Package ‘lvm4net’

February 20, 2015

Title  Latent Variable Models for Networks

Description  Latent variable models for network data using fast inferential
             procedures.

Version  0.1

Author  Isabella Gollini

Maintainer  Isabella Gollini <igollini.stats@gmail.com>

Depends  R (>= 3.0.2), ergm, ellipse

Imports  igraph

License  GPL (>= 2)

LazyData  true

NeedsCompilation  no

Repository  CRAN

Date/Publication  2014-07-04 13:39:23

R topics documented:

  lvm4net-package ......................................................... 2
  boxroc .......................................................... 2
  lsmj ............................................................ 4
  lsm ............................................................ 5
  plot.lsmj ......................................................... 7
  plot.lsm ......................................................... 8
  plotY .......................................................... 9
  PPIgen .......................................................... 10
  PPInet ......................................................... 11
  PPIphy ........................................................ 12
  rotXtoY ........................................................ 12

Index  13
Description

lvm4net provides a range of tools for latent variable models for network data. Most of the models are implemented using a fast variational inference approach. Latent space models for binary networks: the function `lsm` implements the latent space model (LSM) introduced by Hoff et al. (2002) using a variational inference and squared Euclidian distance; the function `lsjm` implements latent space joint model (LSJM) for multiplex networks introduced by Gollini and Murphy (2014). These models assume that each node of a network has a latent position in a latent space: the closer two nodes are in the latent space, the more likely they are connected. Functions for binary bipartite networks will be added soon.

References


---

boxroc

**Boxplot and Roc Curves**

Description

Function to display boxplots and ROC curves to show model fit in terms of in-sample link prediction.

Usage

boxroc(Y, EZ, xiT, BOXPLOT = FALSE, ROC = FALSE, Lroc = 100, labelsPlot = NULL, powdist = 2, cexRocLeg = 0.8, colRoc = seq(2, Ndata + 1), ltyRoc = seq(2, Ndata + 1), lwdRoc = 2, ...)

Arguments

- `Y` 
  \( N \times N \) binary adjacency matrix, or list containing the adjacency matrices.
- `EZ` 
  \( N \times D \) matrix (or list of matrices) containing the posterior means of the latent positions
- `xiT` 
  vector of posterior means of the parameter \( \alpha \)
- `BOXPLOT` 
  logical; if TRUE draws the boxplot. Default `BOXPLOT = FALSE`
- `ROC` 
  logical; if TRUE draws the ROC curve. Default `ROC = FALSE`
**boxroc**

- **Lroc** number of intervals in the ROC curve. Default `Lroc = 100`
- **labelsPlot** main title for the boxplot. Default `labelsPlot = NULL`
- **powdist** vector of power of the distance default `powdist = 2`, squared euclidean distance, the alternative is 1, for the Euclidean distance
- **cexRocLeg** cex for the ROC curve. Default `cexRocLeg = .8`
- **colRoc** col for the ROC curve. Default `colRoc = seq(2, Ndata + 1)`
- **ltyRoc** lty for the ROC curve. Default `ltyRoc = seq(2, Ndata + 1)`
- **lwdRoc** lwd for the ROC curve. Default `lwdRoc = 2`
- **...** Arguments to be passed to methods, such as graphical parameters (see `par`).

**Value**

The area under the ROC curve (AUC) and the selected plots. The closer the AUC takes values to 1 the better the fit.

**References**


**See Also**

- `lsm`, `lsjm`

**Examples**

```r
n <- 20
Y <- network(n, directed = FALSE)[,]

modLSM <- lsm(Y, D = 2)
bp <- boxroc(Y,
EZ = modLSM$lsmEZ,
xiT = modLSM$xiT,
Lroc = 150,
ROC = TRUE,
BOXPLOT = TRUE)

print(bp)
```
**lsjm**

*Latent Space Joint Model*

**Description**

Function to joint modelling of multiple network views using the Latent Space Joint Model (LSJM) Gollini and Murphy (2014). The LSJM merges the information given by the multiple network views by assuming that the probability of a node being connected with other nodes in each view is explained by a unique latent variable.

**Usage**

```r
lsjm(y, D, sigma = 1, xi = rep(0, length(y)), psi2 = rep(2, length(y)),
Niter = 500, tol = 0.1^2, preit = 20, randomZ = FALSE)
```

**Arguments**

- `y` list containing a \(N \times N\) binary adjacency matrix for each network view.
- `D` integer dimension of the latent space
- `sigma` \(D \times D\) variance/covariance matrix of the prior distribution for the latent positions. Default `sigma = 1`
- `xi` vector of means of the prior distributions of \(\alpha\). Default `xi = 0`
- `psi2` vector of variances of the prior distributions of \(\alpha\). Default `psi2 = 2`
- `Niter` maximum number of iterations. Default `Niter = 500`
- `tol` desired tolerance. Default `tol = 0.1^2`
- `preit` Preliminary number of iterations default `preit = 20`
- `randomZ` logical; If `randomZ = TRUE` random initialization for the latent positions is used. If `randomZ = FALSE` and \(D = 2\) or \(3\) the latent positions are initialized using the Fruchterman-Reingold method and multidimensional scaling is used for \(D = 1\) or \(D > 3\). Default `randomZ = FALSE`

**Value**

List containing:

- `ez` \(N \times D\) matrix containing the posterior means of the latent positions
- `vz` \(D \times D\) matrix containing the posterior variance of the latent positions
- `lsmez` list containing a \(N \times D\) matrix for each network view containing the posterior means of the latent positions under each model in the latent space.
- `lsmvz` list containing a \(D \times D\) matrix for each network view containing the posterior variance of the latent positions under each model in the latent space.
- `xit` vector of means of the posterior distributions of \(\alpha\)
- `psiRt` vector of variances of the posterior distributions of \(\alpha\)
- `ell` expected log-likelihood
References


Examples

```
## Simulate Undirected Network
n <- 20
ndata <- 2
Y <- list()
Y[[1]] <- network(n, directed = FALSE)
### create a new view that is similar to the original
for(nd in 2:ndata){
  Y[[nd]] <- Y[[nd - 1]] - sample(c(-1, 0, 1), n * n, replace = TRUE, prob = c(0.05, 0.85, 0.1))
  Y[[nd]] <- 1 * (Y[[nd]] > 0)
  diag(Y[[nd]]) <- 0
}
par(mfrow = c(1, 2))
z <- plotY(Y[[1]], verbose = TRUE, main = 'Network 1')
plotY(Y[[2]], EZ = z, main = 'Network 2')
par(mfrow = c(1, 1))

modLSJM <- lsmj(Y, D = 2)
plot(modLSJM, Y, drawCB = TRUE)
plot(modLSJM, Y, drawCB = TRUE, plotZtilde = TRUE)
```

Description

Latent space models (LSM) are a well known family of latent variable models for network data introduced by Hoff et al. (2002) under the basic assumption that each node has an unknown position in a D-dimensional Euclidean latent space: generally the smaller the distance between two nodes in the latent space, the greater the probability of them being connected. Unfortunately, the posterior distribution of the LSM cannot be computed analytically. For this reason we propose a variational inferential approach which proves to be less computationally intensive than the MCMC procedure proposed in Hoff et al. (2002) (implemented in the latentnet package) and can therefore easily handle large networks. Salter-Townshend and Murphy (2013) applied variational methods to fit the LSM with the Euclidean distance in the vblpcm package. In this package, a distance model with squared Euclidean distance is used. We follow the notation of Gollini and Murphy (2014).

Usage

```
lsmd(Y, D, sigma = 1, xi = 0, psi2 = 2, Niter = 500, Miniter = 10,
tol = 0.1^2, randomZ = FALSE)
```
Arguments

- **Y**  
  \( N \times N \) binary adjacency matrix
- **D**  
  integer dimension of the latent space
- **sigma**  
  \( D \times D \) variance/covariance matrix of the prior distribution for the latent positions. Default \( \text{sigma} = 1 \)
- **xi**  
  mean of the prior distribution of \( \alpha \). Default \( \text{xi} = 0 \)
- **psi2**  
  variance of the prior distribution of \( \alpha \). Default \( \text{psi2} = 2 \)
- **Niter**  
  maximum number of iterations. Default \( \text{Niter} = 500 \)
- **Miniter**  
  minimum number of iterations. Default \( \text{Miniter} = 10 \)
- **tol**  
  desired tolerance. Default \( \text{tol} = 0.1^2 \)
- **randomz**  
  logical; If \( \text{randomz} = \text{TRUE} \) random initialization for the latent positions is used. If \( \text{randomz} = \text{FALSE} \) and \( D = 2 \) or \( 3 \) the latent positions are initialized using the Fruchterman-Reingold method and multidimensional scaling is used for \( D = 1 \) or \( D > 3 \). Default \( \text{randomz} = \text{FALSE} \)

Value

List containing:
- **lsmez**  
  \( N \times D \) matrix containing the posterior means of the latent positions
- **lsmvz**  
  \( D \times D \) matrix containing the posterior variance of the latent positions
- **xit**  
  mean of the posterior distribution of \( \alpha \)
- **psiRt**  
  variance of the posterior distribution of \( \alpha \)
- **ell**  
  expected log-likelihood

References


See Also

plot.lsm

Examples

```r
## Simulate Undirected Network
n <- 20
Y <- network(n, directed = FALSE)[,]

modLSM <- lsm(Y, D = 2)
plot(modLSM, Y)
```
plot.lsjm

Two dimensional plot of Latent Space Joint Model output

Description

Function to plot an object of class 'lsjm'

Usage

## S3 method for class 'lsjm'
plot(x, y, drawCB = FALSE, dimZ = c(1, 2),
     plotZtilde = FALSE, colPl = 1, colEll = rgb(0.6, 0.6, 0.6, alpha = 0.1),
     LEVEL = 0.95, pchplot = 20, pchEll = 19, pchPl = 19, cexPl = 1.1,
     mainZtilde = NULL, arrowhead = FALSE, curve = NULL, xlim = NULL,
     ylim = NULL, main = NULL, ...)  

Arguments

x          object of class 'lsjm'
Y          list containing a $N \times N$ binary adjacency matrix for each network view.
drawCB     logical if drawCB = TRUE draw confidence bounds
dimZ       dimensions of the latent variable to be plotted. Default dimZ = c(1, 2)
plotZtilde if TRUE do the plot for the last step of LSM
colPl      col for the points representing the nodes. Default colPl = NULL
colEll     col for the ellipses. Default rgb(.6, .6, .6, alpha=.1)
LEVEL      levels of confidence bounds shown when plotting the ellipses. Default LEVEL = .95
pchplot    Default pchplot = 20
pchEll     pch for the ellipses. Default pchEll = 19
pchPl      pch for the points representing the nodes. Default pchPl = 19
cexPl      cex for the points representing the nodes. Default cexPl = 1.1
mainZtilde title for single network plots TRUE do the plot for the last step of LSM
arrowhead  logical, if the arrowed are to be plotted. Default arrowhead = FALSE
curve      curvature of edges. Default curve = 0
xlim       range for x
ylim       range for y
main       main title
...        Arguments to be passed to methods, such as graphical parameters (see par).
Examples

```r
## Simulate Undirected Network
n <- 20
Ndata <- 2
Y <- list()
Y[[1]] <- network(n, directed = FALSE)
## create a new view that is similar to the original
for(nd in 2:Ndata){
  Y[[nd]] <- Y[[nd - 1]] - sample(c(-1, 0, 1), n * n, replace = TRUE, prob = c(.05, .85, .1))
  Y[[nd]] <- 1 * (Y[[nd]] > 0)
  diag(Y[[nd]]) <- 0
}

par(mfrow = c(1, 2))
z <- plot(y[[1]], verbose = TRUE, main = 'Network 1')
plot(y[[2]], EZ = z, main = 'Network 2')
par(mfrow = c(1, 1))

modLSJM <- lsjm(Y, D = 2)
plot(modLSJM, Y, drawCB = TRUE)
plot(modLSJM, Y, drawCB = TRUE, plotZtilde = TRUE)
```

---

### plot.lsm

Two dimensional plot of the Latent Space Model output

**Description**

Function to plot an object of class 'lsm'

**Usage**

```r
## S3 method for class 'lsm'
plot(x, Y, drawCB = FALSE, dimZ = c(1, 2), colPl = 1,
     colEll = rgb(0.6, 0.6, 0.6, alpha = 0.1), LEVEL = 0.95, pchplot = 20,
     pchEll = 19, pchPl = 19, cexPl = 1.1, arrowhead = FALSE,
     curve = NULL, xlim = NULL, ylim = NULL, ...)```

**Arguments**

- `x` object of class 'lsm'
- `Y` $N \times N$ binary adjacency matrix
- `drawCB` draw confidence bounds
- `dimZ` dimensions of the latent variable to be plotted. Default `dimZ = c(1, 2)`
- `colPl` col for the points representing the nodes. Default `colPl = NULL`
- `colEll` col for the ellipses. Default `rgb(0.6, 0.6, 0.6, alpha=.1)`
- `LEVEL` levels of confidence bounds shown when plotting the ellipses. Default `LEVEL = 0.95`
`plotY`  

Default `pchplot = 20`  
`pchEll`  
`pchpl`  
`cexpl`  
`arrowhead`  
`curve`  
`xlim`  
`ylim`  
...  
Arguments to be passed to methods, such as graphical parameters (see `par`).

**Examples**

```r
n <- 20
Y <- network(n, directed = FALSE)[,]
modLSM <- lsm(Y, D = 2)
plot(modLSM, Y)

# Plot with 95% CB
plot(modLSM, Y, drawCB = TRUE)
# Plot with 99% CB
plot(modLSM, Y, drawCB = TRUE, LEVEL = .99)
```

---

**plotY**  

*Plot the adjacency matrix of the network*

**Description**

Function to plot the adjacency matrix of the network.

**Usage**

```r
plotY(Y, Ndata = NULL, EZ = NULL, VZ = NULL, dimZ = c(1, 2),
  labels = NULL, colPl = 1, colEll = rgb(0.6, 0.6, 0.6, alpha = 0.1),
  LEVEL = 0.95, pchplot = 20, pchEll = 19, pchPl = 19, cexPl = 1.1,
  arrowhead = FALSE, curve = NULL, lwdLine = 0.3, xlim = NULL,
  ylim = NULL, verbose = FALSE, ...)
```

**Arguments**

- `Y`  
  list, or matrix containing a $N \times N$ binary adjacency matrix for each network view.
- `Ndata`  
  number of network views
- `EZ`  
  posterior mean latent positions
- `VZ`  
  posterior variance latent positions, if specified draw ellipse
**dimZ**
dimensions of Z to be plotted, default \(\text{dimZ} = c(1, 2)\)

**labels**
text to be added in the plot representing the labels of each node. Default \(\text{labels} = \text{NULL}\), no labels are shown

**colPl**
col for the points representing the nodes. Default \(\text{colPl} = \text{NULL}\)

**colEll**
col for the ellipses. Default \(\text{rgb}(.6, .6, .6, \text{alpha} = .1)\)

**LEVEL**
levels of confidence bounds shown when plotting the ellipses. Default \(\text{LEVEL} = .95\)

**pchplot**
Default \(\text{pchplot} = 20\)

**pchEll**
pch for the ellipses. Default \(\text{pchEll} = 19\)

**pchPl**
pch for the points representing the nodes. Default \(\text{pchPl} = 19\)

**cexPl**
cex for the points representing the nodes. Default \(\text{cexPl} = 1.1\)

**arrowhead**
logical, if the arrowed are to be plotted. Default \(\text{arrowhead} = \text{FALSE}\)

**curve**
curvature of edges. Default \(\text{curve} = 0\)

**lwdLine**
lwd of edges. Default \(\text{lwdLine} = .3\)

**xLim**
rangle for x

**yLim**
rangle for y

**verbose**
if \(\text{verbose} = \text{TRUE}\) save the nodal positions

**Examples**

```r
n <- 20
Y <- network(n, directed = FALSE)[,]
plotY(Y)
# Store the positions of nodes used to plot Y, in order to redraw the plot using the same positions
z <- plotY(Y, verbose = TRUE)
plotY(Y, EZ = z)
```

---

**PPIgen**

**PPI genetic interactions**

---

**Description**

The dataset contains a network formed by genetic protein-protein interactions (PPI) between 67 Saccharomyces cerevisiae proteins. The network is formed of 294 links. The data were downloaded from the Biological General Repository for Interaction Datasets (BioGRID) database [http://thebiogrid.org/](http://thebiogrid.org/)

**Format**

Binary adjacency matrix

**Details**

Binary adjacency matrix containing genetic interactions between 67 proteins.
References


See Also

PPIphy

PPInet

PPI genetic and physical interactions data

Description

The dataset contains two undirected networks formed by genetic and physical protein-protein interactions (PPI) between 67 Saccharomyces cerevisiae proteins. The genetic interactions network is formed of 294 links, and the physical interactions network is formed of 190 links. The data were downloaded from the Biological General Repository for Interaction Datasets (BioGRID) database http://thebiogrid.org/

Format

Two binary adjacency matrices

Details

- **PPIgen** Binary adjacency matrix containing genetic interactions between 67 proteins.
- **PPIphy** Binary adjacency matrix containing physical interactions between 67 proteins.

Source

http://thebiogrid.org/

References


See Also

PPIgen, PPIphy
**PPIphy**  
*PPI physical interactions*

**Description**

The dataset contains a network formed by physical protein-protein interactions (PPI) between 67 *Saccharomyces cerevisiae* proteins. The network is formed of 190 links. The data were downloaded from the Biological General Repository for Interaction Datasets (BioGRID) database [http://thebiogrid.org/](http://thebiogrid.org/)

**Format**

Binary adjacency matrix

**Details**

Binary adjacency matrix containing physical interactions between 67 proteins.

**References**


**See Also**

PPIgen

---

**rotXtoY**  
*Rotate X to match Y*

**Description**

Function to rotate X to match Y via singular value decomposition

**Usage**

`rotXtoY(X, Y)`

**Arguments**

- `X`  
  matrix to be rotated
- `Y`  
  objective matrix

**Value**

rotated object $X_{\text{rot}}$, and the rotation matrix $R$
Index

*Topic datasets
  - PPIgen, 10
  - PPInet, 11
  - PPIphy, 12

boxroc, 2

lsjm, 2, 3, 4
lsm, 2, 3, 5
lvm4net (lvm4net-package), 2
lvm4net-package, 2

par, 3, 7, 9, 10
plot.lsjm, 7
plot.lsm, 6, 8
plotY, 9
PPIgen, 10, 11, 12
PPInet, 11
PPIphy, 11, 12

rotXtoY, 12