Package: netcoh (via r-universe)

November 15, 2026

Type Package

Title Statistical Modeling with Network Cohesion

Version 0.2.9001

Maintainer Tianxi Li <tianxili@umich.edu>

Description Model fitting procedures for regression with network cohesion effects, when a network connecting sample individuals is available in a regression problem. In the future, other commonly used statistical models will be added, such as gaussian graphical model. Compatibility updates by Renova.

License GPL (>= 2)

Depends methods,Matrix

Imports Rcpp (>= 0.11.5)

LinkingTo Rcpp,RcppArmadillo

NeedsCompilation yes

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

Date/Publication 2026-11-15 20:40:48 UTC

RemoteUrl https://github.com/renova-r/netcoh

RemoteRef renova-0.2.9001

RemoteSha 2c9319229a522f700146fdbdcb0a8c4b2449bf52

Contents

netcoh-package 2
predict.rncReg 2
rncreg 4
rncReg-class 7
rncregpath 8

Index **11**

netcoh-package *Statistical models with network cohesion.*

Description

The package provides the methods to build a class of statistical models on networks with network cohesion effects being taken into consideration.

Details

Package:	netcoh
Type:	Package
Version:	0.2
Date:	2016-04-29
License:	GPL (>= 2)

The major function `rncreg` accepts `A` network adjacency matrix, `X` design matrix and response `Y` for regression models and then produces the fitted modelfor the tuning parameter `lambda`.

Another function `predict.rncReg` accepts the model fitted by `rncreg` and the full design matrix as well as the full adjacency matrix (with both training and test data).

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu

Maintainer: Tianxi Li <tianxili@umich.edu>

References

Tianxi Li, Elizaveta Levina and Ji Zhu. (2016) *Regression with network cohesion*, http://arxiv.org/pdf/1602.01192v1.pdf

predict.rncReg *make predictions from a "rncReg" object.*

Description

The generic prediction function that makes predictions according an "rncReg" object, which is a regression model fitted with network cohesion.

Usage

```
## S3 method for class 'rncReg'
predict(object,full.X=NULL,full.A,...)
```

Arguments

`object`	An object returned by the function `rncreg`.
`full.X`	A matrix with first n rows being the original training covariates and the last m rows being the new test covariates. If not provided, covariates will not be used in prediction.
`full.A`	An adjacency matrix of the complete network with both training and test samples. The first n vertices are representing the training samples (in the original order) and the last m vertices are reprenting the test samples.
`...`	further arguments passed to or from other methods.

Details

The function first predicts individual effects on test sample by minimizing network cohesion penalty and then incorporates the covariate effects. For full details, please check the reference paper. The predicted test sample individual effects will be returned, as well as the corresponding linear terms. For linear regression model, the predicted response y is also given, which is exactly the same as the linear term. For logistic regression, the predicted probability is also given.

Value

A list with following slots:

`terms`	the linear term in the model.
`alpha`	the predicted individual effects.
`y`	the predicted responses in linear model.
`p`	the predicted probabilities in logistic regression.
`model`	the model used in prediction. This is the same as in `object`.

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <tianxili@umich.edu>

References

Tianxi Li, Elizaveta Levina and Ji Zhu. (2016) *Regression with network cohesion*, http://arxiv.org/pdf/1602.01192v1.pdf

See Also

`rncreg`

Examples

```
set.seed(100)

A <- matrix(0,200,200)
A[1:100,1:100] <- 1
A[101:200,101:200] <- 1
diag(A) <- 0

alpha <- c(rep(1,100),rep(-1,100)) + rnorm(200)*0.5
A <- A[c(1:50,101:150,51:100,151:200),c(1:50,101:150,51:100,151:200)]
alpha <- alpha[c(1:50,101:150,51:100,151:200)]

beta <- rnorm(2)

X <- matrix(rnorm(400),ncol=2)

Y <- X%*%beta + alpha + rnorm(200)

A1 <- A[1:100,1:100]
X1 <- X[1:100,]
Y1 <- matrix(Y[1:100],ncol=1)


## If one wants to regularize the Laplacian by
## using gamma > 0 in rncreg, we suggest use
## centered data.
#mean.x <- colMeans(X1)
#mean.y <- mean(Y1)
#Y1 <- Y1-mean.y
#X1 <- t(t(X1)-mean.x)
#Y <- Y-mean.y
#X <- t(t(X)-mean.x)



m <- rncreg(A=A1,X=X1,Y=Y1,model="linear",lambda=10,gamma=0,cv=5)

p <- predict(m,full.A=A,full.X=X)
```

`rncreg` *Fits a regression model with network cohesion effects.*

Description

fits a regression model such that each samples are following a different regression curve such that connected individuals in a network tend to have similar curves. The function currently fits linear, logistic and Cox's regression model.

Usage

```
rncreg(A,lambda,Y=NULL,X=NULL,dt=NULL,gamma=0.05,
     model=c("linear","logistic","cox"), max.iter=50,tol=1e-4,
     init=NULL, cv=NULL,cv.seed=999,low.dim=NULL,verbose=FALSE)
```

Arguments

A	An nxn symmetric adjacency matrix for the network.
lambda	Tuning parameter for the cohesion penalty.
Y	An nx1 matrix of response, if the model to fit is linear or logistic. It will not be used if one fits Cox's model.
X	An nxp covariate matrix with each row being the covariates for one individual. If one want to fits a model without using any covariate, it can be empty.
dt	Only used to fit Cox's model. An nx2 data.frame such that the first column is the observed time while the second column is the event indicator which is 1 for truely observed events and 0 for censored events.
gamma	The amount of diagonal regularization added to graph Laplacian.
model	Can only be one of "linear", "logistic" or "cox".
max.iter	The maximum number of newton steps to iterate. Only used for logistic model or Cox's model.
tol	The tolerance level for convergence. Only used for logistic model or Cox's model.
init	The initial point for newton algoritm. It should be an (n+p)x1 matrix that stacks alpha and beta. Only used for logistic model or Cox's model. If not specified, zeros will be used.
cv	Number of folds for cross-validation. If unspecified, then no cross-validation will be done.
cv.seed	Random number generator seed for cross-validation.
low.dim	Only used for linear model. If the probelm is a low dimensional problem such that n»p, then using low.dim=TRUE is potentially faster.
verbose	If TRUE, the log likelihood in each newton step will be printed. Only used for logistic model and Cox's model.

Details

The function solves

$$maxL(\alpha, \beta) - \lambda \alpha^T L \alpha.$$

With a proper choice of L function according to the specific model. When the model is linear regression, L is the negative squared error (or gaussian kernel); when the model is logistic regression, L is the binomial log likelihood; when the model is Cox's model, L is the log partial likelihood. gamma is used to regularize the graph Laplacian and is potentially helpful for numerical stability and Cox's model identifiability. Notice that having a positive gamma tends to shrink individual effects to zeros. Thus in linear regression, we suggest first center the data (both predictors and response) before fitting the model. For full details, please check the reference paper.

Value

An object from class rncReg will be returned.

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <tianxili@umich.edu>

References

Tianxi Li, Elizaveta Levina and Ji Zhu. (2016) *Regression with network cohesion*, http://arxiv.org/pdf/1602.01192v1.pdf

See Also

rncReg,predict.rncReg

Examples

```
set.seed(100)

A <- matrix(0,200,200)
A[1:100,1:100] <- 1
A[101:200,101:200] <- 1
diag(A) <- 0

alpha <- c(rep(1,100),rep(-1,100)) + rnorm(200)*0.5
A <- A[c(1:50,101:150,51:100,151:200),c(1:50,101:150,51:100,151:200)]
alpha <- alpha[c(1:50,101:150,51:100,151:200)]

beta <- rnorm(2)

X <- matrix(rnorm(400),ncol=2)

Y <- X%*%matrix(beta,ncol=1) + alpha + rnorm(200)*0.5

delta <- Y
delta[Y>0] <- 1
delta[Y<=0] <- 0

A1 <- A[1:100,1:100]
X1 <- X[1:100,]
Y1 <- matrix(Y[1:100],ncol=1)
delta1 <- matrix(delta[1:100],ncol=1)


## If one wants to regularize the Laplacian
## by using gamma > 0 in rncreg,
## we suggest use centered data.
#mean.x <- colMeans(X1)
#mean.y <- mean(Y1)
#Y1 <- Y1-mean.y
```

```
#X1 <- t(t(X1)-mean.x)
#Y <- Y-mean.y
#X <- t(t(X)-mean.x)

m <- rncreg(A=A1,X=X1,Y=Y1,model="linear",lambda=10,gamma=0,cv=5)
p <- predict(m,full.A=A,full.X=X)

#m <- rncreg(A=A1,X=X1,Y=Y1,model="logistic",lambda=10,gamma=0.01,cv=5)

#m <- rncreg(A=A1,X=X1,dt=data.frame(y=Y1,delta=delta1),model="cox",lambda=10,gamma=0.01,cv=5)
```

`rncReg-class` *Class* `"rncReg"`

Description

A generic class for regression objects with network cohesion.

Objects from the Class

Objects can be created by calls of the form `new("rncReg", ...)`.

Slots

`alpha`: The individual effects of the regression.

`beta`: The fixed effects or covariate coefficients of the regression.

`A`: The network adjacency matrix for which cohession is assumed.

`lambda`: Parameter for cohesion penalty.

`X`: Covariate matrix.

`Y`: Response matrix.

`dt`: The response data frame with the first column being the observed time and the second column being the event indicator.

`gamma`: Regularization parameter for graph Laplacian.

`cv`: Number of folds in cross-validation.

`cv.loss`: Cross-validated prediciton loss. It is MSE for linear regression, binomial deviance for logistic regression and test partial loglikelihood for Cox's model (see reference paper).

`cv.sd`: Standard deviation of cross-validation loss. It can be used for cross-validation by 1 sigma rule. It is more robust to noises.

`model`: The specific regression model used.

Methods

No methods defined with class "rncReg" in the signature.

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <tianxili@umich.edu>

References

Tianxi Li, Elizaveta Levina and Ji Zhu. (2016) *Regression with network cohesion*, http://arxiv.org/pdf/1602.01192v1.pdf
Verweij, Pierre JM, and Hans C. Van Houwelingen. (1993) *Cross-validation in survival analysis*, Statistics in medicine 12, no. 24: 2305-2314.

See Also

rncreg

---

rncregpath *Fits regression models with network cohesion effects for a sequence of cohesion levels.*

Description

Fits models by rncreg for a sequence of lambda values.

Usage

```
rncregpath(A,lambdaseq,Y=NULL,X=NULL,dt=NULL,gamma=0.05,
     model=c("linear","logistic","cox"), max.iter=50,tol=1e-4,
     init=NULL, cv=NULL,cv.seed=999,low.dim=NULL,verbose=FALSE)
```

Arguments

A	An nxn symmetric adjacency matrix for the network.
lambdaseq	A vector of lambda values.
Y	An nx1 matrix of response, if the model to fit is linear or logistic. It will not be used if one fits Cox's model.
X	An nxp covariate matrix with each row being the covariates for one individual. If one want to fits a model without using any covariate, it can be empty.
dt	Only used to fit Cox's model. An nx2 data.frame such that the first column is the observed time while the second column is the event indicator which is 1 for truely observed events and 0 for censored events.
gamma	The amount of diagonal regularization added to graph Laplacian.
model	Can only be one of "linear", "logistic" or "cox".
max.iter	The maximum number of newton steps to iterate. Only used for logistic model or Cox's model.

tol	The tolerance level for convergence. Only used for logistic model or Cox's model.
init	The initial point for newton algoritm. It should be an (n+p)x1 matrix that stacks alpha and beta. Only used for logistic model or Cox's model. If not specified, zeros will be used.
cv	Number of folds for cross-validation. If unspecified, then no cross-validation will be done.
cv.seed	Random number generator seed for cross-validation.
low.dim	Only used for linear model. If the probelm is a low dimensional problem such that n»p, then using low.dim=TRUE is potentially faster.
verbose	If TRUE, the log likelihood in each newton step will be printed. Only used for logistic model and Cox's model.

Details

The function repeatly calls rncreg for the values of lambda.

Value

A list will be returned with the following slots:

models	A list of objects from rncReg class, each of which corresponds to a specific lambda value in lambdaseq.
cv.seq	The sequence of cross-validation loss corresponding to lambdaseq.
cv.sd	The sequence of corssivalidation standard deviation
cv.min.index	The index of model that corresponds to the one with miminum cv loss.
cv.1sd.index	The index of model that corresponds to the one selected by 1 sigma rule.

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <tianxili@umich.edu>

References

Tianxi Li, Elizaveta Levina and Ji Zhu. (2016) *Regression with network cohesion*, http://arxiv.org/pdf/1602.01192v1.pdf

See Also

rncReg,predict.rncReg

Examples

```
set.seed(100)

A <- matrix(0,200,200)
A[1:100,1:100] <- 1
A[101:200,101:200] <- 1
diag(A) <- 0

alpha <- c(rep(1,100),rep(-1,100)) + rnorm(200)*0.5
A <- A[c(1:50,101:150,51:100,151:200),c(1:50,101:150,51:100,151:200)]
alpha <- alpha[c(1:50,101:150,51:100,151:200)]

beta <- rnorm(2)

X <- matrix(rnorm(400),ncol=2)

Y <- X%*%beta + alpha + rnorm(200)

A1 <- A[1:100,1:100]
X1 <- X[1:100,]
Y1 <- matrix(Y[1:100],ncol=1)


## If one wants to regularize the Laplacian
## by using gamma > 0 in rncreg,
## we suggest use centered data.
#mean.x <- colMeans(X1)
#mean.y <- mean(Y1)
#Y1 <- Y1-mean.y
#X1 <- t(t(X1)-mean.x)
#Y <- Y-mean.y
#X <- t(t(X)-mean.x)



m.list <- rncregpath(A=A1,X=X1,Y=Y1,model="linear",
                     lambdaseq=exp(seq(0,log(200),length.out=20)),gamma=0,cv=5)
m.list$cv.seq
```

Index

∗ **classes**
    rncReg-class, 7
∗ **models**
    netcoh-package, 2
    predict.rncReg, 2
    rncreg, 4
    rncregpath, 8
∗ **networks**
    predict.rncReg, 2
    rncreg, 4
    rncregpath, 8
∗ **network**
    netcoh-package, 2
∗ **package**
    netcoh-package, 2
∗ **regression**
    netcoh-package, 2
    predict.rncReg, 2
    rncreg, 4
    rncregpath, 8

netcoh *(*netcoh-package*)*, 2
netcoh-package, 2

predict.rncReg, 2

rncreg, 4
rncReg-class, 7
rncregpath, 8