Package ‘NetworkChange’

August 19, 2017

Title Bayesian Package for Network Changepoint Analysis

Version 0.3

Date 2017-08-18

Author Jong Hee Park [aut,cre], Yunkyu Sohn [aut]

Maintainer Jong Hee Park <jongheepark@snu.ac.kr>

Depends R (>= 2.10.0),

Imports ggplot2, Rmpfr, abind, mvtnorm, grid, LaplacesDemon, MCMCpack, stats, MASS, methods, RColorBrewer, ggvis

Description Network changepoint analysis for undirected network data. The package implements a hidden Markov multilinear tensor regression model (Park and Sohn, 2017, <http://jhp.snu.ac.kr/NetworkChange.pdf>). Functions for break number detection using the approximate marginal likelihood and WAIC are also provided.

License GPL-3

NeedsCompilation no

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2017-08-19 15:40:28 UTC

R topics documented:

`MakeBlockNetworkChange`

Build a synthetic block-structured temporal data with breaks

Description

MakeBlockNetworkChange generates a block-structured temporal data with breaks

Usage

```
MakeBlockNetworkChange(n = 10, break.point = 0.5, block.number = 3,
  base.prob = 0.05, block.prob = 0.5, shape = 10, T = 40,
  type = "merge")
```

Arguments

`n`	The number of nodes within a block. The total number of nodes is n*block.number.
`break.point`	The point of break. 0 indicates the beginning, 0.5 indicates the middle, and 1 indicates the end.
`block.number`	The number of blocks.
`base.prob`	The probability of link among non-block members.
`block.prob`	The probability of link among within-block members.
`shape`	The speed of breaks. The larger shape is, the faster the transition is. 0 is the minimum.
`T`	The length of time.
`type`	The type of network changes. Options are "constant", "merge", "split", "merge-split", "split-merge." If "constant" is chosen, the number of breaks is zero. If "merge" or "split" is chosen, the number of breaks is one. If either "merge-split" or "split-merge" is chosen, the number of breaks is two.

Value

output An output of `MakeBlockNetworkChange` contains a symmetric block-structured temporal network data set with breaks.

MarginalCompare *Compare Log Marginal Likelihood*

Description

Compare Log Marginal Likelihood

Usage

```
MarginalCompare(outlist)
```

Arguments

`outlist`	List of NetworkChange objects

Value

A matrix of log marginal likelihoods.

References

Siddhartha Chib. 1995. "Marginal Likelihood from the Gibbs Output." *Journal of the American Statistical Association*. 90: 1313-1321.

See Also

`WaicCompare`

multiplot *Printing multiple ggplots in oone file*

Description

Print multiple ggplots in one file. Slightly modified for packaging from the original version in the web.

Usage

```
multiplot(..., plotlist = NULL, cols = 1, layout = NULL)
```

Arguments

`...`	A list of ggplot objects separated by commas.
`plotlist`	A list of ggplot objects
`cols`	The number of columns.
`layout`	A matrix specifying the layout. If present, 'cols' is ignored.

Value

A plot object

Author(s)

http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

`NetworkChange` *Changepoint analysis of a degree-corrected multilinear tensor model*

Description

NetworkChange implements Bayesian multiple changepoint models to network time series data using a degree-corrected multilinear tensor decomposition method

Usage

```
NetworkChange(Y, R = 2, m = 1, initial.s = NULL, mcmc = 100,
  burnin = 100, verbose = 0, thin = 1, degree.normal = "eigen",
  UL.Normal = "Orthonormal", DIC = FALSE, Waic = FALSE,
  marginal = FALSE, plotUU = FALSE, plotZ = FALSE, b0 = 0, B0 = 1,
  c0 = NULL, d0 = NULL, u0 = NULL, u1 = NULL, v0 = NULL, v1 = NULL,
  a = NULL, b = NULL)
```

Arguments

`Y`	Reponse tensor
`R`	Dimension of latent space. The default is 2.
`m`	Number of change point. If `m = 0` is specified, the result should be the same as `NetworkStatic`.
`initial.s`	The starting value of latent state vector. The default is sampling from equal probabilities for all states.
`mcmc`	The number of MCMC iterations after burnin.
`burnin`	The number of burn-in iterations for the sampler.
`verbose`	A switch which determines whether or not the progress of the sampler is printed to the screen. If `verbose` is greater than 0 the iteration number, the β vector, and the error variance are printed to the screen every `verbose`th iteration.
`thin`	The thinning interval used in the simulation. The number of MCMC iterations must be divisible by this value.
`degree.normal`	A null model for degree correction. Users can choose "NULL", "eigen" or "Lsym." "NULL" is no degree correction. "eigen" is a principal eigen-matrix consisting of the first eigenvalue and the corresponding eigenvector. " Lsym" is a modularity matrix. Default is "eigen."

UL.Normal	Transformation of sampled U. Users can choose "NULL", "Normal" or "Orthonormal." "NULL" is no normalization. "Normal" is the standard normalization. "Orthonormal" is the Gram-Schmidt orthgonalization. Default is "NULL."
DIC	If `DIC = TRUE`, the deviation information criterion is computed.
Waic	If `Waic = TRUE`, the Watanabe information criterion is computed.
marginal	If `marginal = TRUE`, the log marignal likelihood is computed using the method of Chib (1995).
plotUU	If `plotUU = TRUE` and `verbose > 0`, then the plot of the latent space will be printed to the screen at every `verbose`th iteration. The default is `plotUU = FALSE`.
plotZ	If `plotZ = TRUE` and `verbose > 0`, then the plot of the degree-corrected input matrix will be printed to the screen with the sampled mean values at every `verbose`th iteration. The default is `plotUU = FALSE`.
b0	The prior mean of β. This must be a scalar. The default value is 0.
B0	The prior variance of β. This must be a scalar. The default value is 1.
c0	= 0.1
d0	= 0.1
u0	$u_0/2$ is the shape parameter for the inverse Gamma prior on variance parameters for U. The default is 10.
u1	$u_1/2$ is the scale parameter for the inverse Gamma prior on variance parameters for U. The default is 1.
v0	$v_0/2$ is the shape parameter for the inverse Gamma prior on variance parameters for V. The default is 10.
v1	$v_1/2$ is the scale parameter for the inverse Gamma prior on variance parameters for V. The default is the time length of Y.
a	a is the shape1 beta prior for transition probabilities. By default, the expected duration is computed and corresponding a and b values are assigned. The expected duration is the sample period divided by the number of states.
b	b is the shape2 beta prior for transition probabilities. By default, the expected duration is computed and corresponding a and b values are assigned. The expected duration is the sample period divided by the number of states.

Value

An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The object contains an attribute `Waic.out` that contains results of WAIC and the log-marginal likelihood of the model (`logmarglike`). The object also contains an attribute `prob.state` storage matrix that contains the probability of $state_i$ for each period

References

Jong Hee Park and Yunkyun Sohn. 2017. "Detecting Structural Change in Network Time Series Data using Bayesian Inference." Working Paper.

Peter D. Hoff 2011. "Hierarchical Multilinear Models for Multiway Data." *Computational Statistics \& Data Analysis*. 55: 530-543.

Siddhartha Chib. 1998. "Estimation and comparison of multiple change-point models." *Journal of Econometrics*. 86: 221-241.

Sumio Watanabe. 2010. "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory." *Journal of Machine Learning Research*. 11: 3571-3594. Siddhartha Chib. 1995. "Marginal Likelihood from the Gibbs Output." *Journal of the American Statistical Association*. 90: 1313-1321.

See Also

NetworkStatic

Examples

```
   ## Not run:
   set.seed(1973)
   ## Generate an array (30 by 30 by 40) with block transitions
   from 2 blocks to 3 blocks
   Y <- MakeBlockNetworkChange(n=10, T=40, type ="split")
   G <- 100 ## only 100 mcmc scans to save time
   ## Fit models
   out0 <- NetworkStatic(Y, R=2, mcmc=G, burnin=G, verbose=G, Waic=TRUE)
   out1 <- NetworkChange(Y, R=2, m=1, mcmc=G, burnin=G, verbose=G, Waic=TRUE)
   out2 <- NetworkChange(Y, R=2, m=2, mcmc=G, burnin=G, verbose=G, Waic=TRUE)
   out3 <- NetworkChange(Y, R=2, m=3, mcmc=G, burnin=G, verbose=G, Waic=TRUE)
   outlist <- list(out0, out1, out2, out3)
   ## The true model is out1
   WaicCompare(outlist)
   ## plot latent node positions
   plotU(out1)
   ## plot layer-specific network generation rules
   plotV(out1)

## End(Not run)
```

NetworkStatic *Degree-corrected multilinear tensor model*

Description

NetworkStatic implements a degree-corrected Bayesian multilinear tensor decomposition method

Usage

```
NetworkStatic(Y, R = 2, mcmc = 100, burnin = 100, verbose = 0,
  thin = 1, degree.normal = "eigen", UL.Normal = "Orthonormal",
  plotUU = FALSE, plotZ = FALSE, b0 = 0, B0 = 1, c0 = NULL,
  d0 = NULL, u0 = NULL, u1 = NULL, v0 = NULL, v1 = NULL,
  marginal = FALSE, DIC = FALSE, Waic = FALSE)
```

Arguments

`Y`	Reponse tensor
`R`	Dimension of latent space. The default is 2.
`mcmc`	The number of MCMC iterations after burnin.
`burnin`	The number of burn-in iterations for the sampler.
`verbose`	A switch which determines whether or not the progress of the sampler is printed to the screen. If `verbose` is greater than 0 the iteration number, the β vector, and the error variance are printed to the screen every `verbose`th iteration.
`thin`	The thinning interval used in the simulation. The number of MCMC iterations must be divisible by this value.
`degree.normal`	A null model for degree correction. Users can choose "NULL", "eigen" or "Lsym." "NULL" is no degree correction. "eigen" is a principal eigen-matrix consisting of the first eigenvalue and the corresponding eigenvector. " Lsym" is a modularity matrix. Default is "eigen."
`UL.Normal`	Transformation of sampled U. Users can choose "NULL", "Normal" or "Orthonormal." "NULL" is no normalization. "Normal" is the standard normalization. "Orthonormal" is the Gram-Schmidt orthgonalization. Default is "NULL."
`plotUU`	If `plotUU = TRUE` and `verbose > 0`, then the plot of the latent space will be printed to the screen at every `verbose`th iteration. The default is `plotUU = FALSE`.
`plotZ`	If `plotZ = TRUE` and `verbose > 0`, then the plot of the degree-corrected input matrix will be printed to the screen with the sampled mean values at every `verbose`th iteration. The default is `plotUU = FALSE`.
`b0`	The prior mean of β. This must be a scalar. The default value is 0.
`B0`	The prior variance of β. This must be a scalar. The default value is 1.
`c0`	= 0.1
`d0`	= 0.1
`u0`	$u_0/2$ is the shape parameter for the inverse Gamma prior on variance parameters for U. The default is 10.
`u1`	$u_1/2$ is the scale parameter for the inverse Gamma prior on variance parameters for U. The default is 1.
`v0`	$v_0/2$ is the shape parameter for the inverse Gamma prior on variance parameters for V. The default is 10.
`v1`	$v_1/2$ is the scale parameter for the inverse Gamma prior on variance parameters for V. The default is the time length of Y.
`marginal`	If `marginal = TRUE`, the log marignal likelihood is computed using the method of Chib (1995).
`DIC`	If `DIC = TRUE`, the deviation information criterion is computed.
`Waic`	If `Waic = TRUE`, the Watanabe information criterion is computed.

Value

An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The object contains an attribute `Waic.out` that contains results of WAIC and the log-marginal likelihood of the model (`logmarglike`).

References

Jong Hee Park and Yunkyun Sohn. 2017. "Detecting Structural Change in Network Time Series Data using Bayesian Inference." Working Paper.

Peter D. Hoff 2011. "Hierarchical Multilinear Models for Multiway Data." *Computational Statistics \& Data Analysis*. 55: 530-543.

Sumio Watanabe. 2010. "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory." *Journal of Machine Learning Research*. 11: 3571-3594. Siddhartha Chib. 1995. “Marginal Likelihood from the Gibbs Output.” *Journal of the American Statistical Association*. 90: 1313-1321.

See Also

NetworkChange

Examples

```
   ## Not run:
   set.seed(1973)

   ## generate an array with two constant blocks
   Y <- MakeBlockNetworkChange(n=10, shape=10, T=40, type ="constant")
   out0 <- NetworkStatic(Y, R=2, mcmc=10, burnin=10,
   verbose=10, UL.Normal = "Orthonormal")

   ## contour plot of latent node positions
   plotContour(out0)

   ## plot latent node positions
   plotU(out0)

   ## plot layer-specific network connection rules
   plotV(out0)

## End(Not run)
```

---

plotContour *Contour plot of latent node positions*

Description

Draw a contour plot of latent node positions

Usage

```
plotContour(OUT, main = "", k = 8, my.cols = brewer.pal(k, "Spectral"))
```

Arguments

`OUT`	Output of networkchange objects.
`main`	The title of plot
`k`	The number of levels (nlevels in contour ()).
`my.cols`	Color scale. Use brewer.pal() from RColorBrewer.

Value

A plot object

Examples

```
   ## Not run: set.seed(1973)
   ## generate an array with two constant blocks
   Y <- MakeBlockNetworkChange(n=10, shape=10, T=40, type ="constant")
   out0 <- NetworkStatic(Y, R=2, mcmc=10, burnin=10,
   verbose=10, UL.Normal = "Orthonormal")
   ## contour plot of latent node positions
   plotContour(out0)

## End(Not run)
```

`plotU` *Plot of latent node positions*

Description

Plot latent node positions.

Usage

```
plotU(OUT, Year = NULL, names = NULL, main = "")
```

Arguments

`OUT`	Output of networkchange objects.
`Year`	Starting of the time period. If NULL, 1.
`names`	Node names. If NULL, use natural numbers.
`main`	The title of plot

Value

A plot object

Examples

```
    ## Not run: set.seed(1973)
    ## generate an array with two constant blocks
    Y <- MakeBlockNetworkChange(n=10, shape=10, T=40, type ="constant")
    out0 <- NetworkStatic(Y, R=2, mcmc=10, burnin=10,
    verbose=10, UL.Normal = "Orthonormal")
    ## latent node positions
    plotU(out0)

## End(Not run)
```

`plotV` *Plot of layer-specific network generation rules.*

Description

Plot layer-specific network generation rules.

Usage

```
plotV(OUT, main = "", cex = 2)
```

Arguments

`OUT`	Output of networkchange objects.
`main`	The title of plot
`cex`	point size

Value

A plot object

Examples

```
    ## Not run: set.seed(1973)
    ## generate an array with two constant blocks
    Y <- MakeBlockNetworkChange(n=10, shape=10, T=40, type ="constant")
    out0 <- NetworkStatic(Y, R=2, mcmc=10, burnin=10,
    verbose=10, UL.Normal = "Orthonormal")
    ## latent node positions
    plotV(out0)

## End(Not run)
```

`startS` *Sample a starting value of hidden states*

Description

Sample a starting value of hidden states

Usage

```
startS(Z, Time, m, initial.U, V, s2, R)
```

Arguments

`Z`	Degree-corrected network array data
`Time`	The length of time.
`m`	The number of breaks
`initial.U`	Initialized U matrix.
`V`	Initialized V matrix.
`s2`	Initialized error variance
`R`	The dimensionality of latent space

Value

A state vector

`startUV` *Starting values of U and V*

Description

Initialize starting values of U and V

Usage

```
startUV(Z, R, K)
```

Arguments

`Z`	Degree-corrected network array data.
`R`	The dimensionality of latent space.
`K`	The dimensionality of Z.

Value

A list of U and V

ULUstateSample *Hidden State Sampler*

Description

Sample hidden states from hidden Markov multilinear model

Usage

```
ULUstateSample(m, s, ZMUt, s2, P, SOS.random)
```

Arguments

`m`	The number of break
`s`	Latent state vector
`ZMUt`	Z - MU
`s2`	error variance
`P`	Transition matrix
`SOS.random`	single observation state random perturbation

Value

A list of a state vector, state probabilities, and SOS.random.

updateb *Update time-constant regression parameters*

Description

Update time-constant regression parameters

Usage

```
updateb(Z, MU, s2, XtX, b0, B0)
```

Arguments

`Z`	Degree corrected response tensor
`MU`	Mean array
`s2`	Error variance
`XtX`	X'X
`b0`	Prior mean of beta
`B0`	Prior variance of beta

Value

A vector of regression parameters

`updatebm` *Update regime-changing regression parameters*

Description

Update regime-changing beta

Usage

```
updatebm(ns, K, s, s2, B0, p, ZU)
```

Arguments

`ns`	The number of hidden states
`K`	The dimensionality of Z
`s`	Latent state vector
`s2`	The variance of error
`B0`	The prior variance of beta
`p`	The rank of X
`ZU`	Z - ULU

Value

A vector of regime-changing regression parameters

`updateP` *Update transition matrix*

Description

Update transition matrix

Usage

```
updateP(s, ns, P, A0)
```

Arguments

`s`	Latent state vector
`ns`	The number of hidden states
`P`	Transition matrix
`A0`	Prior of transition matrix

Value

A transtion matrix

---

updateS *Update latent states*

Description

Update latent states

Usage

```
updateS(iter, s, V, m, Zb, Zt, Time, MU.state, P, s2, N.upper.tri,
  random.perturb)
```

Arguments

`iter`	iteration number
`s`	the most recent latent states
`V`	Network generation rules
`m`	The number of breaks
`Zb`	Z - b
`Zt`	Z stacked by time
`Time`	The length of time
`MU.state`	UVU for each state
`P`	Transition matrix
`s2`	error variance
`N.upper.tri`	The number of upper triangular elements
`random.perturb`	If `random.perturb` = TRUE and a single state observation is found, the latent state is randomly selected by equal weights.

Value

A list of vectors containing latent states and their probabilities

---

`updates2m` *Update regime-specific variance*

Description

Update regime-specific variance parameter

Usage

```
updates2m(ns, Zm, MU, c0, d0, Km)
```

Arguments

`ns`	The number of hidden states
`Zm`	The regime-specific holder of Z - beta
`MU`	The mean array.
`c0`	Scalar shape parameter
`d0`	Scalar scale parameter
`Km`	Regime-specific dimensions

Value

A scalar for a regime-specific variance

`updateU` *Update time-constant latent node positions*

Description

Update time-constant latent node positions

Usage

```
updateU(K, U, V, R, Zb, s2, eU, iVU)
```

Arguments

`K`	The dimensionality of Z
`U`	The most recent draw of latent node positions
`V`	Layer-specific network generation rule
`R`	The dimensionality of latent space
`Zb`	Z - beta
`s2`	error variance
`eU`	The mean of U
`iVU`	The variance of U

Value

A matrix of time-constant latent node positions

`updateUm` *Regime-specific latent node positions*

Description

Update regime-specific latent node positions.

Usage

```
updateUm(ns, U, V, R, Zm, Km, ej, s2, eU, iVU, UL.Normal)
```

Arguments

`ns`	The number of latent states
`U`	THe latent node positions
`V`	Layer-specific network generation rule.
`R`	The dimensionality of latent space
`Zm`	Regim-specific Z - beta
`Km`	The dimension of regime-specific Z.
`ej`	Regime indicator.
`s2`	The variance of error.
`eU`	The regim-specific mean of U.
`iVU`	The regim-specific variance of U.
`UL.Normal`	Normalization method for U. "Normal" or "Orthonormal" are supported.

Value

A matrix of regime-specific latent node positions

updateV — *Update layer specific network generation rules*

Description

Update layer specific network generation rules

Usage

```
updateV(Zb, U, R, K, s2, eV, iVV, UTA)
```

Arguments

`Zb`	Z - beta.
`U`	The latent node positions.
`R`	The dimension of latent space.
`K`	The dimension of Z.
`s2`	The variance of error.
`eV`	The mean of V.
`iVV`	The variance of V.
`UTA`	Indicator of upper triangular array

Value

A matrix of layer specific network generation rules

updateVm — *Update V from a change-point network process*

Description

Update layer specific network generation rules from a change-point network process

Usage

```
updateVm(ns, U, V, Zm, Km, R, s2, eV, iVV, UTA)
```

Arguments

`ns`	The number of hidden regimes.
`U`	The latent node positions.
`V`	The layer-specific network generation rule.
`Zm`	The holder of Z - beta.
`Km`	The dimension of regime-specific Z.
`R`	The dimension of latent space.
`s2`	The variance of error.
`eV`	The mean of V
`iVV`	The variance of V
`UTA`	Indicator of upper triangular array

Value

A matrix of regime-specific layer specific network generation rules

---

`WaicCompare` *Compare WAIC*

Description

Compare Widely Applicable Information Criterion

Usage

```
WaicCompare(outlist)
```

Arguments

`outlist`	List of NetworkChange objects

Value

out

A matrix of log marginal likelihoods.

References

Sumio Watanabe. 2010. "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory." *Journal of Machine Learning Research*. 11: 3571-3594.

Andrew Gelman, Jessica Hwang, and Aki Vehtari. 2014. "Understanding predictive information criteria for Bayesian models." *Statistics and Computing*. 24(6):997-1016.

See Also

`MarginalCompare`

Index