Package 'fcd'

July 2, 2014

Type Package

Title Fused Community Detection

Version 0.1

Date 2013-12-27

Author Yang Feng, Richard J. Samworth and Yi Yu

Maintainer Yi Yu <y.yu@statslab.cam.ac.uk>

Depends R (>= 2.15.1), glmnet, MASS, combinat

Description

Efficient procedures for community detection in network studies, especially for sparse networks with not very obvious community structure. The algorithms impose penalties on the differences of the coordinates which represent the community labels of the nodes.

License GPL (>= 2.0)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-15 17:52:32

R topics documented:

`fcd` *Fused community detection.*

Description

Get the fused community detection path object.

Usage

```
fcd(A, K = 2, nlambda = 1e+3, lambda.min.ratio = 1e-05, alpha = 0.8, scale = FALSE)
fcd.start(A, K = 2, nlambda = 1000, lambda.min.ratio = 1e-05, alpha = 0.8, scale = FALSE)
```

Arguments

`A`	input matrix – adjacency matrix of an observed graph based on the non-isolated nodes, of dimension `n.noniso` x `n.noniso`, where `n.noniso` is the number of the non-isolated nodes.
`K`	input integer – the pre-specified number of communities, with the default value 2.
`nlambda`	The number of lambda values - default is 1000.
`lambda.min.ratio`	Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero) - default is 1e-05.
`alpha`	The elasticnet mixing parameter - default is 0.8.
`scale`	The logic indicator of whether scaling the resulting matrix - default is FALSE.

Value

`beta.combind`	the estimators along the path.
`iso.seq`	a vector of indices of the isolated nodes.
`cluster.list`	list of clustering results.
`criteria.list`	list of criteria values.
`final.ratio.cluster`	the final estimator of the community labels according to the ratio cut criterion.
`ratio.location`	the location of the chosen estimator on the path according to the ratio cut criterion.
`final.normalised.cluster`	the final estimator of the community labels according to the normalised cut criterion.
`normalised.location`	the location of the chosen estimator on the path according to the normalised cut criterion.
`lambda.list`	the lambda sequence used for the path.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Feng, Y., Samworth, R. J. and Yu, Y., Fused Community Detection, manuscript.

`fcd.cluster` *Clustering the estimators along the path.*

Description

To use k-means to cluster the estimators along the path and get the community labels.

Usage

```
fcd.cluster(obj, K = 2)
```

Arguments

`obj`	it is an object generated by `fcd.start`,
`K`	input integer – the pre-specified number of communities, with the default value 2.

Value

an array of community labels list, of dimension number of non-isolated nodes x number of effective estimators. Each entry has value from 1 to K, as an index of the community label. Notice, the community labels are usually permutation-invariant.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Yang Feng, Richard J. Samworth and Yi Yu, Fused Community Detection, manuscript.

`fcd.criteria` *The criterion values based along the path.*

Description

The ratio cut and normalised cut values along the path.

Usage

```
fcd.criteria(A, fcd.cluster, K = 2, iso.seq)
```

Arguments

`A`	input matrix – adjacency matrix of an observed graph based on the non-isolated nodes, of dimension `n.noniso` x `n.noniso`, where `n.noniso` is the number of the non-isolated nodes.
`fcd.cluster`	a list of vectors, with each vector as the estimator of the community labels of the non-isolated nodes in the network, of dimension `n.noniso`, values taken from 1 to K, where K is the number of communities.
`K`	the number of the communities, with 2 as the default value.
`iso.seq`	a vector of the indices of those isolated nodes in the graph. If it is missing, `obj` should be offered.

Value

`ratio.list`	a list of ratio cut values for the estimator path.
`normalised.list`	a list of normalised cut values for the estimator path.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Yang Feng, Richard J. Samworth and Yi Yu, Fused Community Detection, manuscript.

fcd.trans — *The graph based penalty transformation matrix*

Description

Get the graph based penalty transformation matrix.

Usage

```
fcd.trans(A)
```

Arguments

A	input matrix – adjacency matrix of an observed graph based on the non-isolated nodes, of dimension n.noniso x n.noniso, where n.noniso is the number of the non-isolated nodes.

Value

the graph based penalty transformation matrix, of dimension $|E| \times n$, where $|E|$ is the number of edges in the graph and n is the number of nodes.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Yang Feng, Richard J. Samworth and Yi Yu, Fused Community Detection, manuscript.

generate — *generate adjacency matrix of stochastic blockmodel, degree-corrected block model or cockroach graph model.*

Description

To generate an adjacency matrix of stochastic blockmodel, degree-corrected block model or cockroach graph model.

Usage

```
gen.sbm(n, theta.in, theta.bw, K, seed)
gen.dcbm(n, theta.in, theta.bw, theta, K, seed)
gen.cr(n1)
```

Arguments

n1	input integer – one quarter of the number of nodes in the graph.
n	input integer – the number of nodes in EACH community.
theta.in	input real number, which is the probability of a within community edge.
theta.bw	input real number, which is the probability of a between community edge.
theta	input vector, of dimension number of nodes in ALL communities, with each entry equal to the individual effect of each node.
K	input integer – the number of communities. Currently, only values 2, 3 and 4 are implemented.
seed	input integer – the random seed you can set.

Value

an adjacency matrix.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Yang Feng, Richard J. Samworth and Yi Yu, Community Detection via Fused Principal Component Analysis, manuscript. Holland, P.W., Laskey, K.B. and Leinhardt, S., 1983. Stochastic block models: first steps. Social Networks 5, 109-137. Karrer, B. and Newman, M.E.J., 2011. Stochastic blockmodels and community structure in networks. Physical Review E 83, 016107.

Examples

```
A1 = gen.sbm(n = 10, theta.in = 0.3, theta.bw = 0.1, K = 2, seed = 2)
A2 = gen.dcbm(n = 10, theta.in = 0.3, theta.bw = 0.1,
theta = seq(from = 0.1, to = 0.5, length.out = 20), K = 2, seed = 2)
A3 = gen.cr(n1 = 10)
```

---

get.cluster *Final estimators of the community labels*

Description

Get the final estimator of the community labels along the path, according to ratio cut or normalised cut criterion.

Usage

```
get.cluster(A, iso.seq, criteria.list, clusters.list)
```

Arguments

A	input matrix – the adjacency matrix of the observed graph. Notice, both isolated and non-isolated nodes are included.
iso.seq	a vector of the indices of the isolated nodes.
criteria.list	the criteria value lists along the path. Notice, only meaningful values are input. For details, please see the listed paper.
clusters.list	the estimators of the community labels along the path.

Value

final.ratio.cluster	the final estimator of the community labels according to the ratio cut criterion.
ratio.location	the location of the chosen estimator on the path according to the ratio cut criterion.
final.normalised.cluster	the final estimator of the community labels according to the normalised cut criterion.
normalised.location	the location of the chosen estimator on the path according to the normalised cut criterion.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Yang Feng, Richard J. Samworth and Yi Yu, Fused Community Detection, manuscript.

---

isolate *Isolated nodes collection*

Description

Get the indices of the isolated nodes in the graph.

Usage

```
isolate(A)
```

Arguments

A	input matrix – adjacency matrix of an observed graph, of dimension n-nodes x n-nodes.

Value

isolate	the indices of the isolated nodes in the graph with adjacency matrix A.
nonisolate	the indices of the non-isolated nodes in the graph with adjacency matrix A.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Yang Feng, Richard J. Samworth and Yi Yu, Fused Community Detection, manuscript.

---

laplacian *Laplacian matrix*

Description

Get the Laplacian matrix of an observed graph. Both unnormalised and symmetric normalised Laplacian matrices are included.

Usage

```
laplacian(A, normalised = FALSE)
```

Arguments

A	input matrix – adjacency matrix of an observed graph, of dimension n-nodes x n-nodes.
normalised	whether the Laplacian matrix is normalised or not.

Value

if `normalised = F`, an unnormalised Laplacian matrix is returned, i.e. $L = D - A$; if `normalised = T`, a symmetric normalised Laplacian matrix is returned, i.e. $L = D^{-1/2}(D - A)D^{-1/2}$. D is the degree diagonal matrix, with diagonal entries $d_i = \sum_{j=1}^{n} A_{ij}$.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Yang Feng, Richard J. Samworth and Yi Yu, Fused Community Detection, manuscript.

mis.cluster *Mis-clustered nodes for balanced designed network.*

Description

Get the mis-clustered nodes number.

Usage

```
mis.cluster(x, K, n)
```

Arguments

x	estimated labels.
K	community number.
n	nodes size for each community.

Value

perms	the permutation coincides with the truth.
mis.cluster	the mis-clustered nodes number.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

References

Yang Feng, Richard J. Samworth and Yi Yu, Fused Community Detection, manuscript.

spectral.clustering *Spectral clustering and its variant.*

Description

Use spectral clustering and its variant for community detection in a network.

Usage

```
spectral.clustering(A, normalised = TRUE, score = FALSE, K = 2, adj = FALSE)
```

Arguments

A	input matrix – adjacency matrix of an observed graph based on the non-isolated nodes, of dimension n.noniso x n.noniso, where n.noniso is the number of the non-isolated nodes.
normalised	a logic variable indicating whether normalised Laplacian matrix is used in community detection.
score	a logical variable indicating whether the RoE is used.
K	input integer – the pre-specified number of communities, with the default value 2.
adj	adjacency matrix is used or not.

Value

a vector of labels are returned.

Author(s)

Yang Feng, Richard J. Samworth and Yi Yu

Index