Package ‘dyads’

October 13, 2022

Type Package
Title Dyadic Network Analysis
Version 1.2.1
Date 2022-08-16
Author Bonne J.H. Zijlstra <B.J.H.Zijlstra@uva.nl>
Maintainer Bonne J.H. Zijlstra <B.J.H.Zijlstra@uva.nl>
Depends R (>= 3.0.0)
Imports stats, CholWishart, MASS, RcppZiggurat, Rfast, mvtnorm
Suggests plyr
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2022-08-17 07:50:02 UTC

R topics documented:

dyads-package ......................................................... 2
b2ML ................................................................. 3
j2 ................................................................. 6
p2 ................................................................. 7
p2ML ............................................................. 9

Index 12
Description

Package for Dyadic Network Analysis.

Details

Package: dyads
Type: Package
Title: Dyadic Network Analysis
Version: 1.2.1
Date: 2022-08-16
Author: Bonne J.H. Zijlstra <B.J.H.Zijlstra@uva.nl>
Maintainer: Bonne J.H. Zijlstra <B.J.H.Zijlstra@uva.nl>
Depends: R (>= 3.0.0)
Imports: stats, CholWishart, MASS, RcppZiggurat, Rfast, mvtnorm
Suggests: plyr
License: GPL (>= 2)

Index of help topics:

b2ML MCMC estimates for the (multilevel) b2 model
dyads-package dyads
j2 MCMC estimates for the j2 model
p2 MCMC estimates for the p2 model
p2ML MCMC estimates for the (multilevel) p2 model


Author(s)

Bonne J.H. Zijlstra Maintainer: Bonne J.H. Zijlstra <B.J.H.Zijlstra@uva.nl>

References

Examples

# create a very small network with covariates for illustrative purposes
S <- c(1,0,1,0,1,0,1,0,1,0)
REC <- (S*-1)+1
D1 <- matrix(c(0,1,0,1,1,0,1,0,1,0,
  0,0,1,0,1,0,1,0,1,0,
  1,1,0,1,0,0,0,0,0,0,
  1,1,1,0,1,0,0,0,0,0,
  1,0,1,0,0,1,0,1,0,1,
  0,0,0,0,0,1,1,1,1,1,
  0,0,0,0,1,0,1,0,1,0,
  0,0,0,0,0,1,0,1,0,1,
  1,0,0,0,1,1,0,1,1,1,
  0,1,0,1,0,1,0,1,0,0,
  1,0,1,1,0,0,0,0,0,0), ncol=10)
D2 <- abs(matrix(rep(S,10), byrow = FALSE, ncol= 10) -
          matrix(rep(REC,10), byrow = TRUE, ncol= 10))
R <- D1*t(D1)
Y <- matrix(c(0,1,1,1,1,1,0,0,1,1,
  0,0,0,1,1,0,0,1,0,0,
  1,1,1,0,1,0,1,0,1,0,
  1,1,1,0,1,0,1,0,1,0,
  1,1,1,0,1,0,1,0,1,0,
  0,1,1,1,0,1,1,1,1,1,
  1,0,1,0,1,0,1,0,1,0,
  0,1,1,1,1,0,1,1,1,1,
  1,0,1,1,0,1,1,1,1,1,
  1,1,1,0,0,1,1,1,1,0), ncol=10)

# estimate p2 model
p2(Y, sender = ~ S, receiver = ~ REC, density = ~ D1 + D2, reciprocity= ~ R,
   burnin = 100, sample = 400, adapt = 10)
# Notice: burn-in, sample size and number of adaptive sequences are
# much smaller than recommended to keep computation time low.
# recommended code:
## Not run:
p2(Y, sender = ~ S, receiver = ~ REC, density = ~ D1 + D2, reciprocity= ~ R)

## End(Not run)

b2ML

MCMC estimates for the (multilevel) b2 model

Description

Estimates the (multilevel) b2 model parameters, which is the bidirectional counterpart of the multilevel p2 model as described in Zijlstra, Van Duijn and Snijders (2006) <doi: 10.1027/1614-2241.2.1.42>.
Usage

b2ML(nets, actor = NULL, density = NULL, adapt = NULL, burnin = NULL, center = NULL, separate = NULL, densVar = NULL, seed = NULL)

Arguments

- **nets**: List with n dichotomous symmetric dependent networks.
- **actor**: Optional matrix with a stacked actor covariate, corresponding to the n networks. Multiple actor covariates can be added as a formula object, see example below.
- **density**: Optional matrix with symmetric a stacked density covariate, with dimensions similar to the n dependent networks. Multiple density covariates can be added as a formula object, see example below.
- **adapt**: Optional number of adaptive sequenses (default is 100).
- **burnin**: Optional specification of number of burn-in iterations (default is 5000).
- **center**: Optional argument for centering predictors (default is TRUE).
- **separate**: Optional argument for estimating separate coefficients for the n dependent networks (default is FALSE).
- **densVar**: Optional argument for estimating densty variance at the network level (default is TRUE).
- **seed**: Optiona specification of random seed (default is 1).

Value

Returns a fitted model of class b2ML, to be opened with the function summary().

Author(s)

Bonne J.H. Zijlstra <b.j.h.zijlstra@uva.nl>

References


Examples

# create two very small networks with covariates for illustrative purposes
Y1 <- matrix(c( 0,1,1,1,1,1,0,
               1,0,1,1,1,1,1,
               1,1,0,1,1,1,0,1,
               1,0,0,1,0,1,0,0,
               1,1,1,0,1,1,0,1,1,
               1,1,0,1,0,1,0,1,1,
               1,1,1,0,1,1,0,1,1,
               1,1,1,0,1,0,1,0,1,1,
               1,1,1,0,0,1,0,1,0,1),
              nrow = 9, ncol = 5, byrow = TRUE)
b2ML

Y1 <- matrix(c( 1,1,0,0,1,1,0,0,1,0, ncol=10)
0,1,1,0,1,1,1,1,1,0), ncol=10)

Y2 <- matrix(c( 0,0,1,0,1,1,0,1,0,0,
0,0,0,0,0,0,1,1,0,0,
1,0,0,1,0,1,0,1,0,0,
0,0,1,0,0,0,1,1,0,0,
1,0,0,0,0,0,1,1,0,0,
1,0,1,0,0,0,1,1,0,0,
0,0,0,1,1,0,1,0,0,
1,1,1,1,1,1,0,1,1,
0,1,0,0,0,0,0,1,0,0,
0,0,0,0,0,0,0,1,0,0), ncol=10)

Y <- list(Y1, Y2)

Aa1 <- c(1,0,1,0,1,1,0,1,0,1)
Aa2 <- c(1,0,0,1,0,0,1,1,0,1)
Aa <- list(Aa1, Aa2)
Aat <- do.call(plyr::rbind.fill.matrix, Aa)

Ab1 <- c(0,0,0,0,0,0,0,0,0,0)
Ab2 <- c(1,1,1,1,1,1,1,1,1,1)
Ab <- list(Ab1, Ab2)
Abt <- do.call(plyr::rbind.fill.matrix, Ab)

Da1 <- abs(matrix(rep(Aa1,10), byrow = FALSE, ncol= 10) -
matrix(rep(Aa1,10), byrow = TRUE, ncol=10))
Da2 <- abs(matrix(rep(Aa2,10), byrow = FALSE, ncol= 10) -
matrix(rep(Aa2,10), byrow = TRUE, ncol=10))
Da <- list(Da1, Da2)
Dat <- do.call(plyr::rbind.fill.matrix, Da)

# estimate b2 model for two networks
M1 <- b2ML(Y,actor= ~ Aat + Abt, density = ~ Dat, adapt = 10, burnin = 100, densVar = FALSE)
summary(M1)
# Notice: burn-in, and number of adaptive sequences are
# much smaller than recommended to keep computation time low.
# recommended code:
## Not run:
M1 <- b2ML(Y,actor= ~ Aat + Abt, density = ~ Dat, densVar = FALSE)
summary(M1)
## End(Not run)

# estimate b2 model for a single network
M2 <- b2ML(list(Y[[1]]),actor= ~ Aat[1:10,], density = ~ Dat[1:10,], adapt = 10, burnin = 100,
densVar = FALSE)
summary(M2)
# Notice: burn-in, and number of adaptive sequences are
# much smaller than recommended to keep computation time low.
# recommended code:
## Not run:
M2 <- b2ML(list(Y[[1]]),actor= ~ Aat[1:10,], density = ~ Dat[1:10,], densVar = FALSE)
summary(M2)
## End(Not run)
**j2**  
*MCMC estimates for the j2 model*

Description


Usage

```r
j2(net, sender = NULL, receiver = NULL, density = NULL, reciprocity = NULL, 
   burnin = NULL, sample = NULL, adapt = NULL, center = NULL, seed = NULL)
```

Arguments

- **net**: Directed dichotomous n*n network (digraph).
- **sender**: Optional sender covariates of length n.
- **receiver**: Optional receiver covariates of length n.
- **density**: Optional density covariates of dimensions n*n.
- **reciprocity**: Optional symmetric reciprocity covariates of dimensions n*n.
- **burnin**: Optional specification of number of burn-in iterations (default is 10000).
- **sample**: Optional specification of number of MCMC samples (default is 40000).
- **adapt**: Optional number of adaptive sequences (default is 100).
- **center**: Optional boolean argument for centering predictors (default is TRUE).
- **seed**: Optional specification of random seed (default is 1).

Value

Returns a matrix with MCMC means, standard deviations, quantiles and effective sample sizes for j2 parameters.

Author(s)

Bonne J.H. Zijlstra <b.j.h.zijlstra@uva.nl>

References

Examples

# create a very small network with covariates for illustrative purposes
S <- c(1,0,1,0,1,0,1,0,1,0)
REC <- c(0,0,1,0,1,0,1,0,1,0)
D1 <- matrix(c(0,1,0,1,0,1,0,1,0,0,
              1,0,0,0,1,0,1,0,1,0,
              1,1,0,1,0,1,0,1,0,0,
              1,0,1,0,1,0,1,0,1,0,
              0,0,0,0,1,0,1,0,1,0,
              1,0,0,0,0,1,0,1,0,0,
              0,0,0,0,0,0,1,0,1,0,
              0,1,0,1,0,1,0,1,0,0,
              0,0,1,1,1,0,0,0,0,0), ncol=10)
D2 <- abs(matrix(rep(S,10), byrow = FALSE, ncol= 10) -
          matrix(rep(REC,10), byrow = TRUE, ncol= 10))
R <- D1*t(D1)
Y <- matrix(c(0,0,1,1,1,1,0,0,1,1,
              0,0,0,1,1,1,0,0,1,0,
              1,1,0,1,1,1,0,0,1,1,
              0,1,1,0,1,1,0,1,1,0,
              1,1,1,0,1,1,0,1,1,0,
              0,1,1,0,1,1,1,0,1,0,
              1,0,1,0,1,0,1,1,0,1,
              0,1,1,0,1,1,1,0,1,0,
              1,0,1,0,1,0,1,1,1,0,
              1,1,1,0,1,1,1,1,0,0), ncol=10)

# estimate j2 model
j2(Y,sender = ~ S, receiver = ~ REC, density = ~ D1 + D2, reciprocity= ~ R,
   burnin = 100, sample = 400, adapt = 10)
# notice: burn-in, sample size and number of adaptive sequenses are
# much smaller than recommended to keep computation time low.
# recommended code:
## Not run:
j2(Y,sender = ~ S, receiver = ~ REC, density = ~ D1 + D2, reciprocity= ~ R)
## End(Not run)

MCMC estimates for the p2 model

Description

Estimates p2 model parameters with the adaptive random walk algorithm as described in Zijlstra, Van Duijn and Snijders (2009) <doi: 10.1348/000711007X255336>. 
Usage

p2(net, sender = NULL, receiver = NULL, density = NULL, reciprocity = NULL,
    burnin = NULL, sample = NULL, adapt = NULL, seed = NULL)

Arguments

net Directed dichotomous n*n network (digraph).
sender Optional sender covariates of length n.
receiver Optional receiver covariates of length n.
density Optional density covariates of dimensions n*n.
reciprocity Optional symmetric reciprocity covariates of dimensions n*n.
burnin Optional specification of number of burn-in iterations (default is 10000).
sample Optional specification of number of MCMC samples (default is 40000).
adapt Optional number of adaptive sequences (default is 100).
seed Optional specification of random seed (default is 1).

Value

Returns a matrix with MCMC means, standard deviations, quantiles and estimated effective sample sizes for p2 parameters.

Author(s)

Bonne J.H. Zijlstra <b.j.h.zijlstra@uva.nl>

References


Examples

# create a very small network with covariates for illustrative purposes
S <- c(1,0,1,0,1,0,1,0,1,0)
REC <- (S*-1)+1
D1 <- matrix(c(0,1,0,1,0,1,0,1,0,1,
               0,0,1,0,1,0,1,0,1,0,
               1,1,0,1,0,0,0,0,0,1,
               1,1,1,0,1,0,0,0,0,1,
               1,0,1,0,1,1,0,1,0,0,
               0,0,0,0,0,1,1,1,1,1,
               0,0,0,0,1,0,1,0,1,1,
               1,0,0,0,1,1,1,1,1,1,
               0,1,0,1,0,1,0,0,0,0,
               1,0,1,1,1,0,0,0,0,0), ncol=10)
D2 <- abs(matrix(rep(S,10), byrow = FALSE, ncol= 10) -
p2ML

MCMC estimates for the (multilevel) p2 model

Description

Estimates the (multilevel) p2 model parameters, as described in Zijlstra, Van Duijn and Snijders (2006) <doi: 10.1027/1614-2241.2.1.42>.

Usage

p2ML(nets, sender = NULL, receiver = NULL, density =~ 1, reciprocity =~ 1, adapt = NULL, burnin = NULL, center = NULL, separate = NULL, seed = NULL)

Arguments

nets
  List with n dichotomous dependent directed networks.
sender
  Optional matrix with a stacked actor-level sender covariate, corresponding to the n networks. Multiple sender covariates can be added as a formula object, see example below
receiver
  Optional matrix with a stacked actor-level receiver covariate, corresponding to the n networks. Multiple receiver covariates can be added as a formula object
density
  Optional stacked matrix with a density covariate, with dimensions similar to the n dependent networks. Multiple density covariates can be added as a formula object, see example below
reciprocity  Optional stacked matrix with a symmetric reciprocity covariate, with dimensions similar to the n dependent networks. Multiple reciprocity covariates can be added as a formula object.

adapt  Optional number of adaptive sequences (default is 125).

burnin  Optional specification of number of burn-in iterations (default is 2500).

center  Optional argument for centering predictors (default is TRUE).

separate  Optional argument for estimating separate coefficients for the n dependent networks (default is FALSE).

seed  Optional specification of random seed (default is 1).

Value

Returns a fitted model of class 2ML, to be opened with the function summary().

Author(s)

Bonne J.H. Zijlstra <b.j.h.zijlstra@uva.nl>

References


Examples

# create two very small networks with covariates for illustrative purposes
Y1 <- matrix(c(0,1,0,1,0,1,0,1,0,0,
               0,1,1,0,1,0,1,0,1,1,
               1,1,0,1,0,0,0,0,1,0,
               1,1,0,1,0,0,0,0,0,1,
               1,0,1,0,1,1,0,1,1,0,
               0,0,0,0,0,1,1,1,1,0,
               0,0,0,0,0,0,1,0,1,0,
               1,0,0,0,1,1,0,1,1,0,
               0,1,0,1,0,1,0,1,0,0,
               0,0,1,1,0,0,0,0,0,0), ncol=10)

Y2 <- matrix(c(0,0,1,0,1,0,0,0,0,1,
               0,0,0,0,0,0,0,0,0,0,
               0,0,0,1,0,0,0,1,0,1,
               0,0,1,0,0,1,1,0,0,1,
               1,0,0,1,0,0,1,0,0,0,
               0,0,1,0,0,1,1,0,1,0,
               0,1,0,0,1,0,0,0,0,1,
               1,0,1,0,1,1,1,0,0,1,
               0,1,0,1,0,0,0,0,0,0,
               0,1,0,1,0,0,0,1,0,0), ncol=10)

Y <- list(Y1, Y2)
Sa1 <- c(1,0,1,0,1,0,1,0,1,0)
Sa2 <- c(1,0,1,0,0,1,1,0,1)
Sa <- list(Sa1, Sa2)
Sat <- Rat <- do.call(plyr::rbind.fill.matrix, Sa)
Sb1 <- c(0,1,1,0,1,0,1,0,1,0)
Sb2 <- c(1,0,1,0,0,1,0,1,0,1)
Sb <- list(Sb1, Sb2)
Sbt <- do.call(plyr::rbind.fill.matrix, Sb)
Da1 <- abs(matrix(rep(Sa1,10), byrow = FALSE, ncol= 10) -
            matrix(rep(Sa1,10), byrow = TRUE, ncol= 10))
Da2 <- abs(matrix(rep(Sa2,10), byrow = FALSE, ncol= 10) -
            matrix(rep(Sa2,10), byrow = TRUE, ncol= 10))
Da <- list(Da1, Da2)
Dat <- do.call(plyr::rbind.fill.matrix, Da)

# estimate p2 model for two networks
M1 <- p2ML(Y, sender= ~ Sat + Sbt, receiver= ~ Rat, density = ~ Dat, adapt = 10, burnin = 100)
summary(M1)
# Notice: burn-in, and number of adaptive sequences are
# much smaller than recommended to keep computation time low.
# recommended code:
## Not run:
M1 <- p2ML(Y, sender= ~ Sat + Sbt, receiver= ~ Rat, density = ~ Dat)
summary(M1)
## End(Not run)

# estimate p2 model for a single network
M2 <- p2ML(list(Y[[1]]),sender= ~ Sat[1:10,] + Sbt[1:10,], receiver= ~ Rat[1:10,],
            density = ~ Dat[1:10,], adapt = 10, burnin = 100)
summary(M2)
# Notice: burn-in, and number of adaptive sequences are
# much smaller than recommended to keep computation time low.
# recommended code:
## Not run:
M2 <- p2ML(list(Y[[1]]),sender= ~ Sat[1:10,] + Sbt[1:10,], receiver= ~ Rat[1:10,],
            density = ~ Dat[1:10,])
summary(M2)
## End(Not run)
Index

* b2 model
  b2ML, 3
  dyads-package, 2

* j2 model
  dyads-package, 2
  j2, 6

* p2 model
  dyads-package, 2
  p2, 7
  p2ML, 9

b2ML, 3

dyads (dyads-package), 2
dyads-package, 2

j2, 6

p2, 7
p2ML, 9