Package ‘netReg’

April 12, 2018

Type Package

Title Network-Regularized Regression Models

Version 1.2.0

Date 2017-10-23

Maintainer Simon Dirmeier <simon.dirmeier@web.de>

Description netReg fits linear regression models using network-penalization. Graph prior knowledge, in the form of biological networks, is being incorporated into the likelihood of the linear model. The networks describe biological relationships such as co-regulation or dependency of the same transcription factors/metabolites/etc. yielding a part sparse and part smooth solution for coefficient profiles.

URL https://github.com/dirmeier/netReg

BugReports https://github.com/dirmeier/netReg/issues

Depends R(>= 3.4)

biocViews Software, StatisticalMethod, Regression, FeatureExtraction, Network, GraphAndNetwork

License GPL-3

Encoding UTF-8

Suggests testthat, knitr, rmarkdown, BiocStyle, lintr, lassoshooting

VignetteBuilder knitr

RoxygenNote 6.0.1

SystemRequirements C++11

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, stats

NeedsCompilation yes

Author Simon Dirmeier [aut, cre]

R topics documented:

`netReg-package` *netReg*

Description

netReg is a package for generalized linear regression that includes prior graphs in the models objective function.

Details

netReg uses *Armadillo*, *OpenBLAS*, *BLAS* and *LAPACK* for fast matrix computations and *Dlib* for constrained derivate-free optimization.

Author(s)

Simon Dirmeier | `<simon.dirmeier@gmx.de>`

References

Friedman J., Hastie T., Hoefling H. and Tibshirani R. (2007), Pathwise coordinate optimization.
The Annals of Applied Statistics

Friedman J., Hastie T. and Tibshirani R. (2010), Regularization Paths for Generalized Linear Models via Coordinate Descent.
Journal of Statistical Software

Fu W. J. (1998), Penalized Regression: The Bridge Versus the Lasso.
Journal of Computational and Graphical Statistics

Cheng W. and Wang W. (2014), Graph-regularized dual Lasso for robust eQTL mapping.
Bioinformatics

Powell M.J.D. (2009), The BOBYQA algorithm for bound constrained optimization without derivatives.
http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf

`cv.edgenet` *Find the optimal shrinkage parameters for edgenet*

Description

Finds the optimal shrinkage parameters using cross-validation for edgenet. We use the BOBYQA algorithm to minimize the sum of squared residuals objective function.

Usage

```
cv.edgenet(X, Y, G.X = NULL, G.Y = NULL, thresh = 1e-05, maxit = 1e+05,
  family = c("gaussian"), epsilon = 0.001, approx.maxit = 10000,
  nfolds = 10, ...)
```

Arguments

X	input matrix, of dimension (n x p) where n is the number of observations and p is the number of covariables. Each row is an observation vector.
Y	output matrix, of dimension (n x q) where n is the number of observations and q is the number of response variables Each row is an observation vector.
G.X	non-negativ affinity matrix for n, of dimensions (p x p) where p is the number of covariables X
G.Y	non-negativ affinity matrix for n, of dimensions (q x q) where q is the number of covariables Y
thresh	threshold for coordinate descent
maxit	maximum number of iterations
family	family of response, e.g. gaussian
epsilon	the threshold criterion for BOBYQA to stop. Usually 1e-3 is a good choice.
approx.maxit	the maximum number of iterations for BOBYQA (if choosen). Usually 1e4 is a good choice.
nfolds	the number of folds to be used - default is 10 (minimum 3, maximum nrow(X)).
...	additional parameters

Value

An object of class cv.edgenet

call	the call that produced the object
lambda	the estimated (p x q)-dimensional coefficient matrix B.hat
psigx	the estimated (q x 1)-dimensional vector of intercepts
psigy	the estimated (q x 1)-dimensional vector of intercepts

Author(s)

Simon Dirmeier, <mail@simon-dirmeier.net>

References

Friedman J., Hastie T., Hoefling H. and Tibshirani R. (2007), Pathwise coordinate optimization.
The Annals of Applied Statistics

Friedman J., Hastie T. and Tibshirani R. (2010), Regularization Paths for Generalized Linear Models via Coordinate Descent.
Journal of Statistical Software

Fu W. J. (1998), Penalized Regression: The Bridge Versus the Lasso.
Journal of Computational and Graphical Statistics

Cheng W. and Wang W. (2014), Graph-regularized dual Lasso for robust eQTL mapping.
Bioinformatics

Powell M.J.D. (2009), The BOBYQA algorithm for bound constrained optimization without derivatives.
http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf

Examples

```
X <- matrix(rnorm(100*10), 100, 10)
b <- rnorm(10)
G.X <- matrix(rpois(10*10,1),10)
G.X <- t(G.X) + G.X
diag(G.X) <- 0

# fit a Gaussian model
Y <- X%*%b + rnorm(100)
cv.edge <- cv.edgenet(X=X, Y=Y, G.X=G.X, family="gaussian")
```

---

edgenet — *Fit a graph-regularized linear regression model using edge-based regularization.*

Description

Fit a graph-regularized linear regression model using edge-penalization. The coefficients are computed using graph-prior knowledge in the form of one/two affinity matrices. Graph-regularization is an extension to previously introduced regularization techniques, such as the LASSO.

Usage

```
edgenet(X, Y, G.X = NULL, G.Y = NULL, lambda = 1, psigx = 1,
  psigy = 1, thresh = 1e-05, maxit = 1e+05, family = c("gaussian"), ...)
```

Arguments

`X`	input matrix, of dimension (`n` x `p`) where `n` is the number of observations and `p` is the number of covariables. Each row is an observation vector.
`Y`	output matrix, of dimension (`n` x `q`) where `n` is the number of observations and `q` is the number of response variables Each row is an observation vector.
`G.X`	non-negativ affinity matrix for `n`, of dimensions (`p` x `p`) where `p` is the number of covariables `X`
`G.Y`	non-negativ affinity matrix for `n`, of dimensions (`q` x `q`) where `q` is the number of covariables `Y`
`lambda`	shrinkage parameter for LASSO.
`psigx`	shrinkage parameter for graph-regularization of `G.X`
`psigy`	shrinkage parameter for graph-regularization of `G.Y`
`thresh`	threshold for coordinate descent
`maxit`	maximum number of iterations
`family`	family of response, e.g. gaussian
`...`	additional params

Value

An object of class edgenet

coefficients	the estimated (p x q)-dimensional coefficient matrix B.hat
intercept	the estimated (q x 1)-dimensional vector of intercepts
call	the call that produced the object
family	the family of the response

Author(s)

Simon Dirmeier | <mail@simon-dirmeier.net>

References

Friedman J., Hastie T., Hoefling H. and Tibshirani R. (2007), Pathwise coordinate optimization.
The Annals of Applied Statistics

Friedman J., Hastie T. and Tibshirani R. (2010), Regularization Paths for Generalized Linear Models via Coordinate Descent.
Journal of Statistical Software

Fu W. J. (1998), Penalized Regression: The Bridge Versus the Lasso.
Journal of Computational and Graphical Statistics

Cheng W. and Wang W. (2014), Graph-regularized dual Lasso for robust eQTL mapping.
Bioinformatics

Examples

```
X <- matrix(rnorm(100*10), 100, 10)
b <- rnorm(10)
G.X <- matrix(rpois(100,1), 10)
G.X <- t(G.X) + G.X
diag(G.X) <- 0

# fit a Gaussian model
Y <- X%*%b + rnorm(100)
fit <- edgenet(X=X, Y=Y, G.X=G.X, family="gaussian")
```

---

predict.gaussian.edgenet

Predict method for gaussian edgenet fits

Description

Predicts the estimated Y.hat values for a newdata design matrix X similar to the other predict methods, e.g. from glm and glmnet

Usage

```
## S3 method for class 'gaussian.edgenet'
predict(object, newdata = NULL, ...)
```

Arguments

`object`	a fitted object of class *gaussian.edgenet*
`newdata`	a new (`m` x `p`)-dimensional design matrix with a variable number of observations `m`, but a constant number of co-variables `p`
`...`	further arguments

Value

A (`m` x `q`)-dimensional matrix

Examples

```
## Not run:
X <- matrix(rnorm(100*10),100,10)
G.X <- matrix(rpois(10*10,1),10)
G.X <- t(G.X) + G.X
diag(G.X) <- 0

Y <- matrix(rnorm(100*10),100,10)
fit <- edgenet(X=X, Y=Y, G.X=G.X, family="gaussian")
pred <- predict(fit, X)

## End(Not run)
```

`yeast` *A sample yeast data set for regression*

Description

The yeast data set is a `list` containing three matrices that can be used as an example for using `netReg`.

Usage

```
data(yeast)
```

Format

A `list` containing three matrices

Details

- `X` (112 x 500)-dimensional binary matrix of 500 genetic markers for 112 yeast samples
- `Y` (112 x 231)-dimensional double matrix of 231 gene expression values for 112 yeast samples
- `GY` (231 x 231)-dimensional adjaceny matrix representing protein-protein interactions for 231 yeast genes

Index