Package 'collpcm'

April 23, 2024

Version 1.4

Type Package

Title Collapsed Latent Position Cluster Model for Social Networks

Date 2024-04-23

Description Markov chain Monte Carlo based inference routines for collapsed latent position cluster models or social networks, which includes searches over the model space (number of clusters in the latent position cluster model). The label switching algorithm used is that of Nobile and Fearnside (2007) <doi:10.1007/s11222-006-9014-7> which relies on the algorithm of Carpaneto and Toth (1980) <doi:10.1145/355873.355883>.

Depends R (>= 2.10), network, latentnet, gtools

License GPL-3

NeedsCompilation yes

Maintainer Jason Wyse <wyseja@tcd.ie>

Author Jason Wyse [aut, cre],
Caitriona Ryan [aut],
Nial Friel [ths]

Repository CRAN

Date/Publication 2024-04-23 21:00:02 UTC

R topics documented:

`collpcm.control` *Specify parameters determining the collapsed LPCM model and MCMC fitting run*

Description

Specify the number of samples to be collected, burn in to be used, sub-sampling interval, whether variable model jumps are allowed, and whether to run a pilot sample in the initial model.

Usage

```
collpcm.control( x = list() , n, d )
```

Arguments

`x`	An optional list setting the set up parameters of the model. Any parameters not set in the list will default to the values described below.
`n`	The number of nodes in the network.
`d`	The dimension of the latent space for model fitting.

Value

`collpcm.control` returns a list giving the set up of the problem containing the following items:

`G`	Initial value of `G` for the chain.
`Gmax`	The maximum allowed value of `G` if doing model search.
`Gprior`	Log of the prior mass on the number of components `G`.
`xi`	Mean of the prior on the model intercept.
`psi`	Standard deviation of the prior on the model intercept.
`gamma`	Twice the rate of the Gamma prior on the cluster precision.
`delta`	Twice the shape of the Gamma prior on the cluster precision.
`alpha`	The parameter of the Dirichlet prior on group weights.
`kappa`	The scaling of the prior mean for the cluster centre (in units of cluster precision).
`betainit`	Initial value given to the intercept for the MCMC run.
`Xinit`	Initial configuration of latent positions for the MCMC run.
`sample`	Number of MCMC samples to be stored.
`burn`	Number of MCMC iterations to discard as burn-in.
`interval`	Number of iterations at which to sub-sample the chain and store i.e. total iterations post burn-in is sample*interval.
`model.search`	Logical; if TRUE (default) the model space for G is searched.
`pilot`	Number of iterations to run as a pilot to adapt the proposal standard deviations for the MCMC chains (in addition to adaptation during burn-in).

sd.beta.prop	Standard deviation of the random walk proposal updating the intercept.
sd.X.prop	Standard deviation of the (possibly multivariate) random walk proposal for an actor's latent position.
gamma.update	Logical; if TRUE (default) then the gamma hyperparameter is updated as part of the MCMC run.
store.sparse	Logical; do a sparse form of storage and don't return or store some of the MCMC run and only keep summary values.
adapt	Logical; if TRUE (default) use an adaptive phase during burn-in to tune the standard deviation of the proposals to get an "optimal" acceptance rate.
adapt.interval	The number of iterations between tweaks of the proposal standard deviations in the adaptation phase.
MKL	Logical; if TRUE (default) compute the maximum Kullback-Liebler configuration of the latent positons from Handcock, Raftery & Tantrum (2007)
verbose	Logical; if TRUE (default) print out progression messages througout the MCMC run and stages of fitting.

Author(s)

Jason Wyse

References

Ryan, C., Wyse, J. and Friel, N. (2017) *Bayesian model selection for the latent position cluster model for Social Networks.* Network Science, volume 5, 70-91.

---

collpcm.fit *Fit a latent position cluster network model with model search*

Description

collpcm.fit is used to fit the latent position cluster model with uncertainty in the number of clusters incorporated. A posterior distribution for the number of clusters is estimated.

Usage

```
collpcm.fit( Y , d = 2, G = NULL, Gmax = NULL, control = list(), Xref = NA )
```

Arguments

Y	A network object containing the network in question.
d	The dimension of the latent position to represent each node in the network (defaults to 2).
G	Give the initial number of groups for the algorithm.
Gmax	Give the maximum allowed number of groups if doing model search.
control	List giving the set up of the algorithm (see collpcm.control)
Xref	Optional latent positions to be used as a reference configuration for the Procrustes rotations.

Value

collpcm.fit returns an object of class collpcm that is a list. The list will have the following slots.

call	The values of each of the arguments used in the model fitting MCMC run.
sample	A list containing the samples from the MCMC run.
Gpost	Estimated posterior distribution of the number of groups/clusters.
Xpostmean	Estimated posterior mean from sampled latent positions.
XpostMKL	MKL posterior latent positions as described in Handcock, Raftery & Tantrum (2007).
Gslot	An indexing vector for the lists of posterior mean and MKL positions.
acceptance.rates	Acceptance rates for different moves of MCMC algorithm.
adapted.sd.prop	The standard deviations of the proposal distributions after the adaptation phase.
timings	A list of timings for each part of the algorithm.

Author(s)

Jason Wyse

References

Ryan, C., Wyse, J. and Friel, N. (2017). *Bayesian model selection for the latent position cluster model for Social Networks.* Network Science, volume 5, 70-91.

Handcock, M. S., Raftery, A. E. and Tantrum, J. (2007). *Model-Based Clustering for Social Networks.* Journal of the Royal Statistical Society, Series A, Vol. 170, 301-354 <doi: 10.1111/j.1467-985X.2007.00471.x>

See Also

collpcm.control

Examples

```
# load the Monks data
data(Monks)

# run the model printing run updates to screen
# this is an illustrative example (it should be run for much longer)
z <- collpcm.fit( Monks, G=3, d=2,
control=list( verbose=TRUE, sample=2500, interval=1, burn=500 ) )

# plot of the collpcm object
plot( z )
```

collpcm.summaryplot *Make a summary plot of a* collpcm *run*

Description

collpcm.summaryplot creates a 2 by 2 summary plot showing traces from the MCMC run as well as the posterior KL positions for the most visited model.

Usage

```
collpcm.summaryplot( x )
```

Arguments

x	An object of class collpcm

Author(s)

Jason Wyse

References

Ryan, C., Wyse, J. and Friel N. (2017) *Bayesian model selection for the latent position cluster model for Social Networks.* Network Science, volume 5, 70-91.

collpcm.undo.label.switching *Correct samples of label vectors for label switching.*

Description

collpcm.undo.label.switching is used to correct sampled label vectors for label switching using the method proposed by Nobile and Fearnside (2007) which relies on the assignment algorithm of Carpaneto and Toth (1980).

Usage

```
collpcm.undo.label.switching( Z, Gsamp = NULL )
```

Arguments

Z	A matrix of dimensions (num samples) by n giving the sampled label vectors for each iteration of the MCMC run.
Gsamp	A vector of length (num samples) giving the corresponding number of groups for each iteration of the MCMC run.

Value

collpcm.undo.label.switching returns a list with the following slots.

call	The function call.
relab	The post processed Z matrix after label switching has been corrected for
label.probs	List giving the probability of belonging to each group for each item, each entry of the list corresponding to a given number of components.
permutation	The permutation applied to each row to correct for label switching.

Author(s)

Jason Wyse

References

Nobile, A. and Fearnside A. T. (2007). *Bayesian finite mixtures with an unknown number of components: The allocation sampler* Statistics and Computing, Vol. 17, 147-162 <doi:10.1007/s11222-006-9014-7>

Carpaneto, G. and Toth, P. (1980). *Algorithm 548: Solution of the Assignment Problem [H]* ACM Transactions on Mathematical Software, Vol. 6, 104-111 <doi:10.1145/355873.355883>

Dolphins *Dolphins*

Description

Network describing social ties between dolphins off doubtful sound.

Usage

```
data(Dolphins)
```

Source

Lusseau, D., Schneider, K., Boisseau, O. J., Hasse, P., Slooten, E. and Dawson, S. M. (2003) *The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations- Can geographic isolation explain this unique trait?* Behavioural Ecology and Sociobiology 54, 396–405.

Karate *Network describing loyalty in the Karate club.*

Description

The well known Karate data.

Usage

```
data(Karate)
```

Source

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

Monks *Monks*

Description

Sampson's aggregated Monk's dataset.

Usage

```
data(Monks)
```

Source

Sampson, S.~F. (1968), *A novitiate in a period of change: An experimental and case study of relationships,* Unpublished Ph.D. dissertation, Department of Sociology, Cornell University.

http://vlado.fmf.uni-lj.si/pub/networks/data/esna/sampson.htm

`plot.collpcm` *Plotting a collpcm object*

Description

Plot the posterior mean latent positions for G groups.

Usage

```
## S3 method for class 'collpcm'
plot( x, ..., G = NULL, label.nodes = NULL, pie = TRUE,

vertex.col = c( "red", "green", "blue", "cyan", "magenta", "orange", "yellow", "purple"),

vertex.cex = 1, object.scale = formals(plot.network.default)[["object.scale"]] )
```

Arguments

`x`	An object of class `collpcm`.
`...`	Additional arguments including.
`G`	The number of groups in the model to be plotted. Defaults to most visited in MCMC run.
`label.nodes`	A vector of labels to print beside corresponding nodes on the plot.
`pie`	Logical; Draw small pie charts to indicate group membership probabilities.
`vertex.col`	The colour for the slices of pie (previous).
`vertex.cex`	Magnify the vertex
`object.scale`	Scale up/down the size of the plotting of vertex and arrows.

Details

This function gives a plot of the latent positions for a given number of groups (assuming the model with the specified number of groups has been visited during the run of the sampler). If argument `pie` is set to `TRUE`, membership probabilities of the nodes are indicated by pie charts with each colour corresponding to a different group in the model. Some of the code to implement this function draws heavily on code contained in the `latentnet` package (Krivitsky & Handcock, 2015).

Author(s)

Jason Wyse

References

Ryan, C., Wyse, J. and Friel, N. (2017). *Bayesian model selection for the latent position cluster model for Social Networks.* Network Science, volume 5, 70-91.

Krivitsky P and Handcock M (2015). latentnet: Latent Position and Cluster Models for Statistical Networks. The Statnet Project (http://www.statnet.org). R package version 2.7.1, http://CRAN.R-project.org/package=latentnet.

`print.collpcm` *Print a collpcm object*

Description

Print a summary of a collpcm object.

Usage

```
## S3 method for class 'collpcm'
print( x, ... )
```

Arguments

`x`	An object of class `collpcm`.
`...`	Optional arguments to lower level functions.

Author(s)

Jason Wyse

References

Ryan, C., Wyse, J. and Friel, N. (2017) *Bayesian model selection for the latent position cluster model for Social Networks.* Network Science, volume 5, 70-91.

`summary.collpcm` *Summarize a collpcm object*

Description

Print a summary of a collpcm object.

Usage

```
## S3 method for class 'collpcm'
summary( object, ... )
```

Arguments

`object`	An object of class `collpcm`.
...	Optional arguments to lower level functions.

Author(s)

Jason Wyse

References

Ryan, C., Wyse, J. and Friel, N. (2017) *Bayesian model selection for the latent position cluster model for Social Networks.* Network Science, volume 5, 70-91.

Index