

# Package ‘AnomalyDetection’

August 13, 2020

**Type** Package

**Title** Anomaly detection on time series of graphs

**Version** 0.1.0

**Author** Guodong Chen

**Maintainer** Guodong Chen <gchen35@jhu.edu>

**Description** Use multiple graph embedding to perform anomaly detection on time series of graphs

**License** GPL-3.

**Depends** foreach (>= 1.4.7),

qcc (>= 2.7),  
doParallel (>= 1.0.15),  
ggplot2 (>= 3.2.1),  
irlba (>= 2.3.3),  
Matrix (>= 1.2-17),  
gtools (>= 3.8.1),  
latex2exp (>= 0.4.0),  
dplyr (>= 0.8.3),  
igraph (>= 1.2.4.1),  
foreach (>= 1.4.7),  
qcc (>= 2.7),  
rARPACK

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Suggests** knitr,  
rmarkdown

**VignetteBuilder** knitr

## R topics documented:

ase . . . . .	2
buildOmni . . . . .	3
diagAug . . . . .	3
doMase . . . . .	4
fast2buildOmni . . . . .	4
fast2doOmni . . . . .	5
generate.tsg . . . . .	5

getdegchange . . . . .	6
getElbows . . . . .	6
getweightchange . . . . .	7
giant.component . . . . .	8
jlcc . . . . .	8
mase . . . . .	9
mase.latent . . . . .	9
pdistXY . . . . .	10
plot.qcc . . . . .	10
plot.qcc.vertex . . . . .	11
pltclique . . . . .	12
project_networks . . . . .	12
ptr . . . . .	13
qcc . . . . .	13
qccAD . . . . .	14
rdpg.sample . . . . .	16
<b>Index</b>	<b>17</b>

ase

*Function to perform graph adjacency spectral embedding (ASE)*

## Description

Function to perform graph adjacency spectral embedding (ASE)

## Usage

```
ase(
  A,
  d = NULL,
  d.max = round(log(nrow(A))),
  diagaug = TRUE,
  approx = TRUE,
  elbow = 2
)
```

## Arguments

A	adjacency matrix
d	number of embedding dimension. If NULL, dimension is chosen automatically.
d.max	maximum number of embedding dimensions to try when d is not provided. Default is round(log(nrow(A))).
diagaug	whether to do diagonal augmentation (TRUE/FALSE)
approx	whether to find a few approximate singular values and corresponding singular vectors of a matrix using irlba package (TRUE/FALSE).
elbow	number of elbow selected in Zhu & Ghodsi method for the scree plot of each individual graph singular values. Default is 2.

**Value**

A matrix with n rows and d columns containing the estimated latent positions

**References**

Zhu, Mu and Ghodsi, Ali (2006), Automatic dimensionality selection from the scree plot via the use of profile likelihood, *Computational Statistics & Data Analysis*, Volume 51 Issue 2, pp 918-930, November, 2006.

Sussman, D.L., Tang, M., Fishkind, D.E., Priebe, C.E. A Consistent Adjacency Spectral Embedding for Stochastic Blockmodel Graphs, *Journal of the American Statistical Association*, Vol. 107(499), 2012

---

buildOmni	<i>Function to build OMNI matrix</i>
-----------	--------------------------------------

---

**Description**

Function to build OMNI matrix

**Usage**

```
buildOmni(Alist, diagaug = FALSE)
```

**Arguments**

Alist	a list (length M) of n x n adjacency matrices or igraph objects
diagaug	whether to do diagonal augmentation (TRUE/FALSE)

**Value**

Omnibus matrix of  $M_n \times M_n$

---

diagAug	<i>Function to perform diagonal augmentation for graph adjacency matrix.</i>
---------	--

---

**Description**

Function to perform diagonal augmentation for graph adjacency matrix.

**Usage**

```
diagAug(A)
```

**Arguments**

A	a hollow adjacency matrix
---	---------------------------

**Value**

a non-hollow adjacency matrix

---

doMase	<i>Function to calculate <math>l_2</math> distance between adjacent latent position estimate for graphs and vertices with MASE</i>
--------	--

---

### Description

Function to calculate  $l_2$  distance between adjacent latent position estimate for graphs and vertices with MASE

### Usage

```
doMase(glist, latpos.list, nmase = 2, dsvd = NULL, attrweight = NULL)
```

### Arguments

glist	a list (length M) of n x n adjacency matrices or igraph objects
latpos.list	a list (length M) of ASE estimates for each graphs
nmase	number of graphs to do joint embedding of MASE. It can only be 2 or M.
dsvd	dsvd is number of dimension to do joint svd for MASE. If NULL then dimension is chosen automatically as the second elbow selected in Zhu & Ghodsi method for the scree plot of the singular values of the concatenated spectral embeddings of individual ASE estimates.

### Value

A list containing a vector tnorm of length M-1, with the latent position estimate difference for graphs, and a matrix pdist with latent position estimate difference for vertices (size n x M-1 ).

---

fast2buildOmni	<i>Function to build OMNI matrix with two matrix</i>
----------------	--

---

### Description

Function to build OMNI matrix with two matrix

### Usage

```
fast2buildOmni(Alist, diagaug = FALSE, attrweight = NULL)
```

### Arguments

Alist	a list (length 2) of n x n adjacency matrices or igraph objects
diagaug	whether to do diagonal augmentation (TRUE/FALSE)

### Value

Omnibus matrix of 2n x 2n

---

fast2doOmni	<i>Function to calculate <math>l_2</math> distance between adjacent latent position estimate for graphs and vertices with OMNI</i>
-------------	--

---

**Description**

Function to calculate  $l_2$  distance between adjacent latent position estimate for graphs and vertices with OMNI

**Usage**

```
fast2doOmni(n, Z)
```

**Arguments**

n	number of vertices in each graph
Z	a matrix of size $2n \times d$ as the latent estimate for the omnibus matrix constructed by adjacent graphs

**Value**

A list containing a numeric value `tnorm`, with the  $l_2$  norm of latent position estimate difference for adjacent graphs, and a vector `pdist` with  $l_2$  distance between latent position estimates for each vertex (size `n`).

---

<code>generate.tsg</code>	<i>Sample a time series of RDPG graph (length <code>tmax</code> &gt; 17) with same 1-1 matched vertices unweighted hollow symmetric undirected graphs, the latent positions i.i.d uniform. Some vertices in 16-th and 17-th graphs are given perturbations so there exists anomalies at 16:17.</i>
---------------------------	--

---

**Description**

Sample a time series of RDPG graph (length `tmax` > 17) with same 1-1 matched vertices unweighted hollow symmetric undirected graphs, the latent positions i.i.d uniform. Some vertices in 16-th and 17-th graphs are given perturbations so there exists anomalies at 16:17.

**Usage**

```
generate.tsg(n, nperturb, cperturb = NULL, rmin, rmax, tmax)
```

**Arguments**

n	number of vertices
nperturb	number of perturbed vertices
cperturb	number of perturbation. Larger <code>cperturb</code> means more obvious anomalies.
rmin, rmax	parameter for <code>uniform[rmin, rmax]</code> .
tmax	number of graphs must be greater than 17.

**Value**

A list containing a vector `tnorm` of length `tmax-1`, with the latent position difference for graphs, and a matrix `pdist` with latent position estimate difference for vertices (size `n x tmax-1`), and a list of length `tmax` of undirected hollow symmetric unweighted graphs

**Examples**

```
# Sample a time series of RDPG graph (length tmax > 17) with same 1-1 matched vertices unweighted
# hollow symmetric undirected graphs, the latent positions i.i.d uniform.
# Some vertices in 16-th and 17-th graphs are given perturbations so there exists anomalies at 16:17.
n <- 100 #number of vertices
nperturb <- 20 #number of perturbed vertices
cperturb <- .12 #number of perturbation, larger cperturb means more obvious anomalies.
rmin <- .2 # parameter for uniform[rmin, rmax].
rmax <- .8 # parameter for uniform[rmin, rmax].
tmax <- 22 # number of graphs must be greater than 17.
#Generate data or load the data you want
glist <- generate.tsg(n, nperturb, cperturb=NULL, rmin, rmax, tmax)$glist
```

---

<code>getdegchange</code>	<i>It extracts (non-zero) degree change <code>deg.change</code> matrix <code>n</code> by <code>m-1</code> from a list of graphs.</i>
---------------------------	--

---

**Description**

It extracts (non-zero) degree change *deg.change* matrix `n` by `m-1` from a list of graphs.

**Usage**

```
getdegchange(gip)
```

**Arguments**

`gip` a list of graphs in `igraph` format.

**Value**

A matrix of size `n x t-1`, with each element to be degree changes

---

<code>getElbows</code>	<i>Given a decreasingly sorted vector, return the given number of elbows</i>
------------------------	--

---

**Description**

Given a decreasingly sorted vector, return the given number of elbows

**Usage**

```
getElbows(dat, n = 3, threshold = FALSE, plot = TRUE, main = "", ...)
```

**Arguments**

<code>dat</code>	a input vector (e.g. a vector of standard deviations), or a input feature matrix
<code>n</code>	the number of returned elbows
<code>threshold</code>	either FALSE or a number. If threshold is a number, then all the elements in <code>d</code> that are not larger than the threshold will be ignored.
<code>plot</code>	logical. When T, it depicts a scree plot with highlighted elbows
<code>main</code>	a string of the plot title

**Value**

a vector of length  $n$

**References**

Zhu, Mu and Ghodsi, Ali (2006), Automatic dimensionality selection from the scree plot via the use of profile likelihood, Computational Statistics & Data Analysis, Volume 51 Issue 2, pp 918-930, November, 2006.

---

<code>getweightchange</code>	<i>get weighted degree change for a list of weighted graphs It extracts (non-zero) weighted degree change matrix <math>n</math> by <math>m-1</math> <code>deg.change</code> from a list of graphs.</i>
------------------------------	--

---

**Description**

get weighted degree change for a list of weighted graphs It extracts (non-zero) weighted degree change matrix  $n$  by  $m-1$  *deg.change* from a list of graphs.

**Usage**

```
getweightchange(gip)
```

**Arguments**

<code>gip</code>	a list of graphs in igraph format.
------------------	------------------------------------

**Value**

A matrix of size  $n \times t-1$ , with each element to be weight degree changes

---

<code>giant.component</code>	<i>find largest connected component in a graph It extracts (non-zero) largest connected subgraph .</i>
------------------------------	--

---

### Description

find largest connected component in a graph It extracts (non-zero) largest connected subgraph .

### Usage

```
giant.component(graph, ...)
```

### Arguments

<code>graph</code>	a graph in igraph format
--------------------	--------------------------

---

<code>jlcc</code>	<i>remove edges which has zero weights for all graphs (if any) and find jointly largest connected component in graphs. Finally it removes all self-loops, It extracts (non-zero) igraph list <i>gip</i> and removes all edges with zero edge weights and return a list of jointly largest connected component in graphs without self-loops .</i>
-------------------	--

---

### Description

remove edges which has zero weights for all graphs (if any) and find jointly largest connected component in graphs. Finally it removes all self-loops, It extracts (non-zero) igraph list *gip* and removes all edges with zero edge weights and return a list of jointly largest connected component in graphs without self-loops .

### Usage

```
jlcc(gip)
```

### Arguments

<code>gip</code>	a list of graphs in igraph format
------------------	-----------------------------------

### Value

A list of graphs in igraph format



mase

*Function to perform joint embedding part of MASE***Description**

Function to perform joint embedding part of MASE

**Usage**

```
mase(
  Adj_list,
  latpos.list,
  dsvd = NULL,
  elbow_mase = 2,
  show.scree.results = FALSE
)
```

**Arguments**

**Adj\_list** a list of adjacency matrices with the same size  $n \times n$

**latpos.list** Individual ASE estimate for the adjacent graphs

**elbow\_mase** number of elbow selected in Zhu & Ghodsi method for the scree plot of the singular values of the concatenated spectral embeddings of MASE.

**show.scree.results** when TRUE, the histogram of the estimated  $d$  for each graph, and the scree plot of the singular values of the graph is shown if  $d$  is not specified.

**d** number of joint embedding dimensions. If NA, dimension is chosen automatically

**Value**

A list containing a matrix  $V$  of size  $n \times d$ , with the estimated invariant subspace, and a list  $R$  with the individual score parameters for each graph (size  $d \times d$  each).

mase.latent

*Function to obtain latent position estimates for MASE (Multiple Adjacency Spectral Embedding) with individual ASE estimates for corresponding graphs*

**Description**

Function to obtain latent position estimates for MASE (Multiple Adjacency Spectral Embedding) with individual ASE estimates for corresponding graphs

**Usage**

```
mase.latent(A, latpos.list, dsvd = NULL)
```

**Arguments**

<code>A</code>	a list (length M-1) of n x n adjacency matrices
<code>latpos.list</code>	a list (length M-1) of individual ASE estimates for the corresponding adjacency matrices
<code>dsvd</code>	dimension for joint embedding. If NULL then dimension is chosen automatically as the second elbow selected in Zhu & Ghodsi method for the scree plot of the singular values of the concatenated spectral embeddings of invidual ASE estimates.

**Value**

A list containing a vector `tnorm` of length M-1, with  $l_2$  norm of latent position estimate difference for graphs, and a matrix `pdist` with  $l_2$  distance between latent position estimate for each vertex (size n x M-1 ).

---

<code>pdistXY</code>	<i>Calculate <math>l_2</math> distance between latent positions for vertices.</i>
----------------------	---

---

**Description**

Calculate  $l_2$  distance between latent positions for vertices.

**Usage**

```
pdistXY(X, Y)
```

**Arguments**

`X, Y` are matrices with n rows and d columns

**Value**

A vector of size  $n$ , with element being  $l_2$  distance between rows of X and Y.

---

<code>plot.qcc</code>	<i>Function to plot Shewhart chart for GraphAD</i>
-----------------------	--

---

**Description**

Function to plot Shewhart chart for GraphAD

**Usage**

```
## S3 method for class 'qcc'
plot(x, l = 1, title, plot.LCL = FALSE)
```

**Arguments**

x	a qcc object
l	length of previous graphs to estimate moving averages and moving standard deviation.
title	a string of the plot title
plot.LCL	a Boolean variable to decide whether to show the anomalies lower than lower limits (LCL $\mu^t - 3\sigma^t$ ) .

**Value**

A control chart ggplot for GraphAD

---

plot.qcc.vertex	<i>Function to plot Shewhart chart for VertexAD</i>
-----------------	---

---

**Description**

Function to plot Shewhart chart for VertexAD

**Usage**

```
## S3 method for class 'qcc.vertex'
plot(x, l = 1, title, plot.LCL = FALSE)
```

**Arguments**

x	a list of qcc object
l	length of previous graphs to estimate moving averages and moving standard deviation.
title	a string of the plot title
plot.LCL	a Boolean variable to decide whether to show the anomalies lower than lower limits (LCL $\mu_i^t - 3\sigma_i^t$ ) .

**Value**

A control chart ggplot for VertexAD

---

pltclique	<i>create a planted clique It creates planted clique for a specific graph for a list of graphs.</i>
-----------	---

---

**Description**

create a planted clique It creates planted clique for a specific graph for a list of graphs.

**Usage**

```
pltclique(gip, p, art.anomaly.v)
```

**Arguments**

gip	a list of graphs in igraph format.
p	is the index of graph to be inserted a planted clique
art.anomaly.v	is the vertex index in igraph.vs format to be planted clique.

**Value**

A list containing a planted clique size as size of art.anomaly.v at p-th graph

---

project_networks	<i>Function to estimated the score matrices of a list of graphs given the common invariant subspace V</i>
------------------	---

---

**Description**

Function to estimated the score matrices of a list of graphs given the common invariant subspace V

**Usage**

```
project_networks(Adj_list, V)
```

**Arguments**

Adj_list	list of adjacency matrices, of size n x n
V	common invariant subspace. A matrix of size n x d.

**Value**

A list containing the score matrices

---

ptr	<i>Run pass-to-rank on a weighted graph.</i>
-----	--

---

### Description

It extracts (non-zero) edge weight vector  $W$  from a graph and replaces it with  $2 * R / (|E| + 1)$  where  $R$  is the rank of  $W$  and  $|E|$  is the number of edges. This does 'no-op' for an unweighted graph.

### Usage

```
ptr(g)
```

### Arguments

`g` a graph in igraph format or an  $n \times 2$  edge list or an  $n \times n$  adjacency matrix

---

qcc	<i>Main function to create a 'qcc' object</i>
-----	---

---

### Description

Main function to create a 'qcc' object

### Usage

```
qcc(
  data,
  type = c("xbar", "R", "S", "xbar.one", "p", "np", "c", "u", "g"),
  sizes,
  center,
  std.dev,
  limits,
  data.name,
  labels,
  newdata,
  newsizes,
  newdata.name,
  newlabels,
  nsigmas = 3,
  confidence.level,
  rules = shewhart.rules,
  plot = TRUE,
  plot.LCL = FALSE,
  ...
)
```

### Arguments

`plot.LCL` a Boolean variable to decide whether to show the anomalies lower than lower limits ( $LCL \mu^t - 3\sigma^t$ ).

## References

see R package qcc

---

qccAD

---

*Function to perform anomaly detection for time series of graphs*


---

## Description

Function to perform anomaly detection for time series of graphs

## Usage

```
qccAD(
  glist,
  method = "OMNI",
  diag.augment = TRUE,
  l = 3,
  d = NULL,
  dsvd = d,
  approx = TRUE,
  par = FALSE,
  numpar = 2,
  elbow = 2,
  plot.figure = TRUE,
  plot.LCL = FALSE
)
```

## Arguments

- |              |   |
|--------------|---|
| glist        | a list of undirected simple graphs (simple graphs are graphs which do not contain self-loop and multiple edges) in igraph format with same number of vertices with vertices are 1-1 matched. Graphs can be weighted or binary. (Say the length of list to be tmax)  |
| method       | a character variable to be chosen among c("OMNI","MASE"). The code will first do OMNIbus embedding (OMNI) or Multiple Adjacency Spectrally Ebedding (MASE) with two adjcaency matrices(can be weighted or not) of all input adjacent graphs sequentially. Then use latent positions to calculate test statistics $y^t =   X^t - X^{t+1}  $ using operator norm. Then for $t = l, \dots, tmax - 1$ , we calculate the moving means $\mu^t$ and moving standard deviations $\sigma^t$ at $t$ by $y^{t-l+1}, \dots, y^{t-1}$ . So only tmax-l time points are plotted as first l graphs have been used as estimating moving means and standard deviations. |
| diag.augment | a Boolean variable to decide whether to do diagonal augmentation when performing adjacency spectral embedding. Default is TRUE.   |
| l            | an integer of the number of graphs in time window in estimating the moving mean and moving standard deviation. l must be less than number of graphs and be greater than 3.  |
| d            | a fixed integer of dimension to perform OMNI and individual ASE for MASE. If d is NULL, then dimension is chosen automatically.   |

dsvd	An integer number of dimension only used in joint embedding for MASE. If NULL, then dimension is chosen automatically as second elbow selected in Zhu & Ghodsi method for the scree plot of the singular values of the concatenated spectral embeddings of MASE.
approx	a Boolean variable to decide whether to use irlba package to find a few approximate singular values and corresponding singular vectors of a matrix. Default is TRUE.
par	a Boolean variable to decide whether to run in parallel. Default is FALSE.
numpar	an integer number to decide number of clusters for parallel implementation. Default is 2.
elbow	number of elbow in Zhu & Ghodsi method for the scree plot of each individual graph singular values for MASE or of each omnibus matrix singular values for OMNI.
plot.figure	a Boolean variable to decide whether plot control chart. Default is TRUE.
plot.LCL	a Boolean variable to decide whether to show the anomalies lower than lower limits ( $LCL \mu^t - 3\sigma^t$ ).

### Value

A list containing a vector GraphAD of length tmax-l which consists of control charts deviations, with the a list VertexAD (length tmax-l) with vectors of anomalous vertices indices for each graph.

### References

Zhu, Mu and Ghodsi, Ali (2006), Automatic dimensionality selection from the scree plot via the use of profile likelihood, Computational Statistics & Data Analysis, Volume 51 Issue 2, pp 918-930, November, 2006.

Levin, K., Athreya, A., Tang, M., Lyzinski, V. and Priebe, C.E., 2017, November. A central limit theorem for an omnibus embedding of multiple random dot product graphs. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 964-967). IEEE.

Arroyo, J., Athreya, A., Cape, J., Chen, G., Priebe, C.E. and Vogelstein, J.T., 2019. Inference for multiple heterogeneous networks with a common invariant subspace. arXiv preprint arXiv:1906.10026.

### Examples

```
glist <- list()
for (i in 1:5) {
  glist[[i]] <- sample_gnp(100,.1)
}
glist[[6]] <- sample_gnp(100,.9)
glist[[7]] <- sample_gnp(100,.1)
for (i in 8:12) {
  glist[[i]] <- sample_gnp(100,.1)
}
result<- qccAD(glist, l=4,d=1,dsvd=1,method="OMNI",
diag.augment = TRUE,approx=FALSE, par=FALSE, numpar=2)
print(result.OMNI$GraphAD) #print the number of deviation for GraphAD, only positive ones are meaningful

# Sample a time series of RDPG graph (length tmax > 17) with same 1-1 matched vertices unweighted
# hollow symmetric undirected graphs, the latent positions i.i.d uniform.
# Some vertices in 16-th and 17-th graphs are given perturbations so there exists anomalies at 16:17.
n <- 100 #number of vertices
```

```

nperturb <- 20 #number of perturbed vertices
cperturb <- .12 #number of perturbation, larger cperturb means more obvious anomalies.
rmin <- .2 # parameter for uniform[rmin, rmax].
rmax <- .8 # parameter for uniform[rmin, rmax].
tmax <- 22 # number of graphs must be greater than 17.
#Generate data or load the data you want
glist <- generate.tsg(n, nperturb, cperturb=NULL, rmin, rmax, tmax)$glist
#Do anomaly detection with OMNI in parallel
result.OMNI <- qccAD(glist, l=11,d=1,dsvd=NULL,method="OMNI",
                    diag.augment = TRUE, approx=FALSE, par=TRUE, numpar=2)
#print the number of deviation for GraphAD, only positive ones are meaningful
print(result.OMNI$GraphAD)

# Do anomaly detection with MASE in parallel
result.MASE<- qccAD(glist, l=11,d=1,dsvd=2,method="MASE",
                    diag.augment = TRUE, approx=FALSE, par=TRUE, numpar=2)
#print the number of deviation for GraphAD, only positive ones are meaningful
print(result.MASE$GraphAD)

```

---

rdpg.sample

---

*Sample RDPG graph with latent position*


---

## Description

Sample RDPG graph with latent position

## Usage

```
rdpg.sample(X)
```

## Arguments

$X$                       latent position matrix of  $d$  columns and  $n$  rows.

## Value

A un-directed hollow symmetric unweighted graph generated from bernoulli  $EA = P = XX^T$



# Index

ase, [2](#)

buildOmni, [3](#)

diagAug, [3](#)  
doMase, [4](#)

fast2buildOmni, [4](#)  
fast2doOmni, [5](#)

generate.tsg, [5](#)  
getdegchange, [6](#)  
getElbows, [6](#)  
getweightchange, [7](#)  
giant.component, [8](#)

jlcc, [8](#)

mase, [9](#)  
mase.latent, [9](#)

pdistXY, [10](#)  
plot.qcc, [10](#)  
plot.qcc.vertex, [11](#)  
pltclique, [12](#)  
project\_networks, [12](#)  
ptr, [13](#)

qcc, [13](#)  
qccAD, [14](#)

rdpg.sample, [16](#)