Package ‘dils’

July 16, 2013

Type Package

Title Data-Informed Link Strength. Combine multiple-relationship networks into a single weighted network.

Version 0.4

Date 2013-07-16

Author Stephen R. Haptonstahl

Maintainer Stephen R. Haptonstahl <srh@haptonstahl.org>

Depends R (>= 2.15.0), igraph

Suggests testthat

Description Combine multiple-relationship networks into a single weighted network. The approach is similar to factor analysys in the that contribution from each constituent network varies so as to maximize the information gleaned from the multimetwork. This implementation uses Principal Component Analysis calculated using ’prcomp’ with bootstrap subsampling.

License MIT + file LICENSE

Collate
’GetSampleFromDataFrame.R’ ’RelationStrengthSimilarity.R’’ScalablePCA.R’ ’GetSampleFromFile.R’ ’GetSample

R topics documented:

`dils-package` *Data-Informed Link Strength. Combine multiple-relationship networks into a single weighted network.*

Description

Combine multiple-relationship networks into a single weighted network. The approach is similar to factor analysys in the that contribution from each constituent network varies so as to maximize the information gleaned from the multimetwork. This implementation uses Principal Component Analysis calculated using 'prcomp' with bootstrap subsampling.

Details

Package:	dils
Type:	Package
Version:	0.4
Date:	2013-07-16
License:	MIT + file LICENSE

Start with a table (data.frame, tab-delimited file, database) where each row/record represents a link between two nodes (a dyad) in a directed or undirected network and each column represents a different relationship between the two nodes, ie. each column is a network. DILS combines these columns/networks into a single network that is a weighted sum of the constituent networks. The resulting DILS network uses information from all of the constituent networks and contains more information than any of the constituent networks. The output is a data.frame of DILS scores for each dyad, therefore is a single network ready for analysis using **igraph** or other social network analysis (SNA) tools.

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

"Discovering Missing Links in Networks Using Similarity Measures", Hung-Hsuan Chen, Liang Gou, Xiaolong (Luke) Zhang, C. Lee Giles. 2012.

`EdgelistFill` *Ensure an edgelist has all dyads and a column of weights.*

Description

Given a matrix or data.frame edgelist, fills in all possible edges not already listed with a weight of 0 or the value of `fillBlanksWith`.

Usage

```
EdgelistFill(elist, fillBlanksWith = 0, nodelist)
```

Arguments

`elist`	data.frame or matrix, see 'Details' for formatting assumptions.
`fillBlanksWith`	numeric, default weight for edges not already listed in elist.
`nodelist`	character, optional list of node names.

Details

The `elist` can be either a data.frame or a matrix with either 2 or 3 columns. Each row is an edge. The first column lists the node the edge is 'from' and the second column lists the node the edge is 'to'. If there is a third column, it lists the weight of the edge.

Value

data.frame, full list of all possible edges with weights for each in third column.

Author(s)

Stephen R. Haptonstahl `<srh@haptonstahl.org>`

References

`https://github.com/shaptonstahl/`

Examples

```
g <- erdos.renyi.game(10, 2/10)
EdgelistFill(get.edgelist(g))
EdgelistFill(get.edgelist(g), nodelist=1:10)

E(g)$weight <- runif(ecount(g))
el <- cbind(get.edgelist(g), E(g)$weight)
EdgelistFill(el)
EdgelistFill(el, nodelist=1:10)
```

`EdgelistFromAdjacency` *Converts adjacency matrix to filled edgelist*

Description

Given the adjacency matrix for a network returns a data.frame listing all possible edges and the weights for each edge.

Usage

```
EdgelistFromAdjacency(A,
  nodelist = paste("node", 1:nrow(A), sep = ""))
```

Arguments

`A`	matrix, see 'Details' for formatting assumptions.
`nodelist`	character, optional list of node names.

Details

This assumes that the row of the adjacency matrix indicates the node the edge is coming 'from', the column represent the node the edge is going 'to', and the value in the adjacency matrix is the weight given to the edge.

Value

data.frame, full list of all possible edges with weights for each in third column.

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

https://github.com/shaptonstahl/

See Also

EdgelistFromIgraph

Examples

```
n <- 10
A <- matrix(rnorm(n*n), nrow=n)
A
EdgelistFromAdjacency(A)

n <- 100
A <- matrix(rnorm(n*n), nrow=n)
A
EdgelistFromAdjacency(A)

n <- 500
A <- matrix(rnorm(n*n), nrow=n)
A
## Not run: EdgelistFromAdjacency(A)
```

EdgelistFromIgraph *Converts an igraph to filled edgelist*

Description

Given an igraph object for a network returns a data.frame listing all possible edges and the weights for each edge.

Usage

```
EdgelistFromIgraph(g, useWeight = FALSE)
```

Arguments

`g`	igraph, from igraph package.
`useWeight`	logical, Should E(g)$weight be used as the weights for the edges?

Details

This function is preferred to the igraph function `get.edgelist` because `get.edgelist` only returns rows for edges that have non-zero weight and does not return weights, if present.

Value

data.frame, full list of all possible edges with weights for each in third column.

Author(s)

Stephen R. Haptonstahl <`srh@haptonstahl.org`>

References

`https://github.com/shaptonstahl/`

See Also

`EdgelistFromAdjacency`

Examples

```
g <- erdos.renyi.game(10, 2/10)
EdgelistFromIgraph(g)

V(g)$name <- letters[1:vcount(g)]
EdgelistFromIgraph(g)

E(g)$weight <- runif(ecount(g))
EdgelistFromIgraph(g, useWeight=TRUE)
```

`GenerateDilsNetwork` *Combine multiple networks into a single weighted network.*

Description

Use ScalablePCA to recover optimal weights for each network, then calculate the weighted average across networks for each edge.

Usage

```
GenerateDilsNetwork(x, subsample = 10000,
  n.subsamples = 1000, ignore.cols, use.cols,
  progress.bar = FALSE)
```

Arguments

`x`	data.frame, data over which to run PCA
`subsample`	numeric or logical, If an integer, size of each subsample. If FALSE, runs PCA on entire data set.
`n.subsamples`	numeric, number of subsamples.
`ignore.cols`	numeric, indices of columns not to include
`use.cols`	numeric, indices of columns to use
`progress.bar`	logical, if TRUE then progress in running subsamples will be shown.

Value

vector, named vector of component weights for first dimension of principal component analysis (see example for comparison to `prcomp`)

Author(s)

Stephen R. Haptonstahl <`srh@haptonstahl.org`>

References

`https://github.com/shaptonstahl/`

See Also

`prcomp`

Examples

```
data(iris)        # provides example data
GenerateDilsNetwork(iris, subsample=10, use.cols=1:4)
GenerateDilsNetwork(iris, subsample=10, ignore.cols=5)
```

GetSampleFromDataFrame

Short description of the function

Description

A longer description of the function. This can be perhaps a paragraph, perhaps more than one.

Usage

```
GetSampleFromDataFrame(n, x)
```

Arguments

`n`	numeric, size of sample.
`x`	data.frame, data whose rows will be sampled.

Value

data.frame, size n random subset of the rows of x

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

https://github.com/shaptonstahl/

See Also

ScalablePCA, GetSampleFromFile, GetSampleFromFile

Examples

```
data(iris)   # provides example data
x <- GetSampleFromDataFrame(10, iris)
```

GetSampleFromDb *Sample from the rows of a (possibly large) database table*

Description

Accesses a dataabase table directly. Returns a data.frame whose rows are the sample.

Usage

```
GetSampleFromDb(n, db)
```

Arguments

n	numeric, size of sample to be taken.
db	connection, connection to the database table containing the data.

Value

data.frame, size n random subset of the rows of filename

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

https://github.com/shaptonstahl/

See Also

ScalablePCA, GetSampleFromDataFrame, GetSampleFromFile

Examples

```
## Not run: x <- GetSampleFromDb(10, my.db)
```

GetSampleFromFile *Sample from the rows of a (possibly large) text file*

Description

Reads a large text file in batches, keeping the rows to be included in the sample. Returns a data.frame whose rows are the sample.

Usage

```
GetSampleFromFile(n, out.of, filename)
```

Arguments

n	numeric, size of sample to be taken.
out.of	numeric, number of rows in the data set not including the header.
filename	character, name of the file containing the data. This must be a tab-delimited file with a header row formatted per the default options for read.delim.

Value

data.frame, size n random subset of the rows of filename

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

https://github.com/shaptonstahl/

See Also

ScalablePCA, GetSampleFromDataFrame, GetSampleFromDb

Examples

```
## Not run: x <- GetSampleFromFile(10, 150, "folder/containing/data.txt")
```

RelationStrengthSimilarity

Calculate the RSS from one node to another

Description

A longer description of the function. This can be perhaps a paragraph, perhaps more than one.

Usage

```
RelationStrengthSimilarity(xadj, v1, v2, radius)
```

Arguments

xadj	numeric matrix, then description of arg1.
v1	numeric Object type, then description of arg2.
v2	numeric Object type, then description of arg2.
radius	numeric, length of longest path examined from v1 to v2.

Details

If v1 and v2 are specified, this returns the RSS from v1 to v2. If not, it calculates the RSS scores for all dyads in the network.

Value

numeric, Relation Strength Similarity score(s).

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

"Discovering Missing Links in Networks Using Similarity Measures", Hung-Hsuan Chen, Liang Gou, Xiaolong (Luke) Zhang, C. Lee Giles. 2012.

https://github.com/shaptonstahl/

See Also

ScalablePCA

Examples

```
M <- matrix(0, nrow=6, ncol=6)
M[1,2] <- M[2,1] <- 1
M[2,3] <- M[3,2] <- 1
M[3,4] <- M[4,3] <- 1
M[4,5] <- M[5,4] <- 1
M[5,6] <- M[6,5] <- 1
M[6,1] <- M[1,6] <- 1
M[1,4] <- M[4,1] <- 1
```

```
M
RelationStrengthSimilarity(xadj=M, v1=5, v2=6, radius=1)
RelationStrengthSimilarity(xadj=M, v1=5, v2=6, radius=2)
RelationStrengthSimilarity(xadj=M, v1=5, v2=6, radius=3)
RelationStrengthSimilarity(xadj=M, v1=5, v2=6, radius=4)
RelationStrengthSimilarity(xadj=M, radius=2)
## Not run: RelationStrengthSimilarity(xadj=M, radius=3)
```

RssCell *Calculate the RSS from one node to another*

Description

This is a helper function for RelationStrengthSimilarity that returns the RSS for a single directed dyad.

Usage

```
RssCell(xadj, v1, v2, radius)
```

Arguments

xadj	numeric matrix, adjacency matrix where the [i,j] entry gives the strength of the link from node i to node j.
v1	numeric, index of the 'from' node.
v2	numeric, index of the 'to' node.
radius	numeric, length of longest path examined from v1 to v2.

Details

This is an internal function. There are no guardians and it assumes that the adjacency matrix xadj has had zeros entered on the diagonal and then each row divided by the row mean.

Value

numeric, the Relation Strength Similarity score from v1 to v2.

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

"Discovering Missing Links in Networks Using Similarity Measures", Hung-Hsuan Chen, Liang Gou, Xiaolong (Luke) Zhang, C. Lee Giles. 2012.

https://github.com/shaptonstahl/

See Also

RelationStrengthSimilarity

Examples

```
M <- matrix(0, nrow=6, ncol=6)
M[1,2] <- M[2,1] <- 1
M[2,3] <- M[3,2] <- 1
M[3,4] <- M[4,3] <- 1
M[4,5] <- M[5,4] <- 1
M[5,6] <- M[6,5] <- 1
M[6,1] <- M[1,6] <- 1
M[1,4] <- M[4,1] <- 1
M
M <- sweep(M, 1, rowMeans(M), "/")
M
RssCell(xadj=M, v1=5, v2=6, radius=1)
RssCell(xadj=M, v1=5, v2=6, radius=2)
RssCell(xadj=M, v1=5, v2=6, radius=3)
RssCell(xadj=M, v1=5, v2=6, radius=4)
```

RssThisRadius *Calculate part of the RSS from one node to another*

Description

This is a helper function for RelationStrengthSimilarity that returns the component of RSS contributed by paths of one particular length r.

Usage

```
RssThisRadius(x, v1, v2, r, prepped = FALSE)
```

Arguments

x	numeric matrix, adjacency matrix where the [i,j] entry gives the strength of the link from node i to node j.
v1	numeric, index of the 'from' node.
v2	numeric, index of the 'to' node.
r	numeric, length of paths examined from v1 to v2.
prepped	logical, whether or not the adjacency matrix x has had zeros entered on the diagonal and each row divided by the row sum.

Value

numeric, the part of the Relation Strength Similarity score from v1 to v2 contributed by paths of length r.

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

"Discovering Missing Links in Networks Using Similarity Measures", Hung-Hsuan Chen, Liang Gou, Xiaolong (Luke) Zhang, C. Lee Giles. 2012.

`https://github.com/shaptonstahl/`

See Also

`RelationStrengthSimilarity`

Examples

```
M <- matrix(0, nrow=6, ncol=6)
M[1,2] <- M[2,1] <- 1
M[2,3] <- M[3,2] <- 1
M[3,4] <- M[4,3] <- 1
M[4,5] <- M[5,4] <- 1
M[5,6] <- M[6,5] <- 1
M[6,1] <- M[1,6] <- 1
M[1,4] <- M[4,1] <- 1
M
RssThisRadius(x=M, v1=5, v2=6, r=1)
RssThisRadius(x=M, v1=5, v2=6, r=2)
RssThisRadius(x=M, v1=5, v2=6, r=3)
RssThisRadius(x=M, v1=5, v2=6, r=4)
```

`ScalablePCA` *Perform Principal Component Analysis on a large data set*

Description

Runs 'prcomp' on subsamples of the data set and compiles the results for the first dimension.

Usage

```
ScalablePCA(x, filename = NULL, db = NULL,
  subsample = 10000, n.subsamples = 1000, ignore.cols,
  use.cols, progress.bar = FALSE)
```

Arguments

`x`	data.frame, data over which to run PCA
`filename`	character, name of the file containing the data. This must be a tab-delimited file with a header row formatted per the default options for `read.delim`.
`db`	Object type, database connection to table containing the data (NOT IMPLEMENTED)
`subsample`	numeric or logical, If an integer, size of each subsample. If FALSE, runs PCA on entire data set.
`n.subsamples`	numeric, number of subsamples.
`ignore.cols`	numeric, indices of columns not to include
`use.cols`	numeric, indices of columns to use
`progress.bar`	logical, if TRUE then progress in running subsamples will be shown.

Value

vector, named vector of component weights for first dimension of principal component analysis (see example for comparison to prcomp)

Author(s)

Stephen R. Haptonstahl <srh@haptonstahl.org>

References

https://github.com/shaptonstahl/

See Also

prcomp

Examples

```
data(iris)        # provides example data
prcomp(iris[,1:4], center=FALSE, scale.=FALSE)$rotation[,1]
ScalablePCA(iris, subsample=10, use.cols=1:4)
ScalablePCA(iris, subsample=10, ignore.cols=5)
```

Index