CI) from the data to the CI from a ortho-unimodal reference data generated using a Gaussian copula. This method is similar to them method described in Helgeson and Bair (2016) except a Gaussian copula approach is used to account for feature correlation. The CI is defined to be the sum of the within-cluster sum of squares about the cluster means divided by the total sum of squares. Smaller values of the CI indicate a stronger clustering.

Usage

```
UNPaC_Copula(x, cluster, cluster.fun, nsim = 100,
  var_selection = FALSE, gamma = 0.1, p.adjust = "fdr", k = 2,
  rho = 0.02, cov = "glasso", center = TRUE, scale = FALSE)
```

Arguments

x	a dataset with n observations (rows) and p features (columns)
cluster	labels generated by clustering method
cluster.fun	function used to cluster data. Function should return list containing a component "cluster." Examples include kmeans and pam.
nsim	a numeric value specifying the number of unimodal reference distributions used for testing (default=100)
var_selection	should dimension be reduced using feature filtering procedure? See description below. (default=FALSE)
gamma	threshold for feature filtering procedure. See description below. Not used if var_selection=FALSE (default=0.10)
p.adjust	p-value adjustment method for additional feature filtering. See p.adjust for options. (default="fdr"). Not used if p.adjust="none."
k	integer value specifying the number of clusters to test (default=2)
rho	a regularization parameter used in implementation of the graphical lasso. See documentation for lambda in huge. Not used if cov="est" or cov="banded"
cov	method used for approximating the covariance structure. options include: "glasso" (See huge), "banded" (See band.chol.cv) and "est" (default = "glasso")
center	should data be centered such that each feature has mean equal to zero prior to clustering (default=TRUE)
scale	should data be scaled such that each feature has variance equal to one prior to clustering (default=FALSE)

Details

There are three options for the covariance matrix used in generating the Gaussian copula: sample covariance estimation, `cov="est"`, which should be used if n>p; the graphical lasso, `cov="glasso"`, which should be used if n<p; and k-banded covariance, `cov="banded"`, which can be used if n<p and it can be assumed that features farther away in the ordering have weaker covariance. The graphical lasso is implemented using the `huge` function. When `cov="banded"` is selected the k-banded covariance Cholesky factor of Rothman, Levina, and Zhu (2010) is used to estimate the covariance matrix. Cross-validation is used for selecting the banding parameter. See documentation in `band.chol.cv`.

In high dimensional (n<p) settings a dimension reduction step can be implemented which selects features based on an F-test for difference in means across clusters. Features having a p-value less than a threshold `gamma` are retained. For additional feature filtering a p-value adjustment procedure (such as p.adjust="fdr") can be used. If no features are retained the resulting p-value for the cluster significance test is given as 1.

Value

The function returns a list with the following components:

- `selected_features`: A vector of integers indicating the features retained by the feature filtering process.
- `sim_CI`: vector containing the cluster indices for each generated unimodal reference distribution
- `pvalue_emp`: the empirical p-value: the proportion of times the cluster index from the reference data is smaller the cluster index from the observed data
- `pvalue_norm`: the normalized p-value: the simulated p-value based on comparison to a standard normal distribution

Author(s)

Erika S. Helgeson, David Vock, Eric Bair

References

- Helgeson E and Bair E (2016). “Non-Parametric Cluster Significance Testing with Reference to a Unimodal Null Distribution." arXiv preprint arXiv:1610.01424.
- Rothman, A. J., Levina, E., and Zhu, J. (2010). “A new approach to Cholesky-based covariance regularization in high dimensions." Biometrika 97(3): 539-550.

Examples

```
# K-means example
test1 <- matrix(rnorm(100*50), nrow=100, ncol=50)
test1[1:30,1:50] <- rnorm(30*50, 2)
test.data<-scale(test1,scale=FALSE,center=TRUE)
cluster<-kmeans(test.data,2)$cluster
UNPaCResults <- UNPaC_Copula(test.data,cluster,kmeans, nsim=100,cov="est")
```

```
# Hierarchical clustering example

test <- matrix(nrow=1200, ncol=75)
theta <- rep(NA, 1200)
theta[1:500] <- runif(500, 0, pi)
theta[501:1200] <- runif(700, pi, 2*pi)
test[1:500,seq(from=2,to=50,by=2)] <- -2+5*sin(theta[1:500])
test[501:1200,seq(from=2,to=50,by=2)] <- 5*sin(theta[501:1200])
test[1:500,seq(from=1,to=49,by=2)] <- 5+5*cos(theta[1:500])
test[501:1200,seq(from=1,to=49,by=2)] <- 5*cos(theta[501:1200])
test[,1:50] <- test[,1:50] + rnorm(50*1200, 0, 0.2)
test[,51:75] <- rnorm(25*1200, 0, 1)
test.data<-scale(test,center=TRUE,scale=FALSE)
# Defining clustering function
hclustFunction<-function(x,k){
 D<-dist(x)
 xn.hc <- hclust(D, method="single")
 list(cluster=cutree(xn.hc, k))}

cluster=hclustFunction(test.data,2)$cluster
UNPaCResults <- UNPaC_Copula(test.data,cluster,hclustFunction, nsim=100,cov="est")
```

`UNPaC_num_clust` *Unimodal Non-Parametric Cluster (UNPaC) Test for Estimating Number of Clusters*

Description

UNPaC for estimating the number of clusters Compares the cluster index (CI) from the original data to that produced by clustering a simulated ortho-unimodal reference distribution generated using a Gaussian copula. The CI is defined to be the sum of the within-cluster sum of squares about the cluster means divided by the total sum of squares. The number of clusters is chosen to maximize the difference between the data cluster index and the reference cluster indices, but additional rules are also implmented (See below). This method is similar to them method described in Helgeson and Bair (2016) except a Gaussian copula approach is used to account for feature correlation and the rules for choosing the number of clusters are as described below.

Usage

```
UNPaC_num_clust(x, k = 10, cluster.fun, nsim = 1000, cov = "glasso",
  rho = 0.02, scale = FALSE, center = FALSE, var_selection = FALSE,
  p.adjust = "none", gamma = 0.1, d.power = 1)
```

Arguments

x	a dataset with n observations (rows) and p features (columns)
k	maximum number of clusters considered. (default=10)

cluster.fun	function used to cluster data. Function should return list containing a component "cluster." Examples include kmeans and pam.
nsim	a numeric value specifying the number of unimodal reference distributions used for testing (default=1000)
cov	method used for approximating the covariance structure. options include: "glasso" (See huge), "banded" (See band.chol.cv) and "est" (default = "glasso")
rho	a regularization parameter used in implementation of the graphical lasso. See documentation for lambda in huge. Not used if cov="est" or cov="banded"
scale	should data be scaled such that each feature has variance equal to one prior to clustering (default=FALSE)
center	should data be centered such that each feature has mean equal to zero prior to clustering (default=TRUE)
var_selection	should dimension be reduced using feature filtering procedure? See description below. (default=FALSE)
p.adjust	p-value adjustment method for additional feature filtering. See p.adjust for options. (default="fdr"). Not used if p.adjust="none."
gamma	threshold for feature filtering procedure. See description below. Not used if var_selection=FALSE (default=0.10)
d.power	Power in estimating the low of the within cluster dispersion for comparison to the Gap statistic. See clusGap.

Details

There are three options for the covariance matrix used in generating the Gaussian copula: sample covariance estimation, cov="est", which should be used if n>p; the graphical lasso, cov="glasso", which should be used if n<p; and k-banded covariance, cov="banded", which can be used if n<p and it can be assumed that features farther away in the ordering have weaker covariance. The graphical lasso is implemented using the huge function. When cov="banded" is selected the k-banded covariance Cholesky factor of Rothman, Levina, and Zhu (2010) is used to estimate the covariance matrix. Cross-validation is used for selecting the banding parameter. See documentation in band.chol.cv.

In high dimensional (n<p) settings a dimension reduction step can be implemented which selects features based on an F-test for difference in means across clusters. Features having a p-value less than a threshold gamma are retained. For additional feature filtering a p-value adjustment procedure (such as p.adjust="fdr") can be used. If no features are retained the resulting p-value for the cluster significance test is given as 1.

Value

The function returns a list with the following components:

- BestK: A matrix with 1 row and 4 columns named: "Max_CI","Max_CI_wi_1SE","Max_scaled_CI" and "Max_logWCSS_wi_1SE". These correspond to the number of clusters, K, chosen by four different rules. "Max_CI choses K to maximize the difference in CI's between the true data and the reference data. "Max_CI_wi_1SE" uses the "1-SE" criterion as in Tibshirani et al (2001), except for the CI. "Max_scaled_CI" chooses K to maximize the difference in CIs

from the observed and reference data scaled by the standard error of the reference data CIs. "Max_logWCSS_wi_1SE" uses the Gap statistic and the "1-SE" criterion (Tibshirani et al, 2001) for choosing K.

- `full_process`: A matrix containing the number of clusters, K, evaluated, the CI from the data, the average CI from the null distribution, the difference between the data CI and average null CI, the standard error for the difference in CIs, the log of the within cluster dispersion from the data, the average log of within cluster dispersion from the null data, The difference in within cluster dispersion (the Gap statistic), and the standard error for the Gap statistic.
- `selected_features`: A vector of integers indicating the features retained by the feature filtering process.

Author(s)

Erika S. Helgeson, David Vock, Eric Bair

References

- Helgeson E and Bair E (2016). Non-Parametric Cluster Significance Testing with Reference to a Unimodal Null Distribution. arXiv preprint arXiv:1610.01424.
- Tibshirani, R., Walther, G. and Hastie, T. (2001). Estimating the number of data clusters via the Gap statistic. Journal of the Royal Statistical Society B, 63, 411-423.

Examples

```
test1 <- matrix(rnorm(100*50), nrow=100, ncol=50)
 test1[1:30,1:50] <- rnorm(30*50, 2)
 test.edit<-scale(test1,center=TRUE,scale=FALSE)
 UNPaC_k<-UNPaC_num_clust(test.edit,k=5,kmeans,nsim=100,cov="est")
```

Index