## Package ‘CIDnetworks’

### March 11, 2015

**Type**: Package  
**Title**: Generative Models for Complex Networks with Conditionally Independent Dyadic Structure  
**Version**: 0.8.0  
**Date**: 2015-03-10  
**Description**: Generative models for complex networks with conditionally independent dyadic structure. Now supports directed arcs!  
**License**: GPL (>= 3)  
**Depends**: R (>= 3.0.0)  
**Imports**: mvtnorm, msm, methods, Rcpp (>= 0.11.0), igraph, numDeriv, pbivnorm  
**LinkingTo**: Rcpp  
**Author**: Samrachana Adhikari [aut], Beau Dabbs [cre, aut], Brian Junker [aut], Mauricio Sadinle [aut], Tracy Sweet [aut], A.C. Thomas [aut]  
**Maintainer**: Beau Dabbs <bdabbs@andrew.cmu.edu>  
**NeedsCompilation**: yes  
**Repository**: CRAN  
**Date/Publication**: 2015-03-11 23:15:52

### R topics documented:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CIDnetworks-package</td>
<td>2</td>
</tr>
<tr>
<td>CIDnetworks-classes</td>
<td>2</td>
</tr>
<tr>
<td>CIDnetworks-data</td>
<td>4</td>
</tr>
<tr>
<td>CIDnetworks-gibbs</td>
<td>5</td>
</tr>
<tr>
<td>CIDnetworks-helpers</td>
<td>8</td>
</tr>
<tr>
<td>CIDnetworks-master</td>
<td>9</td>
</tr>
</tbody>
</table>

**Index**: 11
CIDnetworks-package  

Model Complex Networks With Multiple Components

Description

The CIDnetworks package allows a user to construct hierarchical models of complex networks using one of many different modes for latent structure.

Details

Package: CIDnetworks
Type: Package
Version: 0.8.0
Date: 2014-03-10
License: GPLv3

Author(s)

Beau Dabbs, Brian Junker, Mauricio Sadinle, Tracy Sweet, A.C. Thomas
Maintainer: Beau Dabbs <bdabbs@andrew.cmu.edu>

Examples

```r
data(dolphins)
model.plain <- CID.Gibbs (input=dolphins, burnin=10, draws=10)
```

CIDnetworks-classes  

Classes of CIDnetwork subcomponents

Description

Each of the reference classes used as components in the CIDnetworks methodology.

Usage

BETA(...)
EdgeCOV (... , cov.type="Edge")
SenderCOV (...)
ReceiverCOV (...)

CIDnetworks-classes

SendRecCOV (...)  
IdenticalCOV (...)  
HBM (...)  
LSM (...)  
LVM (...)  
MMSBM (...)  
SBM (...)  
SR (...)

Arguments

... Arguments passed to the respective classes.

cov.type Specifies the type of covariate effect. Edge specifies Edge-level covariate effects.

Details

Each of these functions can serve as a subcomponent in the main CIDnetwork class object. Information on the nodes, edge list, and so forth will be passed automatically by any routine creating a CID object. Options are generally provided by default. Arguments for each class:

- **BETA**: required: (nothing). Parameters: intercept.sr.
- **EdgeCOV**: required: covariates (matrix). Parameters: Corresponding coefficient vector coef.cov.
- **SenderCOV**: required: covariates (vector of length n.nodes). Parameters: Corresponding coefficient vector coef.cov.
- **ReceiverCOV**: required: covariates (vector of length n.nodes). Parameters: Corresponding coefficient vector coef.cov.
- **SendRecCOV**: required: covariates (vector of length n.nodes). Parameters: Corresponding coefficient vector coef.cov.
- **IdenticalCOV**: required: covariates (vector of length n.nodes). Parameters: Corresponding coefficient vector coef.cov.
- **HBM**: required: n.groups (single value). Parameters: block.value, membership (for nodes to blocks), tree.parent (for blocks).
- **LSM**: required: dimension (single value). Parameters: latent.space.pos.
- **LVM**: required: dimension (single value). Parameters: latent.space.pos.
- **MMSBM**: required: n.groups (single value). Parameters: b.vector, membership.edge, membership.node.
- **SBM**: required: n.groups (single value). Parameters: b.vector, membership.
- **SR**: required: (nothing). Parameters: intercept.sr.

Value

Each expression yields a Reference Class object for the respective submodel. If generate=TRUE, it will produce an outcome value for that class depending on its specific properties.
CIDnetworks-data

Author(s)
A.C. Thomas <act@acthomas.ca>

CIDnetworks-data CIDnetworks: Included Data Sets

Description
Data sets included with the CIDnetworks package.

Usage
- Lazega
- c.elegans
- dolphins
- prison

Format
Lazega: A list including six sociomatrices (three directed, three symmetrized) and one attributes matrix for the nodes.
c.elegans: An symmetric valued "sociomatrix" for the number of connections between neurons.
dolphins: A symmetric sociomatrix.
prison: (Thanks to UCINET) In the 1950s John Gagnon collected sociometric choice data from 67 prison inmates. All were asked, "What fellows on the tier are you closest friends with?" Each was free to choose as few or as many "friends" as he desired.

Author(s)
A.C. Thomas <act@acthomas.ca>

References
CIDnetworks-gibbs

CIDnetworks: Bayesian Inference

Description

Usage of CID Gibbs Samplers.

Usage

CID.Gibbs (input,
   outcome,
   components,
   class.outcome = NULL,
   fill.in.missing.edges = missing(outcome),
   new.chain = FALSE,
   draws = 100,
   burnin = -1,
   thin = 10,
   report = 100,
   auto.converge = FALSE,
   extend.max=10,
   extend.count=100,
   verbose=2,
   ...)

## S3 method for class 'CID.Gibbs'
print(x, ...)
## S3 method for class 'CID.Gibbs'
summary(object, ...)
## S3 method for class 'CID.Gibbs'
plot(x, ...)
## S3 method for class 'summary.CID.Gibbs'
print(x, ...)

likelihood.plot(x, ...)
intercept.plot(x, mode = c("standard","trace"), ...)
COV.plot(x, mode = c("standard","trace","scatterplot"), ...)
LSM.plot(x, ...)
SBM.plot(x, ...)
MMSBM.plot(x, ...)
SR.plot(x, ...)

network.plot (x, fitted.values=FALSE, ...)
sociogram.plot (x, component.color=0, vertexcolor, add.labels = TRUE, ...)
n.nodes(object)
edge.list(object)
is.net.directed(object)
net.density(object)
outcome(object)
node.names(object)
inDegree(object)
outDegree(object)
socio(object)
value.mat(CID.Gibbs.object, prob = TRUE)
value.mat.mean(object, prob = TRUE)
switcheroo(CID.Gibbs.object)

Arguments

input
An object containing information about the edges in a network. Must be one of the following classes: Matrix, CIDnetwork, or CID.Gibbs. If input is a square matrix, it is assumed to be a sociomatrix. Otherwise a matrix with 2 columns and a number of rows equal to the number of edges is required. Providing a CIDnetwork will use the associated edge.list. Providing a CID.Gibbs object will continue the MCMC chain from the last draw.

outcome
If an edgelist is provided as input and outcome is missing, the edges provided are assumed to be the ones in a binary sociomatrix. Otherwise, an outcome value must be specified for each edge in the edgelist, and any edges not provided are assumed to have no data.

CID.Gibbs.object
If desired, an existing CID.Gibbs output object can be loaded instead of a new network specification.

components
A list of sub-components, including (COV, HBM, LSM, LVM, MMSBM, SBM, SR).

class.outcome
One of "ordinal" (default, values from 0 to higher integers), "binary" (ordinal in 0 and 1) or "gaussian" (unbounded continuous values). Class is auto-detected if NULL remains in place.

fill.in.missing.edges
If TRUE, the edge list will be augmented with zeroes for all unspecified but possible edges. By default, if an outcome is specified, these edges will not be added.

new.chain
If a CID.Gibbs object is provided, the default value of FALSE will return both the old and new MCMC chain combined. A value of TRUE will drop the old chain completely.

draws
Number of draws to return.

burnin
Number of draws to burnin. A negative value will automatically determine burnin amount.
CIDnetworks-gibbs

thin    Amount of draws to thin the chain by.
report  Number of draws between reporting total draws so far.
auto.converge When true, a Geweke convergence test on log-likelihood to detect convergence.
extend.max Maximum number of times chain will be extended until it returns without converging.
extend.count Number of draws to extend chain by if convergence test fails
verbose Level of output to be displayed while running. A value of 0 will return little or no output. A value of 1 will only notify of warnings of misuse. A value of 2 will report progress of MCMC chain. A value of 3 or higher will report debugging values.
... Further arguments to be passed to the Gibbs sampler routine or the plot routine. See details for more.
x, object An object outputted from CID.Gibbs.
mode Controls which diagnostic plot is made.
fitted.values If TRUE, plots the fitted tie strength under the Gibbs sampler. If FALSE, plots the network outcomes as entered.
component.color If non-zero, colors the nodes in the sociogram according to the output of the Gibbs sampler.
vertexcolor User-passed vertex colors for sociogram.plot .
add.labels When true, node labels are included on nodes.
trace If selected, displays the Gibbs sampler trace plot for the intercept rather than a point and interval.
prob In value.mat, converts the linear predictor value to the probability of a binary edge.

Details

This is the main routine for running a Gibbs sampler on any of the CID models. See the vignettes for more information.

Value

CID.Gibbs outputs a list containing a CID object, the results of the Gibbs sampler, and the Deviance Information Criterion estimate for the Gibbs.

Author(s)

A.C. Thomas <act@acthomas.ca>
CIDnetworks-helpers  

*Functions to aid in the use and construction of CIDnetworks objects*

Description

Functions to aid in the use and construction of CIDnetworks objects

Usage

- `l.diag (nn)`
- `u.diag (nn)`
- `ordinal.maker (vec, cuts=quantile(vec, c(0.25, 0.5, 0.75)))`
- `unwrap.CID.Gibbs (gibbs.out)`
- `mat.cov.to.edge.list.cov (Xmat, n.nodes = dim(Xmat)[1], arc.list = make.arc.list(n.nodes))`

Arguments

- **nn**  
The number of rows in the square matrix for which we wish to extract the lower or upper diagonal matrix.
- **vec**  
The elements to be divided into ordinal categories.
- **cuts**  
The cut points at which to divide vec into ordinal categories. Default values separate vec into quartiles.
- **gibbs.out**  
The list object of draws from the Gibbs sampler. This re-sorts the object into a matrix form for easier consumption.
- **Xmat**  
A three-dimensional array of covariates, with n.nodes rows and columns. Each slice is a different covariate.
- **n.nodes**  
Number of nodes in network
- **arc.list**  
List of potential edges in network.

Details

These functions are included for the convenience of users of CIDnetworks. `l.diag` and `u.diag` provide the indices of a matrix to extract the lower and upper diagonal elements. `ordinal.maker` will turn any numeric vector into a series of ordinal integers for easy use in a CIDnetworks outcome. `Xmat` converts a sociomatrix-style array of covariates into one that can easily be used by the COV() component.

Author(s)

A.C. Thomas <act@acthomas.ca>
Forward Simulation and Declaration of CID Network Models

Description

Usage of the CID main class.

Usage

CID (input, outcome, n.nodes, intercept = 0, components,
     class.outcome="ordinal", fill.in.missing.edges=missing(outcome),
     generate=FALSE, verbose=2, ...)  
CID.generate (...)

Arguments

input A matrix object containing information about the edges in a network. If the input is a matrix with 2 columns the input is assumed to be a set of edges. If input is a square matrix, it is assumed to be a sociomatrix. If no input is provided, a new network is generated containing n.nodes.

outcome A set of outcome values for each edge in the edgelist given as input. If outcome is missing, the edgelist is assumed to represent observed edges in a binary network. outcome requires no value is a sociomatrix is provided.

n.nodes Number of nodes in generated networks

intercept Intercept to be used when generating new networks

components A list of components to use when generating new networks. Possible components include SR, SBM, MMSBM, LSM, LVM HBM EdgeCOV, SenderCOV, ReceiverCOV, SendRecCov, IdenticalCOV

class.outcome One of "ordinal" (default, values from 0 to higher integers), "binary" (ordinal in 0 and 1) or "gaussian" (unbounded continuous values). Class is auto-detected if NULL remains in place.

fill.in.missing.edges If TRUE, the edge list will be augmented with zeroes for all unspecified but possible edges. By default, if an outcome is specified, these edges will not be added.

generate When TRUE a new network with n.nodes nodes is generated as specified by the intercept and components passed to CID.

verbose Level of output to be displayed while running. A value of 0 will return little or no output. A value of 1 will only notify of warnings of misuse. A value of 2 will report progress of MCMC chain. A value of 3 or higher will report debugging values.

... Arguments passed to the master class.
Details

CID generates a CIDNetwork object that can be passed to many of the plotting and examination methods in the CIDNetworks package. Initial conversion of network data in standard edgelist or sociomatrix form allows easier manipulation within the CIDNetworks framework.

CID can also be used to simulate networks from any of the models that CIDNetworks can fit. CID.generate is a simple wrapper which calls CID with generate = TRUE.

Value

A CIDnetwork object.

Author(s)

A.C. Thomas <act@acthomas.ca>
## Index

**+Topic datasets**
- CIDnetworks-data, 4

**+Topic package**
- CIDnetworks-package, 2
- BETA (CIDnetworks-classes), 2
- BETAcid (CIDnetworks-classes), 2
- c.elegans (CIDnetworks-data), 4
- CID (CIDnetworks-master), 9
- CID.Gibbs (CIDnetworks-gibbs), 5
- CID.metric (CIDnetworks-helpers), 8
- CIDnetwork (CIDnetworks-master), 9
- CIDnetworks (CIDnetworks-package), 2
- CIDnetworks-classes, 2
- CIDnetworks-data, 4
- CIDnetworks-gibbs, 5
- CIDnetworks-helpers, 8
- CIDnetworks-master, 9
- CIDnetworks-package, 2
- COV.plot (CIDnetworks-gibbs), 5
- COVARIATEcid (CIDnetworks-classes), 2
- dolphins (CIDnetworks-data), 4
- edge.list (CIDnetworks-gibbs), 5
- EdgeCOV (CIDnetworks-classes), 2
- HBM (CIDnetworks-classes), 2
- HBMcid (CIDnetworks-classes), 2
- IdenticalCOV (CIDnetworks-classes), 2
- inDegree (CIDnetworks-gibbs), 5
- intercept.plot (CIDnetworks-gibbs), 5
- is.net.directed (CIDnetworks-gibbs), 5
- l.diag (CIDnetworks-helpers), 8
- Lazega (CIDnetworks-data), 4
- likelihood.plot (CIDnetworks-gibbs), 5
- LSM (CIDnetworks-classes), 2
- LSM.plot (CIDnetworks-gibbs), 5
- LSMcid (CIDnetworks-classes), 2
- LVM (CIDnetworks-classes), 2
- LVMcid (CIDnetworks-classes), 2
- mat.cov.to.edge.list.cov (CIDnetworks-helpers), 8
- MMSBM (CIDnetworks-classes), 2
- MMSBM.plot (CIDnetworks-gibbs), 5
- MMSBMcid (CIDnetworks-classes), 2
- n.nodes (CIDnetworks-gibbs), 5
- net.density (CIDnetworks-gibbs), 5
- network.plot (CIDnetworks-gibbs), 5
- node.names (CIDnetworks-gibbs), 5
- ordinal.maker (CIDnetworks-helpers), 8
- outcome (CIDnetworks-gibbs), 5
- outDegree (CIDnetworks-gibbs), 5
- plot.CID.Gibbs (CIDnetworks-gibbs), 5
- print.CID.Gibbs (CIDnetworks-gibbs), 5
- print.summary.CID.Gibbs (CIDnetworks-gibbs), 5
- prison (CIDnetworks-data), 4
- ReceiverCOV (CIDnetworks-classes), 2
- SBM (CIDnetworks-classes), 2
- SBM.plot (CIDnetworks-gibbs), 5
- SBMcid (CIDnetworks-classes), 2
- SenderCOV (CIDnetworks-classes), 2
- SendRecCOV (CIDnetworks-classes), 2
- socio (CIDnetworks-gibbs), 5
- sociogram.plot (CIDnetworks-gibbs), 5
- SR (CIDnetworks-classes), 2
- SR.plot (CIDnetworks-gibbs), 5
- SRcid (CIDnetworks-classes), 2
- summary.CID.Gibbs (CIDnetworks-gibbs), 5
- switcheroo (CIDnetworks-gibbs), 5
- u.diag (CIDnetworks-helpers), 8
unwrap.CID.Gibbs (CIDnetworks helpers), 8
value.mat (CIDnetworks gibbs), 5