

Package ‘blkergm’

August 27, 2014

Version 1.1

Date 2014-08-26

Title Fitting block ERGM given the block structure on social networks

LinkingTo ergm

Depends network (>= 1.7-1), ergm (>= 3.1-0), statnet.common (>= 3.1-0)

Description This package is an extension to the “ergm” package which implements the block ergms.

License GPL-3

URL <http://statnet.org>

Author Xiaolin Yang [aut, cre], Stephen E. Fienberg [aut], Alessandro Rinaldo [aut], Han Liu [aut], Michael Rosenblum [aut]

Maintainer Xiaolin Yang <xyang@stat.cmu.edu>

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-08-27 08:33:25

R topics documented:

blkergm-package	2
blkergm	3
blkergm-terms	4
InitErgmTerms	5
zachary	6

Index	7
--------------	----------

blkergm-package	<i>An extension of "ergm" package for handling models with block structure</i>
-----------------	--

Description

This package is an extension to the package **ergm** which takes the block information into consideration for ERGMs. The implementation is based on the **ergm.userterms** package. In this package, several (block) models are considered. They include Beta model, block Erdos-Renyi, block Beta, block Markov random graph models under different assumptions. The first three models are those assuming independence between edges. The block Markov random graph models assume dependence within blocks, independence or dependence on between block edges. The package implements these models by taking different terms in the model. See [blkergm.terms](#) for explanation of those terms. Also see **statnet**, **ergm** and **network** for more information on how to use ERGMs to analyse social network data.

Details

Package:	blkergm
Type:	Package
Version:	1.0
Date:	2013-09-16
License:	GPL(>= 2)

References

Hunter DR, Goodreau SM, Handcock MS (2011) [ergm.userterms](#): A Template Package for Extending **statnet**.

See Also

statnet, network, ergm, ergm-terms

Examples

```
## Not run:
data(zachary)
summary(zachary~blktriangle(c(16,34),1)+blktriangle(c(16,34),2))
fit <- ergm(zachary~blkedges(c(16,34),1:3), MPLEonly=TRUE)
summary(fit)

## End(Not run)
```

blkerGM

*The main function to fit block ERGMs***Description**

This is the main function to fit block ERGMs.

Usage

```
blkerGM(formula, offset.coef=NULL, target.stats=NULL, eval.loglik=TRUE,
estimate=c("MLE", "MPLE"), control=control.ergm(), verbose=FALSE, ...)
```

Arguments

formula	The formula is expressed as <code>nw~blkerGM.terms</code>
offset.coef	See ergm for more explanation
target.stats	See ergm for more explanation
eval.loglik	See ergm for more explanation
estimate	The estimation methods include "MLE" and "MPLE" for ERGMs, they both work for block ERGMs.
control	See ergm for more explanation
verbose	See ergm for more explanation
...	Additional arguments, to be passed to lower-level functions.

References

Hunter DR, Goodreau SM, Handcock MS (2011) **ergm.userterms**: A Template Package for Extending **statnet**.

See Also

statnet, network, ergm, ergm-terms

Examples

```
## Not run:
data(zachary)
summary(zachary~blkdegseq(c(0,16,34),2))
summary(zachary~blktriangle(c(16,34),2))
#block Beta model
fit <- blkerGM(zachary~blkdegseq(c(0,16,34),1)+blkdegseq(c(0,16,34),2)+blkedges(c(16,34),2))
summary(fit)

## End(Not run)
```

Description

This package contains R functions and terms used for fitting block ERGMs. Depending on different dependence and homogeneous assumptions on the model, the models include different terms. The package considers these assumptions and codes the terms using the "ergm.userterms" package to give an easy user interface to fit block models. The meaning and usage of these terms are listed below.

Terms for the block ergms

`blkedges(blockbdr, blockid)` *The number of edges in a given block with blockid:* Assume the rows and columns of the network adjacency matrix is ordered by which block they belong to. `blockbdr` argument is a vector specifying the boundary of blocks. For example, `(5,12,17)` specifies the boundary of blocks in the way that there are 3 blocks in this network and nodes 1-5 are in block 1, 6-12 are in block 2 and 13-17 are in block 3. `blockid` specifies the block which we want to retrieve the number of edges from. Take the above example, there are 3 blocks in that network, so that the number of between blocks is 3. They are block 1-2, block 1-3 and block 2-3. These blocks are ordered as 1,1-2,1-3,2,2-3,3. Then `blkedges(c(5, 12, 17), blockid=3)` means the number of edges in block 1-3 with block structure `(5,12,17)` explained above.

`blktriangle(blockbdr, blockid)` *The number of triangles in a given block with blockid:* The `blockbdr` is the same as that in `blkedges`. This term aims to return the number of triangles in a given block with `blockid`. Different from the `blockid` in `blkedges`, it only computes the number of triangles within a block. In the above example, there are 3 blocks. The `blockid` only takes value of 1, 2 and 3. More specifically, `blktriangle((5, 12, 17), 3)` means the number of triangles in block 3 in this network.

`blkstar(k, blockbdr, blockid)` *The number of k-stars in a given block with blockid:* `blockbdr` and `blockid` are the same as in `blktriangle`. `k` can be a vector or scalar specifying "k"-stars. Note that, this term only returns the number of k-stars within each block.

`blktriangle2(blockbdr, blockid)` *The number of triangles within each block with blockid and those formed by edges between this block and other blocks:* `blockid` ranges from 1 to 3 in the above example.

`blkstar2(k, blockbdr, blockid)` *The number of k-stars within each block with blockid and those formed by edges between this block and other blocks:* `k` can be a vector or a scalar specifying "k"-stars. `blockid` ranges from 1 to 3 in the above example.

`blkstar3(k, blockbdr, blockid)` *The number of k-stars formed by edges between two blocks* `blockid` can only take values for those off diagonal blocks. In the above example, `blockid` can take 2, 3 and 5.

`degseq` *Degree sequence of a given network* The number of values returned by this term is equal to the number of nodes in the network. They correspond to the nodal degrees of all nodes.

`blkdegseq(blockbdr,blockid)` *The degree sequence in a given block with blockid: blockid ranges from 1-3 in the above example. Please add one more 0 in front of blockbdr vector to indicate the starting position of the first node.*

Author(s)

Xiaolin Yang Maintainer: Who to complain to <xyang@stat.cmu.edu> Xiaolin Yang

References

Hunter DR, Goodreau SM, Handcock MS (2011) **ergm.userterms**: A Template Package for Extending **statnet**.

See Also

statnet, network, ergm, ergm-terms

Examples

```
## Not run:
data(zachary)
summary(zachary~degseq)
summary(zachary~blkdegseq(c(0,16,34),1)+blkdegseq(c(0,16,34),2))
summary(zachary~blkedges(c(16,34),1:3))
summary(zachary~blktriangle(c(16,34),1:2))
summary(zachary~blkstar(1:3,c(16,34),1:2))
summary(zachary~blktriangle2(c(16,34),1:2))
summary(zachary~blkstar2(1:3,c(16,34),1:2))
summary(zachary~blkstar3(1:3,c(16,34),2))

## End(Not run)
```

InitErgmTerms

User-defined terms used in Exponential Family Random Graph Models

Description

These functions are required for the interface between ergm package and ergm.userterms package. These functions are written for the purpose of initialize the terms in particular models. See [blkergm-terms](#) for more information.

References

Hunter DR, Goodreau SM, Handcock MS (2011) **ergm.userterms**: A Template Package for Extending **statnet**.

See Also

statnet, network, ergm, ergm-terms

zachary

Zachary Karate Club social network data

Description

[zachary.club](#) is Social network of friendships between 34 members of a karate club at a US university in the 1970. It is an undirected network without attributes of vertices.

Usage

```
data(zachary)
```

Format

zachary is a [network](#) object with 34 vertices and 78 undirected edges. To obtain additional summary information about it, type `summary(zachary)`.

References

W. W. Zachary. *An information flow model for conflict and fission in small groups*, *Journal of Anthropological Research* 33, 452-473 (1977).

See Also

[network](#), [plot.network](#), [ergm](#), [zachary.club](#)

Index

- *Topic **block models**
 - blkergm-terms, 4
- *Topic **datasets**
 - zachary, 6
- *Topic **ergm-terms**
 - blkergm-terms, 4
- *Topic **models**
 - blkergm, 3
 - blkergm-package, 2
 - InitErgmTerms, 5
- *Topic **package**
 - blkergm, 3
 - blkergm-package, 2
 - InitErgmTerms, 5
- blkdegseq (blkergm-terms), 4
- blkedges (blkergm-terms), 4
- blkergm, 3
- blkergm-package, 2
- blkergm-terms, 4, 5
- blkergm.terms, 2, 3
- blkergm.terms (blkergm-terms), 4
- blkkstar (blkergm-terms), 4
- blkkstar2 (blkergm-terms), 4
- blkkstar3 (blkergm-terms), 4
- blktriangle (blkergm-terms), 4
- blktriangle2 (blkergm-terms), 4
- blktriangle3 (blkergm-terms), 4
- degseq (blkergm-terms), 4
- ergm, 3, 6
- ergm.userterms, 2
- ergm.userterms (InitErgmTerms), 5
- InitErgmTerm.blkdegseq (InitErgmTerms), 5
- InitErgmTerm.blkedges (InitErgmTerms), 5
- InitErgmTerm.blkkstar (InitErgmTerms), 5
- InitErgmTerm.blkkstar2 (InitErgmTerms), 5
- InitErgmTerm.blkkstar3 (InitErgmTerms), 5
- InitErgmTerm.blktriangle (InitErgmTerms), 5
- InitErgmTerm.blktriangle2 (InitErgmTerms), 5
- InitErgmTerm.blktriangle3 (InitErgmTerms), 5
- InitErgmTerm.degseq (InitErgmTerms), 5
- InitErgmTerm.mindegree (InitErgmTerms), 5
- InitErgmTerm.nedges (InitErgmTerms), 5
- InitErgmTerm.nmkstar (InitErgmTerms), 5
- InitErgmTerm.nmtriangle (InitErgmTerms), 5
- InitErgmTerms, 5
- network, 6
- nedges (blkergm-terms), 4
- nmkstar (blkergm-terms), 4
- nmtriangle (blkergm-terms), 4
- plot.network, 6
- zachary, 6
- zachary.club, 6