Package ‘MEclustnet’

October 12, 2022

Title  Fit the Mixture of Experts Latent Position Cluster Model to Network Data

Version 1.2.2

Description
  Functions to facilitate model-based clustering of nodes in a network in a mixture of experts setting, which incorporates covariate information on the nodes in the modelling process. Isobel Claire Gormley and Thomas Brendan Murphy (2010) <doi:10.1016/j.stamet.2010.01.002>.

Imports  MASS, mvtnorm, nnet, ellipse, latentnet, vegan, e1071

Depends  R (>= 2.10)

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Isobel Claire Gormley [aut, cre], Thomas Brendan Murphy [aut]

Maintainer Isobel Claire Gormley <claire.gormley@ucd.ie>

Repository CRAN

Date/Publication 2019-10-10 11:50:05 UTC

R topics documented:

calclambda ................................................. 2
calcloglikelihood ........................................ 3
calcm ....................................................... 3
calcpis ..................................................... 4
formatting.covars ........................................ 5
invariant .................................................. 6
labelswitch ............................................... 7
lawyers.adjacency.advice ............................... 8
lawyers.adjacency.coworkers ........................... 9
**calclambda**

```r
lawyers.adjacency.friends .................................................. 9
lawyers.covariates .......................................................... 10
MEclustnet ................................................................. 10
plotMEclustnet ............................................................... 13
summaryMEclustnet ............................................................ 15
updatebeta ................................................................. 17
updateK ............................................................... 18
updatemu ............................................................ 19
updatesigma2 .......................................................... 20
updatetau ............................................................ 21
updatez ............................................................ 22
us.twitter.adjacency ......................................................... 23
us.twitter.covariates ......................................................... 24
```

**Index**

| calclambda | Title Compute mixing proportions |

**Description**

Function to compute the each observation’s mixing proportions which are modeled as a logistic function of their covariates.

**Usage**

```r
calclambda(tau, x.mix)
```

**Arguments**

- `tau`  
  A matrix of logistic regression coefficients, with G rows and number of columns equal to the number of covariates in the mixing proportions model plus 1, for the intercept.

- `x.mix`  
  A matrix of covariates in the mixing proportions model (including dummy variables for any factor covariates), with a column of 1’s appended at the front.

**Value**

An n x G matrix of mixing proportions.

**References**


**See Also**

```r
MEclustnet
```
calcloglikelihood

Calculate the log likelihood function of the data.

Description
This function calculates the log likelihood function of the data.

Usage
calcloglikelihood(pis, y)

Arguments
pis Vector of link probabilities.
y Vector version of the adjacency matrix, with the diagonal removed.

Value
The value of the log likelihood function.

References

See Also
MEclustnet

calcm

Totals the number of observations in each cluster.

Description
Update the count of the number of observations in each cluster.

Usage
calcm(m, G, K)

Arguments
m Vector of length G containing the number of nodes in each cluster.
G The number of clusters in the model being fitted.
K Vector of length n detailing the number of the cluster to which each node belongs.
**calcpis**

*Calculate link probabilities.*

**Description**

Function calculates link probabilities between nodes.

**Usage**

```r
calcpis(beta, x.link, delta, n.tilde)
```

**Arguments**

- `beta`: Vector of regression coefficients in the link probabilities.
- `x.link`: Matrix, with $n^2 - n$ rows and the same number of columns as covariates (including the intercept), giving the differences in covariates for all pairs of nodes.
- `delta`: Vector of Euclidean distances between locations in the latent space of all pairs of nodes.
- `n.tilde`: Length of the vector version of the adjacency matrix, with the diagonal removed i.e. $n^2 - n$.

**Value**

A vector of length $n^2 - n$ providing the link probabilities between all pairs of nodes.

**References**


**See Also**

*MEclustnet*
formatting.covars Reformat matrix of covariates.

Description

This function reformats the matrix of input covariates into the required format for the link probabilities and for the mixing proportions.

Usage

formatting.covars(covars, link.vars, mix.vars, n)

Arguments

covars The n x p data frame of node specific covariates passed in to the overall MEclustnet function. The first column should be a column of 1’s and categorical variables should be factors.
link.vars A vector detailing the column numbers of the matrix covars that should be included in the link probabilities model.
mix.vars A vector detailing the column numbers of the matrix covars that should be included in the mixing proportions probabilities model.
n The number of nodes in the network.

Details

For the link regression model, the difference in the link.vars covariates, for all pairs of nodes is calculated. For the mixing proportions model, the required representation of the mix.vars required is formed, where for categorical/factor variables a dummy value representation is used.

Value

A list with

x.link A matrix with \( n^2 \) rows and length(link.vars) columns, detailing the differences in covariates for all pairs of nodes.

x.mix A matrix with n rows and number of columns equal to the number of variables detailed in mix.vars, where dummy variable representations will be used for categorical.factor covariates.

References


See Also

MEclustnet
Examples

```r
data(us.twitter.covariates)
link.vars = c(1)
mix.vars = c(1,5,7,8)
res = formatting.covars(us.twitter.covariates, link.vars, mix.vars, nrow(us.twitter.covariates))
dim(res$x.link)
dim(res$x.mix)
```

---

**invariant**

Account for invariance of configurations.

Description

This function accounts for the fact that configurations in the latent space are invariant to rotations, reflections and translations.

Usage

```r
invariant(z, zMAP)
```

Arguments

- `z` An n x d matrix of latent locations in the d dimensional space for each of n nodes.
- `zMAP` The maximum a posteriori configuration of latent locations used as the template to which all sampled configurations are mapped.

Details

Procrustean rotations, reflections and translations (note: NOT dilations) are employed to best match z to zMAP.

Value

The transformed version of the input configuration z that best matches zMAP.

References


See Also

`MEclustnet`
**labelswitch**

*Label switching correction.*

**Description**

This function corrects for the issue of label switching when fitting mixture models in a Bayesian setting.

**Usage**

```r
labelswitch(mu, sigma2, lambda, tau, K, G, d, perms, muMAP, iter, uphill, burnin, thin, s, x.mix)
```

**Arguments**

- `mu` A G x d matrix of mean latent locations.
- `sigma2` A vector of length G containing the covariance of the latent locations within each cluster.
- `lambda` An n x G matrix of mixing proportions.
- `tau` A matrix of logistic regression coefficients, with G rows and number of columns equal to the number of covariates in the mixing proportions model plus 1, for the intercept.
- `K` Vector of length n detailing the number of the cluster to which each node belongs.
- `G` The number of clusters in the model being fitted.
- `d` The dimension of the latent space.
- `perms` A G! x G matrix of all possible permutations of 1:G (output by permutations(G), say).
- `muMAP` A G x d matrix of maximum a posteriori latent location means, obtained at the end of the uphill only section of the MCMC chain. Used as the template to correct for label switching.
- `iter` Iteration number.
- `uphill` Number of iterations for which uphill only steps in the MCMC chain should be run.
- `burnin` Number of iterations of the MCMC chain which should not be included in a posteriori summaries.
- `thin` Thinning frequency of the MCMC chain to ensure independent samples.
- `s` Number of columns in the reformatted covariates matrix for the mixing proportions model, output by `formatting.covars`.
- `x.mix` The reformatted covariates matrix for the mixing proportions model, output by `formatting.covars`. 
Details

The muMAP matrix is used as the reference to which each new estimate the cluster means is matched to correct for any label switching which may have occurred during sampling. A sum of squares function is employed as the loss function.

Value

A list containing: list(mu, sigma2, lambda, tau, K)

mu The label-corrected matrix of cluster means.

sigma2 The label-corrected vector of cluster covariances.

lambda The label-corrected matrix of mixing proportions.

tau The label-corrected matrix of logistic regression coefficients for the mixing proportions model.

K The label-corrected vector of length n detailing the number of the cluster to which each node belongs.

References


See Also

MEclustnet

lawyers.adjacency.advice

Adjacency matrix detailing the presence or absence of advice links between the ‘Lazega Lawyers’.

Description

Data on whether or not 71 lawyers in a northeastern American law firm asked each other for advice.

Usage

lawyers.adjacency.advice

Format

A 71 x 71 binary matrix, with 0 down the diagonal.

Source

lawyers.adjacency.coworkers

Adjacency matrix detailing the presence or absence of coworker links between the 'Lazega Lawyers'.

Description
Data on whether or not 71 lawyers in a northeastern American law firm work with each other.

Usage
lawyers.adjacency.coworkers

Format
A 71 x 71 binary matrix, with 0 down the diagonal.

Source

lawyers.adjacency.friends

Adjacency matrix detailing the presence or absence of friendship links between the 'Lazega Lawyers'.

Description
Data on whether or not 71 lawyers in a northeastern American law firm are friends outside of work.

Usage
lawyers.adjacency.friends

Format
A 71 x 71 binary matrix, with 0 down the diagonal.

Source
lawyers.covariates  

**A matrix of covariates of the ‘Lazega Lawyers’**.

**Description**

Covariates on each of 71 lawyers in a northeastern American law firm. Note the first column is a column of 1’s.

**Usage**

lawyers.covariates

**Format**

A data frame with 71 observations on the following 8 variables.

- **Intercept** a column of 1s should always be the first column.
- **Seniority** a factor with levels 1 = partner, 2 = associate.
- **Gender** a factor with 1 = male, 2 = female.
- **Office** a factor with levels 1 = Boston, 2 = Hartford and 3 = Providence
- **Years** a numeric vector detailing years with the firm.
- **Age** a numeric vector detailing the age of each lawyer.
- **Practice** a factor with levels 1 = litigation and 2 = corporate.
- **School** a factor with levels 1 = Harvard or Yale, 2 = University of Connecticut and 3 = Other.

**Source**


---

**MEclustnet**

**MEclustnet: A package for model-based clustering of nodes in a network, accounting for covariates.**

**Description**

The main function of interest is MEclustnet which will fit a mixture of experts latent position cluster model to a binary network.

MEclustnet will fit a mixture of experts latent position cluster model to a binary network.
Usage

MEclustnet(Y, covars, link.vars = c(1:ncol(covars)),
mix.vars = c(1:ncol(covars)), G = 2, d = 2, itermax = 10000,
uphill = 100, burnin = 1000, thin = 10, rho.input = 1,
verbose = TRUE, ...)

Arguments

Y An n x n binary matrix of links between n nodes, with 0 on the diagonal and 1 indicating a link.
covars An n x p data frame of node specific covariates. Categorical variables should be factors. First column should be a column of 1s, and should always be passed in.
link.vars A vector of the column numbers of the data frame covars to be included in link probability model. If none are to be included, this argument should be 1.
mix.vars A vector of the column numbers of the data frame covars to be included in mixing proportions model. If none are to be included, argument should be 1.
G The number of clusters in the model to be fitted.
d The dimension of the latent space.
itermax Maximum number of iterations in the MCMC chain.
uphill Number of iterations for which uphill only steps in the MCMC chain should be run to find maximum a posteriori estimates.
burnin Number of burnin iterations in the MCMC chain.
thin The degree of thinning to be applied to the MCMC chain.
rho.input Scaling factor to achieve desirable acceptance rates in Metropolis-Hastings steps.
verbose Print progress updates to screen? Recommended as the models are slow to run.
... Additional arguments.

Details

This function fits the mixture of experts latent position cluster model to a binary network via a Metropolis-within-Gibbs sampler. Covariates can influence either the link probabilities between nodes and/or the cluster memberships of nodes.

Value

An object of class MEclustnet, which is a list containing:

zstore An n x d x store.dim array of sampled latent location matrices, where store.dim is the number of post burnin thinned iterations.
betastore A store.dim x p matrix of sampled beta vectors, the logistic regression parameters of the link probabilities model.
Kstore A store.dim x n matrix of sampled cluster membership vectors.
mustore A G x d x store.dim array of sampled cluster mean latent location matrices.
sigma2store A store.dim x G matrix of sampled cluster variances.
**MEclustnet**

**lambdastore** An n x G x store.dim array of sampled mixing proportion matrices.

**taustore** A G x s x store.dim array of sampled tau vectors, the logistic regression parameters of the mixing proportions model, where s is the length of tau.

**LLstore** A vector of length store.dim storing the loglikelihood from each stored iteration.

**G** The number of clusters fitted

**d** The dimension of the latent space

**countbeta** Count of accepted beta values

**counttau** Count of accepted tau values

**MEclustnet functions**

MEclustnet

**References**


**See Also**

MEclustnet

**Examples**

```r
#################################################################
# An example from the Gormley and Murphy (2010) paper, using the Lazega lawyers friendship network.
# Number of iterations etc. are set to low values for illustrative purposes.
# Longer run times are likely to be required to achieve sufficient mixing.
library(latentnet)
data(lawyers.adjacency.friends)
data(lawyers.covariates)

link.vars = c(1)
mix.vars = c(1,4,5)

fit = MEclustnet(lawyers.adjacency.friends, lawyers.covariates,
                 link.vars, mix.vars, G=2, d=2, itermax = 500, burnin = 50, uphill = 1, thin=10)

# Plot the trace plot of the mean of dimension 1 for each cluster.
matplot(t(fit$mustore[,1,]), type="l", xlab="Iteration", ylab="Parameter")

# Compute posterior summaries
summ = summaryMEclustnet(fit, lawyers.adjacency.friends)
plot(summ$zmean, col=summ$Kmode, xlab="Dimension 1", ylab="Dimension 2", pch=summ$Kmode, main = "Posterior mean latent location for each node.")

# Plot the resulting latent space, with uncertainties
plotMEclustnet(fit, lawyers.adjacency.friends, link.vars, mix.vars)
```
# An example analysing a 2016 Twitter network of US politicians.

# Number of iterations etc. are set to low values for illustrative purposes.
# Longer run times are likely to be required to achieve sufficient mixing.

library(latentnet)
data(us.twitter.adjacency)
data(us.twitter.covariates)

link.vars = c(1)
mix.vars = c(1,5,7,8)

fit = MEclustnet(us.twitter.adjacency, us.twitter.covariates, link.vars, mix.vars, G=4, d=2, itermax = 500, burnin = 50, uphill = 1, thin=10)

# Plot the trace plot of the mean of dimension 1 for each cluster.
matplot(t(fit$mustore[,1,]), type="l", xlab="Iteration", ylab="Parameter")

# Compute posterior summaries
summ = summaryMEclustnet(fit, us.twitter.adjacency)

plot(summ$zmean, col=summ$Kmode, xlab="Dimension 1", ylab="Dimension 2", pch=summ$Kmode, main = "Posterior mean latent location for each node.")

# Plot the resulting latent space, with uncertainties
plotMEclustnet(fit, us.twitter.adjacency, link.vars, mix.vars)

# Examine which politicians are in which clusters...
clusters = list()
for(g in 1:fit$G)
{
  clusters[[g]] = us.twitter.covariates[summ$Kmode==g,c("name", "party")]
}
clusters

---

**plotMEclustnet**  
*Plot latent position network.*

**Description**

Function to plot the resulting fitted network, using first two dimensions only.

**Usage**

plotMEclustnet(fit, Y, link.vars, mix.vars)
Arguments

- **fit**: An object storing the output of the function `MEclustnet`.
- **Y**: The n x n binary adjacency matrix, with 0 down the diagonal, that was passed to `MEclustnet`.
- **link.vars**: A vector of the column numbers of the data frame `covars` to be included in link probability model. If none are to be included, this argument should be 1.
- **mix.vars**: A vector of the column numbers of the data frame `covars` to be included in mixing proportions model. If none are to be included, argument should be 1.

Details

This function will plot the posterior mean latent location for each node in the network. The colour of each node reflects the posterior modal cluster membership, and the ellipses are 50% posterior sets illustrating the uncertainty in the latent locations. The grey lines illustrate the observed links between the nodes.

References


See Also

- `MEclustnet`

Examples

```r
library(latentnet)
data(us.twitter.adjacency)
data(us.twitter.covariates)

link.vars = c(1)
mix.vars = c(1,5,7,8)

fit = MEclustnet(us.twitter.adjacency, us.twitter.covariates, 
                  link.vars, mix.vars, G=4, d=2, itermax = 500, burnin = 50, uphill = 1, thin=10)

# Plot the trace plot of the mean of dimension 1 for each cluster.
matplot(t(fit$mustore[,1,]), type="l", xlab="Iteration", ylab="Parameter")

# Compute posterior summaries
summ = summaryMEclustnet(fit, us.twitter.adjacency)

plot(summ$zmean, col=summ$Kmode, xlab="Dimension 1", ylab="Dimension 2", pch=summ$Kmode,
```
summaryMEclustnet

    main = "Posterior mean latent location for each node."

# Plot the resulting latent space, with uncertainties
plotMEclustnet(fit, us.twitter.adjacency, link.vars, mix.vars)

# Examine which politicians are in which clusters...
clusters = list()
for(g in 1:fit$G)
{
    clusters[[g]] = us.twitter.covariates[summ$Kmode==g,c("name", "party")]
}
clusters

summaryMEclustnet  

Summary of MEclustnet object.

Description

Summary of the output of the function MEclustnet which fits a mixture of experts latent position
cluster model.

Usage

summaryMEclustnet(fit, Y)

Arguments

fit       An object storing the output of the function MEclustnet.
Y         The n x n binary adjacency matrix, with 0 down the diagonal, that was passed to
           MEclustnet.

Value

A list with:

AICM  The value of the AICM criterion for the fitted model.
BICM  The value of the BICM criterion for the fitted model.
BICMCMC  The value of the BICMCMC criterion for the fitted model.
betamean       The posterior mean vector of the regression coefficients for the link probabilities model.
betasd         The standard deviation of the posterior distribution of beta.
taumean        A matrix with G rows, detailing the posterior mean of the regression coefficients for the
                mixing proportions model.
tausd          The standard deviation of the posterior distribution of tau.
mumean         A G x d matrix containing the posterior mean of the latent locations’ mean.
**summaryMEclustnet**

*meansd* The standard deviation of the posterior distribution of mu.

*sigma2mean* A vector of length G containing the posterior mean of the latent locations’ covariance.

*sigma2sd* The standard deviation of the posterior distribution of the latent locations’ covariance.

*Kmode* A vector of length n detailing the posterior modal cluster membership for each node.

*zmean* An n x d matrix containing the posterior mean latent location for each node.

**References**


**See Also**

MEclustnet

**Examples**

```r
library(latentnet)
data(us.twitter.adjacency)
data(us.twitter.covariates)

link.vars = c(1)
mix.vars = c(1,5,7,8)

fit = MEclustnet(us.twitter.adjacency, us.twitter.covariates,
                 link.vars, mix.vars, G=4, d=2, itermax = 500, burnin = 50, uphill = 1, thin=10)

# Plot the trace plot of the mean of dimension 1 for each cluster.
matplot(t(fit$mustore[,1,]), type="l", xlab="Iteration", ylab="Parameter")

# Compute posterior summaries
summ = summaryMEclustnet(fit, us.twitter.adjacency)

plot(summ$zmean, col=summ$Kmode, xlab="Dimension 1", ylab="Dimension 2", pch=summ$Kmode,
     main = "Posterior mean latent location for each node."
)

# Plot the resulting latent space, with uncertainties
plotMEclustnet(fit, us.twitter.adjacency, link.vars, mix.vars)

# Examine which politicians are in which clusters...
clusters = list()
for(g in 1:fit$G)
{
  clusters[[g]] = us.twitter.covariates[summ$Kmode==g,c("name", "party")]
}
```
updatebeta

**clusters**

<table>
<thead>
<tr>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>updatebeta</td>
<td>Update the logistic regression parameters in the link probabilities model.</td>
</tr>
</tbody>
</table>

**Description**

The Metropolis-Hastings update step for the logistic regression parameters in the link probabilities model, using a surrogate proposal distribution.

**Usage**

```r
updatebeta(beta, p, x.link, delta, y, epsilon, psi, psi.inv, pis, countbeta, rho, n.tilde)
```

**Arguments**

- `beta` Vector of regression coefficients in the link probabilities.
- `p` Length of beta.
- `x.link` Matrix, with $n^2 - n$ rows and the same number of columns as covariates (including the intercept), giving the differences in covariates for all pairs of nodes.
- `delta` Vector of Euclidean distances between locations in the latent space of all pairs of nodes.
- `y` Vector version of the adjacency matrix, with the diagonal removed.
- `epsilon` Mean of the multivariate normal prior on beta.
- `psi` Covariance of the multivariate normal prior on beta.
- `psi.inv` Inverse covariance of the multivariate normal prior on beta.
- `pis` Vector of length $n^2 - n$ providing the link probabilities between all pairs of nodes.
- `countbeta` Counter for number of steps for which the proposed beta value was accepted.
- `rho` Scaling factor to be used to adjust the acceptance rate.
- `n.tilde` Length of the vector version of the adjacency matrix, with the diagonal removed i.e. $n^2 - n$.

**Details**

See appendix of the paper detailed below for details.

**Value**

A list:

- `beta` The returned version of the beta parameter vector.
- `countbeta` The count of the number of acceptances of beta to that point in the MCMC chain.
References

See Also
MEclustnet

updateK

Update the cluster membership vector.

Description
A Gibbs update step for K, the cluster membership vector.

Usage
updateK(G, K, z, mu, sigma2, Id, lambda)

Arguments
G The number of clusters being fitted.
K The cluster membership vector.
z The n x d matrix of latent locations.
mu The G x d matrix of cluster means.
sigma2 The G vector of cluster covariances.
Id An identity matrix of dimension d.
lambda The n x G matrix of mixing proportions.

Value
The cluster membership vector.

References

See Also
MEclustnet
updatemu

Update the mean of each cluster.

Description

A Gibbs step to update the mean of each cluster.

Usage

updatemu(G, z, K, m, sigma2, omega2, Id, mu, d)

Arguments

G  The number of clusters being fitted.
z  The n x d matrix of latent locations.
K  The cluster membership vector.
m  Vector of length G containing the number of nodes in each cluster.
sigma2  The covariance of each cluster.
omega2  Covariance of the multivariate normal prior distribution on the means. Note this is a scalar value, as the prior covariance is diagonal.
Id  A d x d identity matrix.
mu  The G x d matrix of cluster means.
d  The dimension of the latent space.

Value

The G x d matrix of cluster means.

References


See Also

MEclustnet
updatesigma2  

*Update variances in each cluster.*

**Description**

A Gibbs step to update variances in each cluster.

**Usage**

`updatesigma2(G, alpha, m, d, sigma02, z, K, mu, sigma2)`

**Arguments**

- **G**  
  The number of clusters being fitted.
- **alpha**  
  Degrees of freedom of the scaled inverse Chi squared prior distribution on the cluster variances.
- **m**  
  Vector of length G containing the number of nodes in each cluster.
- **d**  
  Dimension of the latent space.
- **sigma02**  
  Scaled factor of the scaled inverse Chi squared prior distribution on the cluster variances.
- **z**  
  The n x d matrix of latent locations.
- **K**  
  The cluster membership vector.
- **mu**  
  The G x d matrix of cluster means.
- **sigma2**  
  The G vector of cluster variances.

**Value**

The G vector of cluster variances.

**References**


**See Also**

`MEclustnet`
**Description**

The Metropolis-Hastings update step for the logistic regression parameters in the mixing proportions model, using a surrogate proposal distribution.

**Usage**

```r
updatetau(G, x.mix, lambda, Sigmag, Sigmag.inv, K, gammag, tau, counttau, rho)
```

**Arguments**

- `G` The number of clusters being fitted.
- `x.mix` A matrix of covariates in the mixing proportions model (including dummy variables for any factor covariates), with a column of 1’s appended at the front.
- `lambda` An n x G matrix of mixing proportions.
- `Sigmag` Covariance matrix of the multivariate normal prior for tau.
- `Sigmag.inv` The inverse of Sigmag.
- `K` The cluster membership vector.
- `gammag` Mean vector of the multivariate normal prior for tau.
- `tau` A matrix of logistic regression coefficients, with G rows and number of columns equal to the number of covariates in the mixing proportions model plus 1, for the intercept.
- `counttau` Counter for number of steps for which the proposed tau value was accepted.
- `rho` Scaling factor to be used to adjust the acceptance rate.

**Value**

A list:

- `tau` The returned version of the tau parameter vector.
- `lambda` The returned version of the lambda matrix.
- `counttau` The count of the number of acceptances of tau to that point in the MCMC chain.

**References**


**See Also**

`MEclustnet`
**Description**

A Metropolis-Hastings update step for the latent locations.

**Usage**

```r
updatez(n, z, x.link, delta, beta, y, mu, K, sigma2, Id, pis, iter, uphill,
        countz, delete, d, n.tilde)
```

**Arguments**

- `n`: The number of nodes.
- `z`: The n x d matrix of latent locations.
- `x.link`: Matrix, with \( n^2 - n \) rows and the same number of columns as covariates (including the intercept), giving the differences in covariates for all pairs of nodes.
- `delta`: Vector of Euclidean distances between locations in the latent space of all pairs of nodes.
- `beta`: Vector of regression coefficients in the link probabilities.
- `y`: Vector version of the adjacency matrix, with the diagonal removed.
- `mu`: The G x d matrix of cluster means.
- `K`: The cluster membership vector.
- `sigma2`: The covariance of each cluster.
- `Id`: A d dimensional identity matrix.
- `pis`: A vector of length \( n^2 - n \) providing the link probabilities between all pairs of nodes.
- `iter`: Iteration number.
- `uphill`: Number of iterations for which uphill only steps in the MCMC chain should be run.
- `countz`: Counter for number of steps for which the proposed z value was accepted.
- `delete`: Index of the terms to be deleted in order to delete the diagonal terms from the vector version of the adjacency matrix.
- `d`: The dimension of the latent space.
- `n.tilde`: Length of the vector version of the adjacency matrix, with the diagonal removed i.e. \( n^2 - n \).
Value

A list:

- **z**: The returned matrix of latent locations.
- **delta**: Vector of Euclidean distances between locations in the latent space of all pairs of nodes.
- **pis**: A vector of length \( n^2 - n \) providing the link probabilities between all pairs of nodes.
- **countz**: Counter for \( z \) acceptance rate.

References


See Also

- [MEclustnet](#)

---

**Description**

Network data on whether or not 69 US politicians are friends/followers on Twitter.

**Usage**

```r
us.twitter.adjacency
```

**Format**

A 69 x 69 binary matrix, with 0 down the diagonal.

**Source**

With thanks to Dr. Derek Greene. School of Computer Science, University College Dublin.
us.twitter.covariates  

A matrix of covariates of the US politicians.

Description

Covariates on each of 69 US politicians. Note the first column is a column of 1’s.

Usage

us.twitter.covariates

Format

A data frame with 69 observations on the following 8 variables.

‘1’ a column of 1s should always be the first column.
twitter_id  Twitter number.
twitter_name  Twitter name.
nname  Actual name.
party  a factor with levels Democrat Republican
location  a factor with levels detailing location.
role  a factor with levels Candidate, Representative and Senator
gender  a factor with levels Female and Male

Source

Index

* datasets
  lawyers.adjacency.advice, 8
  lawyers.adjacency.coworkers, 9
  lawyers.adjacency.friends, 9
  lawyers.covariates, 10
  us.twitter.adjacency, 23
  us.twitter.covariates, 24

  calclambda, 2
  calcloglikelihood, 3
  calcm, 3
  calcpis, 4

  formatting.covars, 5, 7

  invariant, 6

  labelswitch, 7
  lawyers.adjacency.advice, 8
  lawyers.adjacency.coworkers, 9
  lawyers.adjacency.friends, 9
  lawyers.covariates, 10

  MEclustnet, 2–6, 8, 10, 12, 14–16, 18–21, 23
  MEclustnet-package (MEclustnet), 10

  plotMEclustnet, 13

  summaryMEclustnet, 15

  updatebeta, 17
  updateK, 18
  updatemu, 19
  updatesigma2, 20
  updatetau, 21
  updatez, 22
  us.twitter.adjacency, 23
  us.twitter.covariates, 24