Package: NetworkReg (via r-universe)

September 6, 2024

Type Package

Title Regression Model on Network-Linked Data with Statistical Inference

Version 1.1

Date 2024-02-02

Description Linear regression model with nonparametric network effects on network-linked observations. The model is proposed by Le and Li (2022) <arXiv:2007.00803> and is assumed on observations that are connected by a network or similar relational data structure. The model does not assume that the relational data or network structure to be precisely observed; thus, the method is provably robust to a certain level of perturbation of the network structure. The package contains the estimation and inference function for the model.

License GPL (>= 2)

Imports Matrix, stats, randnet, RSpectra

NeedsCompilation no

Author Tianxi Li [aut, cre], Can M. Le [aut]

Maintainer Tianxi Li <tianxili@umn.edu>

Date/Publication 2024-02-08 21:00:02 UTC

Repository https://tianxili.r-universe.dev

RemoteUrl https://github.com/cran/NetworkReg

RemoteRef HEAD

RemoteSha 89a9b59e64caac0b6cade9d672263b780de481a1

Contents

net.gen.from.P — *generates a network from the given connection probability*

Description

Generates an adjacency matrix from a given probability matrix, according independent Bernoulli – the so-called inhomogeneous Erdos-Renyi model. It is used to generate new networks from a given model.

Usage

```
net.gen.from.P(P, mode = "undirected")
```

Arguments

`P`	connection probability between nodes
`mode`	"undirected" (default) if the network is undirected, so the adjacency matrix will be symmetric with only upper diagonal entries being generated as independent Bernoulli. Otherwise, the adjacency matrix gives independent Bernoulli everywhere.

Value

An adjacency matrix

Author(s)

Can M. Le and Tianxi Li.

Maintainer: Tianxi Li <tianxili@umn.edu>

NetworkReg — *Regression Model on Network-Linked Data with Statistical Inference*

Description

Linear regression model with nonparametric network effects on network-linked observations. The model is proposed by Le and Li (2022) <arXiv:2007.00803> on observations that are connected by a network or similar relational data structure. The model does not assume that the relational data or network structure to be precisely observed; thus, the method is provably robust to a certain level of perturbation of the network structure. The package contains the estimation and inference function for the model.

Details

Package:	NetworkReg
Type:	Package
Version:	1.1
Date:	2024-02-02
License:	GPL (>= 2)

Author(s)

Can M. Le and Tianxi Li.

Maintainer: Tianxi Li <tianxili@umn.edu>

References

Can M. Le and Tianxi Li. Le, C. M., & Li, T. (2022). Linear regression and its inference on noisy network-linked data. Journal of the Royal Statistical Society Series B: Statistical Methodology, 84(5), 1851-1885.

`SP.Inf` *Fitting Linear Regression Models on Network-Linked Data*

Description

SP.Inf is used to the regression model on network-linked data by subspace project and produce the inference result.

Usage

```
SP.Inf(X, Y, A, K, r = NULL, sigma2 = NULL, thr = NULL, alpha.CI = 0.05,
boot.thr = TRUE, boot.n = 50)
```

Arguments

`X`	the covariate matrix where each row is an observation and each column is a covariate. If an intercept is to be included in the model, the column of ones should be in the matrix.
`Y`	the column vector of response.
`A`	the network information. The most natural choice is the adjacency matrix of the network. However, if the network is assumed to be noisy and a better estimate of the structural connection strength, it can also be used. This corresponds to the Phat matrix in the original paper. A Laplacian matrix can also be used, but it should be flipped. See 'Details'.

`K`	the dimension of the network eigenspace for network effect.
`r`	the covariate-network cofounding space dimension. This is typically unknown and can be unspecified by using the default value 'NULL'. If so, the user should provide a threshold or resort to a tuning procedure by either the theoretical rule or a bootstrapping method, as described in the paper.
`sigma2`	the variance of random noise. Typically unknown.
`thr`	threshold for r estimation. If r is unspecified, we will use the thereshold to select r. If this is also 'NULL', aa theoretical threshold or a bootsrapping method can be evoked to estimate it.
`alpha.CI`	the 1-alpha.CI confidence level will be produced for the parameters.
`boot.thr`	logical. Only effective if both r and thr are NULLs. If FALSE, the theoretical threshold will be used to select r. Otherwise, the bootstrapping procedure will be used to find the threshold.
`boot.n`	the number of bootstrapping samples used when boot.thr is TRUE.

Details

The model fitting procedure is following the paper exactly, so please check the procedure and theory in the paper. If the Laplacian matrix L=D-A is the network quantity to use, notice that typically we treat the smallest values and their corresponding eigenvectors as network cohesive space. Therefore, one should consider flip the Laplacian matrix by using cI - L as the value for A, where c is sufficiently large to ensure PSD of cI-L.

Value

A list object with

`beta`	estimate of beta, the covariate effects
`alpha`	individual effects
`theta`	coefficients of confounding effects with respect to the covariates
`r`	confounding dimension
`sigma`	estimated random noise variance
`cov.hat`	covariance matrix of beta
`coef.mat`	beta and the confidence intervals according to alpha.CI and the p-values of the significance test
`fitted`	fitted value of response
`chisq.val`	the value of the chi-square statistic for the significance test for network effect
`chisq.p`	the p-value of the significance test for network effect

Author(s)

Can M. Le and Tianxi Li.

Maintainer: Tianxi Li <tianxili@umn.edu>

References

Le, C. M., & Li, T. (2022). Linear regression and its inference on noisy network-linked data. Journal of the Royal Statistical Society Series B: Statistical Methodology, 84(5), 1851-1885.

Examples

```
library(randnet)
library(RSpectra)
### data generating procedure in Section 5.3 of the paper

n <- 1000
big.model <- BlockModel.Gen(lambda=n^(1/2),n=n,beta=0.2,K=4)
P <- big.model$P
big.X <- cbind(rnorm(n),runif(n),rexp(n))

eigen.P <- eigs_sym(A=P,k=4)
X.true <- big.X
X.true <- scale(X.true,center=TRUE,scale=TRUE)*sqrt(n/(n-1))
X.true <- cbind(sqrt(n)*eigen.P$vectors[,1],X.true)
X.svd <- svd(X.true)
x.proj <- X.svd$v%*%(t(X.svd$u)/X.svd$d)
Theta <- X.svd$v%*%(t(X.svd$v)/(X.svd$d^2))*n
R <- X.svd$u
U <- eigen.P$vectors[,1:4]
true.SVD <- svd(t(R)%*%U,nu=4,nv=4)
V <- true.SVD$v
r <- 1
U.tilde <- U%*%V
R.tilde <- R%*%true.SVD$u
theta.tilde <- matrix(c(sqrt(n),0,0,0),ncol=1)
beta.tilde <- matrix(sqrt(n)*c(0,1,1,1),ncol=1)
Xtheta <- R.tilde%*%theta.tilde
Xbeta <- R.tilde%*%beta.tilde

theta <- solve(t(X.true)%*%X.true,t(X.true)%*%Xtheta)
beta <- solve(t(X.true)%*%X.true,t(X.true)%*%Xbeta)
alpha.coef <- matrix(sqrt(n)*c(0,1,1,1),ncol=1)
alpha <- U.tilde%*%alpha.coef

EY <- Xtheta+Xbeta + alpha


#### model fitting


A <- net.gen.from.P(P)
Khat <- BHMC.estimate(A, K.max = 15)$K  ### estimate K to use

## model fitting
Y <- EY + rnorm(n)
fit <- SP.Inf(X.true,Y,A,K=Khat,alpha=0.05,boot.thr=FALSE)
### In general, boot.thr = T works better for small sample but is slower.
```

```
### It was used in the paper.
fit$coef.mat
### notice that beta1 inference is meaningful here. Check the paper.
beta
fit$chisq.p

## find a parametric estimation of the network. This is generally not available.
rsc <- reg.SP(A,K=Khat,tau=0.1)
est <- SBM.estimate(A,rsc$cluster)
Phat <- est$Phat
fit2 <- SP.Inf(X.true,Y,Phat,K=Khat,alpha=0.05,boot.thr=FALSE)
fit2$coef.mat
### notice that beta1 inference is meaningful here. Check the paper.
```

Index