Package 'NetCluster'

January 20, 2025

Type Package

Version 0.2

Date 2010-05-09

Title Clustering for networks

Author Mike Nowak <michael.nowak@gmail.com>, Solomon Messing
<solomon.messing@gmail.com>, Sean J Westwood
<seanjw@stanford.edu>, and Dan McFarland
<dmcfarla@stanford.edu>

Maintainer Sean J Westwood <seanjw@stanford.edu>

Depends stats, sna

Description Facilitates network clustering and evaluation of cluster
configurations.

License GPL-2

LazyLoad yes

Repository CRAN

Date/Publication 2012-10-29 08:57:21

NeedsCompilation no

Contents

---

clustConfigurations   *Fit of Cluster Configurations for Networks*

Description

Evaluates clustering solutions for n = 1, n = 2, ..., n = n clusters, by comparing the clustered matrix to the observed correlation matrix. Returns a correlation vector and a plot. Designed for networks.

Usage

```
clustConfigurations(vertices, hclustresult, observedcorrelation)
```

Arguments

vertices   scalar value indicating the number of vertices

hclustresult   hclust result matrix object (or similar object that works with the cutree() function)

observedcorrelation
                  the observed correlation matrix

Details

This function helps the user discern the number of clusters that best describe the underlying data. It loops through all of possible clusters (1 through n, where n is the number of actors in the network). For each solution corresponding to a given number of clusters, it uses the cutree() to assign the vertices (or columns) to their respective clusters corresponding to that solution.

From this, the function generates a matrix of within- and between- cluster correlations. When there is one cluster for each vertex in the network, the cell values will be identical to the observed correlation matrix. When there is one cluster for the whole network, the values will all be equal to the average correlation across the observed matrix.

From a visual inspection of the correlation matrix, the user can decide on the proper number of clusters in this network.

Value

clustConfigurations$correlations
                  a vector of length n showing correlation between clustered and observed correlation matrix

Author(s)

Mike Nowak <michael.nowak@gmail.com>

Examples

```
# Generate socmatrix
socmatrix = matrix(c(1,1,0,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0), nrow = 5, ncol = 5)
socmatrix
socmatrix_cors <- cor(socmatrix)
socmatrix_cors

# To use correlation values in hierarchical clustering, they must
# first be coerced into a "dissimilarity structure" using dist().
# We subtract the values from 1 so that they are all greater than
# or equal to 0; thus, highly dissimilar (i.e., negatively
# correlated) actors have higher values.
dissimilarity <- 1 - socmatrix_cors
socmatrix_dist <- as.dist(dissimilarity)
socmatrix_dist

# hclust() performs a hierarchical agglomerative clustering
# operation based on the values in the dissimilarity matrix
# yielded by as.dist() above. The standard visualization is a
# dendrogram.
socmatrix_hclust <- hclust(socmatrix_dist)
plot(socmatrix_hclust)

# cutree() allows us to use the output of hclust() to set
# different numbers of clusters and assign vertices to clusters
# as appropriate. For example:
cutree(socmatrix_hclust, k=2)

# Now we'll try to figure out the number of clusters that best
# describes the underlying data. To do this, we'll loop through
# all of the possible numbers of clusters (1 through n, where n is
# the number of actors in the network). For each solution
# corresponding to a given number of clusters, we'll use cutree()
# to assign the vertices to their respective clusters
# corresponding to that solution.
#
# From this, we can generate a matrix of within- and between-
# cluster correlations. Thus, when there is one cluster for each
# vertex in the network, the cell values will be identical to the
# observed correlation matrix, and when there is one cluster for
# the whole network, the values will all be equal to the average
# correlation across the observed matrix.
#
# We can then correlate each by-cluster matrix with the observed
# correlation matrix to see how well the by-cluster matrix fits
# the data. We'll store the correlation for each number of
# clusters in a vector, which we can then plot.

# First, find n:
num_vertices = ncol(socmatrix)

# Next, use the clustConfigurations function:
```

```
clustered_observed_cors <-clustConfigurations(num_vertices,socmatrix_hclust,socmatrix_cors)

# Choose n where the line starts to flatten beyond 45 degrees.
# Three looks like a good number for this example.

num_clusters = 3

clusters <- cutree(socmatrix_hclust, k = num_clusters)
clusters

( cluster_cor_mat <- clusterCorr(socmatrix_cors, clusters) )
```

---

clusterCorr   *Cluster correlation matrix for networks*

Description

clusterCorr by-cluster correlation matrix

Usage

```
clusterCorr(observed_cor_matrix, cluster_vector)
```

Arguments

observed_cor_matrix
                       observed correlation matrix

cluster_vector   vector of cluster membership

Value

clusterCorr   a by-cluster correlation matrix

Author(s)

Mike Nowak <michael.nowak@gmail.com>

Examples

```
# Generate socmatrix
socmatrix = matrix(c(1,1,0,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,1,0), nrow = 5, ncol = 5)
socmatrix
socmatrix_cors <- cor(socmatrix)
socmatrix_cors

# To use correlation values in hierarchical clustering, they must
# first be coerced into a "dissimilarity structure" using dist().
# We subtract the values from 1 so that they are all greater than
# or equal to 0; thus, highly dissimilar (i.e., negatively
```

```
# correlated) actors have higher values.
dissimilarity <- 1 - socmatrix_cors
socmatrix_dist <- as.dist(dissimilarity)
socmatrix_dist

# hclust() performs a hierarchical agglomerative clustering
# operation based on the values in the dissimilarity matrix
# yielded by as.dist() above. The standard visualization is a
# dendrogram.
socmatrix_hclust <- hclust(socmatrix_dist)
plot(socmatrix_hclust)

# cutree() allows us to use the output of hclust() to set
# different numbers of clusters and assign vertices to clusters
# as appropriate. For example:
cutree(socmatrix_hclust, k=2)

# Now we'll try to figure out the number of clusters that best
# describes the underlying data. To do this, we'll loop through
# all of the possible numbers of clusters (1 through n, where n is
# the number of actors in the network). For each solution
# corresponding to a given number of clusters, we'll use cutree()
# to assign the vertices to their respective clusters
# corresponding to that solution.
#
# From this, we can generate a matrix of within- and between-
# cluster correlations. Thus, when there is one cluster for each
# vertex in the network, the cell values will be identical to the
# observed correlation matrix, and when there is one cluster for
# the whole network, the values will all be equal to the average
# correlation across the observed matrix.
#
# We can then correlate each by-cluster matrix with the observed
# correlation matrix to see how well the by-cluster matrix fits
# the data. We'll store the correlation for each number of
# clusters in a vector, which we can then plot.

# First, find n:
num_vertices = ncol(socmatrix)

# Next, use the clustConfigurations function:
clustered_observed_cors <-clustConfigurations(num_vertices,socmatrix_hclust,socmatrix_cors)

# Choose n where the line starts to flatten beyond 45 degrees.
# Three looks like a good number for this example.

num_clusters = 3

clusters <- cutree(socmatrix_hclust, k = num_clusters)
clusters

( cluster_cor_mat <- clusterCorr(socmatrix_cors, clusters) )
```

---

generate_cluster_cor_mat
                              Generate Cluster Correlation Matrix

---

Description

generate_cluster_cor_mat generates the cluster correlation matrix to examine the within- and between-cluster correlations.

Usage

```
generate_cluster_cor_mat(observed_cor_matrix, cluster_vector)
```

Arguments

observed_cor_matrix
                    observed correlation matrix

cluster_vector   vector of clusters

Value

cluster_cor_mat
                    a cluster correlation matrix

Author(s)

Mike Nowak <michael.nowak@gmail.com>

Examples

```
##
```

---

permute_matrix   *Triad census for networks*

Description

permute_matrix permute the network to examine the within- and between-cluster correlations.

Usage

```
permute_matrix(mem_vector, adj_matrix)
```

Arguments

`mem_vector`	vector of cluster membership
`adj_matrix`	adjacency matrix

Value

`permute_matrix` a permuted matrix

Author(s)

Mike Nowak <michael.nowak@gmail.com>

Examples

```
##
```

Index