

# Package ‘NetworkReg’

July 21, 2025

**Type** Package

**Title** Generalized Linear Regression Models on Network-Linked Data with Statistical Inference

**Version** 2.0

**Date** 2024-10-10

**Description** Linear regression model and generalized linear models with nonparametric network effects on network-linked observations. The model is originally proposed by Le and Li (2022) <[doi:10.48550/arXiv.2007.00803](https://doi.org/10.48550/arXiv.2007.00803)> and is assumed on observations that are connected by a network or similar relational data structure. A more recent work by Wang, Le and Li (2024) <[doi:10.48550/arXiv.2410.01163](https://doi.org/10.48550/arXiv.2410.01163)> further extends the framework to generalized linear models. All these models are implemented in the current package. 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** stats, randnet, RSpectra

**NeedsCompilation** no

**Encoding** UTF-8

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

**Maintainer** Jianxiang Wang <[jw1881@scarletmail.rutgers.edu](mailto:jw1881@scarletmail.rutgers.edu)>

**RoxygenNote** 7.3.1

**Repository** CRAN

**Date/Publication** 2024-11-01 08:00:08 UTC

## Contents

NetworkReg-package . . . . .	2
net.gen.from.P . . . . .	2
SP.Inf . . . . .	3
<b>Index</b>	<b>6</b>

---

NetworkReg-package	<i>Generalized Linear Regression Models on Network-Linked Data with Statistical Inference</i>
--------------------	---

---

## 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:	2.0
Date:	2024-10-10
License:	GPL (>= 2)

## Author(s)

Jianxiang Wang, Can M. Le, and Tianxi Li.  
 Maintainer: Jianxiang Wang <jw1881@scarletmail.rutgers.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.

Wang J, Le C M, Li T. Perturbation-Robust Predictive Modeling of Social Effects by Network Subspace Generalized Linear Models. *arXiv preprint arXiv:2410.01163*, 2024.

---

net.gen.from.P	<i>generates a network from the given connection probability</i>
----------------	--

---

## 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

---

 SP.Inf

---

*Fitting Generalized Linear Model 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, model = "linear", 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.

**model** the type of Generalized Linear Regression. The "linear" ,"logistic" and "poisson" represents linear regression, logistic regression and poisson regression. The default is linear regression.

<code>r</code>	the covariate-network confounding 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.
<code>sigma2</code>	the variance of random noise for linear regression. Typically unknown.
<code>thr</code>	threshold for <code>r</code> estimation. If <code>r</code> is unspecified, we will use the threshold to select <code>r</code> . If this is also 'NULL', a theoretical threshold or a bootstrapping method can be evoked to estimate it.
<code>alpha.CI</code>	the 1- <code>alpha.CI</code> confidence level will be produced for the parameters.
<code>boot.thr</code>	logical. Only effective if both <code>r</code> and <code>thr</code> are NULLs. If FALSE, the theoretical threshold will be used to select <code>r</code> . Otherwise, the bootstrapping procedure will be used to find the threshold.
<code>boot.n</code>	the number of bootstrapping samples used when <code>boot.thr</code> 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

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

### Author(s)

Jianxiang Wang, Can M. Le, and Tianxi Li. Maintainer: Jianxiang Wang <jw1881@scarletmail.rutgers.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.

Wang J, Le C M, Li T. Perturbation-Robust Predictive Modeling of Social Effects by Network Subspace Generalized Linear Models. *arXiv preprint arXiv:2410.01163*, 2024.

## Examples

```
set.seed(1)
library(randnet)
library(RSpectra)
### Example data generation procedure from Section 5.3 of the paper with logistic regression
n <- 1000
big.model <- BlockModel.Gen(lambda=n^(2/3), n=n, beta=0.2, K=4)
P <- big.model$P
big.X <- cbind(rnorm(n), runif(n), rexp(n))
eigen.P <- eigs_sym(A=P, k=4)
X.true <- big.X
X.true <- scale(X.true, center=TRUE, scale=TRUE) * sqrt(n/(n-1))
beta <- matrix(c(1,1,1), ncol=1)
Xbeta <- X.true %*% beta
U <- eigen.P$vectors[,1:4]
alpha.coef <- matrix(sqrt(n) * c(1, 1, 1, 1), ncol=1)
alpha <- U %*% alpha.coef
EY <- (1 + exp(-Xbeta - alpha))^(1)
## Model fitting
A <- net.gen.from.P(P)
Y <- rbinom(n, 1, EY)
fit <- SP.Inf(X.true, Y, A, K=4, model=c("logistic"), alpha=0.05, boot.thr=FALSE)
fit$coef.mat
```

# Index

- \* **models**

- SP.Inf, [3](#)

- \* **package**

- NetworkReg-package, [2](#)

- \* **regression**

- SP.Inf, [3](#)

net.gen.from.P, [2](#)

NetworkReg-package, [2](#)

SP.Inf, [3](#)