Package ‘qut’

January 19, 2021

Type Package

Title Quantile Universal Threshold

Version 2.2

Description Thresholding based tests for null hypothesis of the form A beta =c, and the Quantile Universal Threshold (QUT) for lasso regularization of Generalized Linear Models (GLM) and square-root lasso to obtain a sparse model with a good compromise between high true positive rate and low false discovery rate. Giacobino et al. (2017) <doi:10.1214/17-EJS1366>. Sardy et al. (2017) <arXiv:1708.02908>.

License GPL-2

Depends Matrix, glmnet, lars, flare

NeedsCompilation no

Author Jairo Diaz-Rodriguez [aut, cre, cph],
Sylvain Sardy [aut, ths],
Caroline Giacobino [aut],
Nick Hengartner [aut]

Maintainer Jairo Diaz-Rodriguez <adjairo@uninorte.edu.co>

Repository CRAN

Date/Publication 2021-01-19 10:00:02 UTC

R topics documented:

QUT-package *Quantile Universal Threshold*

Description

Selection of a threshold parameter λ for GLM-lasso and Square-root lasso. The method consists in considering a null model, finding the theoretical distribution of the threshold parameter under the null, and setting λ to an upper quantile of that distribution. Although this strategy does not use the data to select λ but simply considers the behavior under the null model, it provides a theoretically and computationally sound selection.

Details

Package:	QUT
Type:	Package
License:	GPL-2

Author(s)

Jairo Diaz Rodriguez <jairo.diaz@unige.ch>

References

C. Giacobino, J. Diaz Rodriguez, S. Sardy, N. Hengartner. Quantile universal threshold for model selection. 2016

affinelassotest *Affine lasso test*

Description

Perform thresholding tests for null hypothesis of the form H0: A beta = c.

Usage

```
affinelassotest(y, Xdata, family = gaussian, alpha, beta0=NA, cc = NA, lambdas = NA,
outrescale = NA, intercept = TRUE, group.sizes = rep(1, ncol(X)), A = ncol(X),
LAD = FALSE, composite = TRUE, M = round(min(10000, max(1000, 1e+09/nrow(X)/ncol(X)))))
```

Arguments

`y` response variable. Quantitative for family=`gaussian`, or family=`poisson` (non-negative counts). For family=`binomial` should be a factor with two levels.

`Xdata` input matrix, of dimension n x p; each row is an observation vector.

`family` response type (see above). Default is `gaussian`.

`alpha` desired level of the test.

`beta0` if known, value of the nuissance parameter. Otherwise it takes beta0=0, and the statistic is asymptotically a pivot for non-Gaussian.

`cc` vector c

`lambdas` if not provided, the code performs M Monte Carlo simulation to obtain the empirical distribution Λ and the corresponding value in the α-quantile for testing. Otherwise, vector with all the Monte Carlo values of lambda. Default is NA.

`outrescale` object containing all variables corresponding to the rescaling and test options. If not provided, this is calculated authomatically with function `processX`. Default is NA.

`intercept` should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).

`group.sizes` the vector of group sizes for affine group lasso. The number of elements is L and sum(group.sizes) should be equal to P. If L==P, then the lasso test is employed, otherwise group lasso. Default is no groups, so `rep(1,ncol(X))`.

`A` if A is a matrix it tests A beta = c. If A is a vector, then it gives the indexes of the parameters to be tested. Used if family=`gaussian`. Default is to test beta=0, so A=ncol(X).

`LAD` set TRUE if LAD lasso test. Default is FALSE

`composite` set TRUE if composite test (O & +). Default is TRUE

`M` number of Monte Carlo Simulations to estimate the distribution Λ.

Value

`lambda.alpha` value of λ in the α-quantile

`lambda.data` value of λ for the current data.

`rejectH0` result of the test. TRUE if H0 is rejected.

`lambdas` values of λ of the Monte Carlo simulation under the null hypothesis.

`outrescale` object containing all the rescaling variables.

`pvalue` approximate p-value for current data.

Author(s)

Sylvain Sardy and Jairo Diaz Rodriguez

References

Thresholding tests. Sylvain Sardy, Caroline Giacobino, Jairo Diaz.

Examples

```
# Test H0:beta=0
P=200
N=20
s=1
A=P
alpha=0.05
X=matrix(rnorm(N*P),N,P)
M=100 #Leave the default or select higher value for better level.

#when H0 is not rejected
beta_scal=0
beta=c(rep(beta_scal, s), rep(0, P-s))
y=X%*%beta+rnorm(N)
out=affinelassotest(y,X,gaussian,alpha,M=M)
print(out$rejectH0)

#when H0 is rejected
beta_scal=10
beta=c(rep(beta_scal, s), rep(0, P-s))
y=X%*%beta+rnorm(N)
out=affinelassotest(y,X,gaussian,alpha,M=M)
print(out$rejectH0)
```

---

chemometrics *Chemometrics data set from Sardy (2008)*

Description

Fuel octane level measurements with sample size N = 434 and P = 351 spectrometer measurements.

Usage

```
data(chemometrics)
```

Format

A data frame with 434 observations on the following 2 variables.

- `y` a numeric vector
- `x` a matrix with 351 columns

References

S. Sardy. On the practice of rescaling covariates. International Statistical Review. 2008

Examples

```
data(chemometrics)
```

internetAd *InternetAd data set from Kushmerick (1999)*

Description

Classification of N = 2359 possible advertisements on internet pages based on P = 1430 features

Usage

```
data(internetAd)
```

Format

A data frame with 2359 observations on the following 2 variables.

- `y` a numeric vector
- `x` a matrix with 1430 columns

References

N. Kushmerick. Learning to remove internet advertisements. In Proceedings of the Third Annual Conference on Autonomous Agents. 1999

Examples

```
data(internetAd)
```

lambdaqut *Quantile Universal Threshold, regularization parameter for GLM-lasso*

Description

Computes the Quantile Universal Threshold for GLM-lasso.

Usage

```
lambdaqut(y, X, family = gaussian, alpha.level = 0.05, M = 1000,
qut.standardize = TRUE, intercept = TRUE, no.penalty = NULL, offset = NULL,
bootstrap=TRUE,beta0=NA,method='lasso',fixbeta0=FALSE)
```

Arguments

`y` response variable. Quantitative for family=`gaussian`, or family=`poisson` (non-negative counts). For family=`binomial` should be a factor with two levels.

`X` input matrix, of dimension n x p; each row is an observation vector.

`family` response type (see above). Default is `gaussian`.

`alpha.level` level, such that quantile $\tau = (1-\texttt{alpha.level})/\gamma$. Default is 0.05.

`M` number of Monte Carlo Simulations to estimate the distribution Λ. Default is 1000.

`qut.standardize` standardize matrix X with a quantile-based standardization. Default is TRUE.

`intercept` should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).

`no.penalty` unpenalized subset of covariates.

`offset` a vector of length `n` that is included in the linear predictor. Useful for the "poisson" family (e.g. log of exposure time), or for refining a model by starting at a current fit. Default is NULL.

`bootstrap` set TRUE if it is desired to bootstrap matrix X when computing the Quantile Universal Threshold (Random scenario). Default is TRUE.

`beta0` coefficients of the unpenalized covariates for generating the null data for the Quantile Universal Threshold. By default is `NA` and it is estimated using the unpenalized covariates and/or the intercept if TRUE. If it is desired to set `beta0` in advance, then it should be a vector of size the number of unpenalized covariates including the intercept if `intercept=TRUE`, in the same order. If there are not unpenalized covariates and `intercept=TRUE`, then it must be a real number.

`method` objective function for the zero thresholding. Select `lasso` for GLM-lasso or `sqrtlasso` for Square-root lasso.

`fixbeta0` used when `beta0` is numeric. When TRUE, it does not estimate `beta0` for each monte carlo simulation.

Value

`lambda` value of the Quantile Universal Threshold.

`Xnew` standardized matrix X; `Xnew=X`×`scale.factor`.

`scale.factor` scale factor for `Xnew`.

`lambda.max` smallest lambda that sets the lasso estimates to the zero vector.

`beta0` estimated value of the intercept when family is not `gaussian`.

Author(s)

Jairo Diaz

References

C. Giacobino, J. Diaz, S. Sardy, N. Hengartner. Quantile universal threshold for model selection. 2016 Jianqing Fan, Shaojun Guo and Ning Hao. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. Journal of the Royal Statistical Society: Series B. 2012 Stephen Reid, Robert Tibshirani, and Jerome Friedman. A Study of Error Variance Estimation in Lasso Regression. 2013

See Also

`qut`

Examples

```
X=matrix(rnorm(20*200),20,200)
y=rnorm(20)+1
lambda=lambdaqut(y,X,family=gaussian)
```

`predict.qut` *Make predictions from a "qut" object.*

Description

Similar to other predict methods, this function predicts fitted values from a fitted "qut" object

Usage

```
## S3 method for class 'qut'
predict(object, newx, mode = "glm", offset = NULL,...)
## S3 method for class 'qut'
coef(object, mode = "glm",...)
```

Arguments

`object`	fitted "qut" model object.
`newx`	matrix of new values for X at which predictions are to be made. Must be a matrix.
`mode`	make predictions with lasso coefficients (type=`lasso`) or with fitted glm coefficients (type=`glm`). Default is `glm`.
`offset`	if an offset is used in the fit, then one must be supplied for making predictions
`...`	not used. Other arguments to predict.

Value

a vector/matrix of fitted values

Author(s)

Jairo Diaz

See Also

qut

Examples

```
set.seed(1234)
x=matrix(rnorm(200*20),200,20)
y1=x[,1]*10+rnorm(100)

fit1=qut(y1,x,family=gaussian,sigma=1)
predict(fit1,newx=x[1:5,])
predict(fit1,newx=x[1:5,],mode='lasso')
y1[1:5]

coef(fit1,mode='lasso')
coef(fit1,mode='glm')
```

processX *Process X matrix*

Description

Rescales and transforms the X matrix according to the desired parameters, and sets all the options required by the test.

Usage

```
processX(X, family = gaussian, alpha, intercept = TRUE, group.sizes = rep(1, ncol(X)),
A = ncol(X), LAD = FALSE, composite = TRUE,
M = min(10000, max(1000, 1e+10/nrow(X)/ncol(X))))
```

Arguments

`X` input matrix, of dimension n x p; each row is an observation vector.

`family` response type (see above). Default is `gaussian`.

`alpha` alpha for quantile rescaling; if alpha=0, then no rescaling.

`intercept` should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).

`group.sizes` the vector of group sizes for affine group lasso. The number of elements is L and sum(group.sizes) should be equal to P. If L==P, then the lasso test is employed, otherwise group lasso. Default is no groups, so `rep(1,ncol(X))`.

A if A is a matrix it tests A beta = c. If A is a vector, then it gives the indexes of the parameters to be tested. Used if family=gaussian. Default is to test beta=0, so A=ncol(X).

LAD set TRUE if LAD lasso test. Default is FALSE

composite set TRUE if composite test (O & +). Default is TRUE

M number of Monte Carlo Simulations to estimate the distribution Λ.

Value

an object containing all the variables corresponding to the rescaling and test options.

Author(s)

Sylvain Sardy and Jairo Diaz

Examples

```
# Test H0:beta=0
P=200
N=20
s=1
A=P
alpha=0.05
X=matrix(rnorm(N*P),N,P)
outrescale=processX(X,gaussian,alpha)
M=100 #Leave the default or select higher value for better level.

#when H0 is not rejected
beta_scal=0
beta=c(rep(beta_scal, s), rep(0, P-s))
y=X%*%beta+rnorm(N)
out=affinelassotest(y,X,gaussian,alpha,M=M,outrescale=outrescale)
print(out$rejectH0)

#when H0 is rejected
beta_scal=10
beta=c(rep(beta_scal, s), rep(0, P-s))
y=X%*%beta+rnorm(N)
out=affinelassotest(y,X,gaussian,alpha,M=M,outrescale=outrescale)
print(out$rejectH0)
```

qut *Fit a low dimensional GLM or Square-root lasso using the Quantile Universal Threshold*

Description

Variable selection with GLM-lasso or Square-root lasso choosing the penalty parameter λ with the Quantile Universal Threshold. The procedure goes towards sparse estimation of the coefficients for good selection of the important predictors.

Usage

```
qut(y,X,fit,family=gaussian,alpha.level=0.05,M=1000,qut.standardize=TRUE,
intercept=TRUE,offset=NULL,bootstrap=TRUE,sigma=ifelse(n>2*p,'ols','qut'),beta0='iterglm',
estimator='unbiased',type=c('glmnet','lars','flare'),lambda.seq=0,penalty.factor=rep(1,p),
lambda.min.ratio=ifelse(n<p,0.01,0.0001),nlambda=ifelse(type=='flare',2,100),
lambda=NULL,...)
```

Arguments

`y` response variable. Quantitative for family=`gaussian`, or family=`poisson` (non-negative counts). For family=`binomial` should be a factor with two levels.

`X` input matrix, of dimension n x p; each row is an observation vector.

`fit` a user supplied `glmnet` or lars object. Typical usage is to leave it empty so that the program computes the regularization path using the algorithm selected in `type`. WARNING: use with care, if supplied, object options must match with user supplied options.

`family` response type (see above). Default is `gaussian`.

`alpha.level` level, such that quantile $\tau = (1-\texttt{alpha.level})$. Default is 0.05.

`M` number of Monte Carlo Simulations to estimate the distribution Λ. Default is 1000.

`qut.standardize` standardize matrix X with a quantile-based standardization. Default is TRUE. It is not used for sqrt-lasso.

`intercept` should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).

`offset` a vector of length `n` that is included in the linear predictor. Useful for the `poisson` family (e.g. log of exposure time), or for refining a model by starting at a current fit. Default is NULL.

`bootstrap` set TRUE if it is desired to bootstrap matrix X when computing the Quantile Universal Threshold (Random scenario). Default is TRUE.

`sigma` standard deviation of the Gaussian errors. Used only if family=`gaussian`. When sigma = ’qut’, it is estimated based on the Quantile Universal Threshold (default if `n <= 2p`); when sigma = ’rcv’, it is estimated using Refitted Cross Validation in Fan et al. 2012; and when sigma = ’cv’, it is estimated using cross validation as in Reid et al. 2013. If `sigma` is a positive real number, then that value is used for the standard deviation. If `n>p` and sigma=’ols’ it is estimated using the ordinary least squeares estimator (default if `n>2p`)

`beta0` coefficients of the unpenalized covariates for generating the null data for the Quantile Universal Threshold. When is ’iterglm’ (Default) or ’iter’, it is estimated using one step iteration of the entire procedure with maximum likelihood estimation or the lasso estimation, respectively. If ’noiter’ then it is estimated without iterating. If it is desired to set `beta0` in advance, then it should be a vector of size the number of unpenalized covariates including the intercept if `intercept=TRUE`, in the same order. If there are not unpenalized covariates and `intercept=TRUE`, then it must be a real number.

`estimator` type of estimation of sigma when `sigma` = 'qut'. It can be equal to 'unbiased' (standard unbiased formula), or 'mle' (maximum likelihood formula).

`type` algorithm for solving the optimization problem. It can be `lars` (`type`='lars') or `glmnet` (`type`='glmnet') for GLM-lasso, or `flare` (`type`='flare') for Square-root lasso. For GLM-lasso, if family is not gaussian, penalty.factor is different from default, or offset different from NULL, glmnet will be always used. Default is 'glmnet'.

`lambda.seq` preset lambda sequence when type = 'glmnet'. If `lambda.seq`<2 the sequence of lambdas goes decreasing from `lambda.max` to `lambda.qut`. If `lambda.seq`= 0, `lambda` sequence is equispaced. If `lambda.seq`= 1, `lambda` sequence is equispaced in the log scale. Use `lambda.seq`=2 for `glmnet` default options. Default is 0.

`penalty.factor` separate penalty factors can be applied to each coefficient. This is a number that multiplies lambda to allow differential shrinkage. Can be 0 for some variables, which implies no shrinkage, and that variable is always included in the model. Default is 1 for all variables (and implicitly infinity for variables listed in exclude). Note: the penalty factors are internally rescaled to sum to n, and the lambda sequence will reflect this change.

`lambda.min.ratio`
smallest value for lambda, as a fraction of `lambda.max`. As in `glmnet`.

`nlambda` the number of `lambda`. As in `glmnet`. Default is 100.

`lambda` a user supplied `lambda` sequence. As in `glmnet`. Not used when `type`='flare'.

`...` `glmnet` or `lars` options.

Value

`lambda` value of the Quantile Universal Threshold.

`fit` object fitted by `glmnet` or `lars`.

`beta` coefficients obtained with the Quantile Universal Threshold.

`betaglm` coefficients obtained fitting GLM with the non zero coefficients in `beta`.

`beta0` estimated value of the intercept when family is not `gaussian`.

`family` response type

`sigma` standard deviation estimate of the errors (when family=`gaussian`)

`scale.factor` scale factor used for standardizing X.

Author(s)

Jairo Diaz Rodriguez

References

C. Giacobino, J. Diaz, S. Sardy, N. Hengartner. Quantile universal threshold for model selection. 2016 Jianqing Fan, Shaojun Guo and Ning Hao. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. Journal of the Royal Statistical Society: Series B. 2012 Stephen Reid, Robert Tibshirani, and Jerome Friedman. A Study of Error Variance Estimation in Lasso Regression. 2013

See Also

lambdaqut

Examples

```
set.seed(1234)
X=matrix(rnorm(50*500),50,500)
beta=c(rep(10,5),rep(0,500-5))
y=X %*% beta+rnorm(50)

outqut=qut(y,X,type='glmnet',family=gaussian,sigma=1) #Fitting with qut
betaqut=outqut$beta[-1]

outcv=cv.glmnet(X,y,family='gaussian') #fitting with Cross-Validation
betacv=coef(outcv$glmnet.fit,s=outcv$lambda.min)[-1]

results=rbind(  c(sum(betaqut[1:5]!=0),sum(betaqut[-(1:5)]!=0)),
c(sum( betacv[1:5]!=0), sum(betacv[-(1:5)]!=0)) )
colnames(results)=c('True Positive','False Positive')
rownames(results)=c('qut','cv')

print(results)
```

---

riboflavin *Riboflavin data set from Buhlmann et al. (2013)*

Description

Dataset of riboflavin production by Bacillus subtilis containing n = 71 observations of p = 4088 predictors (gene expressions) and a one-dimensional response (riboflavin production)

Usage

```
data(riboflavin)
```

Format

A data frame with 71 observations on the following 2 variables.

y a numeric vector

x a matrix with 4088 columns

References

Buhlmann, P., Kalisch, M. and Meier, L. (2013). High-dimensional statistics with a view towards applications in biology. To appear in Annual Review of Statistics and its Applications.

Examples

```
data(riboflavin)
```

sigmaqut *Estimation of σ based on the Quantile Universal Threshold*

Description

Estimation of σ using a two layer estimation scheme as in Refitted Cross Validation, by performing variable selection with the Quantile Universal Threshold, and obtaining the two estimations of sigma with the ordinary least squares estimator.

Usage

```
sigmaqut(y, X, estimator = "unbiased", intercept = TRUE,
alpha.level = "default", M = 1000, qut.standardize = TRUE,
penalty.factor = rep(1, p), offset = NULL, ...)
```

Arguments

`y` response variable. Quantitative for family=`gaussian`, or family=`poisson` (non-negative counts). For family=`binomial` should be a factor with two levels.

`X` input matrix, of dimension n x p; each row is an observation vector.

`estimator` type of estimation of sigma when `sigma` = 'qut'. It can be equal to 'unbiased' (standard unbiased formula), or 'mle' (maximum likelihood formula).

`intercept` should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).

`alpha.level` level, such that quantile $\tau = (1-\texttt{alpha.level})/\gamma$. Default is $1/(\sqrt{\pi \log(p)})$.

`M` number of Monte Carlo Simulations to estimate the distribution Λ. Default is 1000.

`qut.standardize` standardize matrix X with a quantile-based standardization. Default is TRUE.

`penalty.factor` separate penalty factors can be applied to each coefficient. As in `qut`.

`offset` a vector of length `n` that is included in the linear predictor. As in `qut`.

`...` other `glmnet` options.

Value

Estimator of σ

Note

`lambdaqut`,`qut`

Author(s)

Jairo Diaz

sigmarcv *Variance estimation using refitted cross-validation*

Description

Variance estimation using refitted cross-validation in ultrahigh dimensional regression.

Usage

```
sigmarcv(y, X, cv = FALSE, fit = NA, intercept = TRUE)
```

Arguments

`y`	response variable. Quantitative for family=`gaussian`, or family=`poisson` (non-negative counts). For family=`binomial` should be a factor with two levels.
`X`	input matrix, of dimension n x p; each row is an observation vector.
`cv`	when FALSE, variance is estimated using Refitted Cross Validation in Fan et al. 2012; and when TRUE, it is estimated using cross validation as in Reid et al. 2013. Default is FALSE.
`fit`	A user supplied `glmnet` or lars object. Typical usage is to leave it empty so that the program computes the regularization path using the algorithm selected in `type`. WARNING: use with care, if supplied, object options must match with user supplied options.
`intercept`	should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).

Value

Estimator of σ

Author(s)

Jianqing Fan, Shaojun Guo. Modified by Jairo Diaz.

References

Jianqing Fan, Shaojun Guo and Ning Hao. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. Journal of the Royal Statistical Society: Series B. 2012

ztf *Zero thresholding function*

Description

Obtains the value of the minimum regularization parameter that sets all coefficients to zero for different types of thresholding tests.

Usage

```
ztf(y,Xdata, family=gaussian, A=ncol(Xdata), cc=NA, intercept=TRUE,
group.sizes=rep(1,ncol(Xdata)), LAD=FALSE, outrescale=NA, composite=TRUE, alpha=0,
M=min(1.e4, max(1000,1.e10/nrow(Xdata)/ncol(Xdata))))
```

Arguments

`y` response variable. Quantitative for family=`gaussian`, or family=`poisson` (non-negative counts). For family=`binomial` should be a factor with two levels.

`Xdata` input matrix, of dimension n x p; each row is an observation vector.

`family` response type (see above). Default is `gaussian`.

`A` if A is a matrix it tests A beta = c. If A is a vector, then it gives the indexes of the parameters to be tested. Used if family=`gaussian`. Default is to test beta=0, so A=ncol(X).

`cc` vector c

`intercept` should intercept(s) be fitted (default=TRUE) or set to zero (FALSE).

`group.sizes` the vector of group sizes for affine group lasso. The number of elements is L and sum(group.sizes) should be equal to P. If L==P, then the lasso test is employed, otherwise group lasso. Default is no groups, so `rep(1,ncol(X))`.

`LAD` set TRUE if LAD lasso test. Default is FALSE

`outrescale` object containing all variables corresponding to the rescaling and test options. If not provided, this is calculated authomatically with function `processX`. Default is NA.

`composite` set TRUE if composite test (O & +). Default is TRUE

`alpha` alpha for quantile rescaling; if alpha=0, then no rescaling.

`M` number of Monte Carlo Simulations to estimate the distribution Λ.

Value

value of the minimum regularization parameter that sets all coefficients to zero

Author(s)

Sylvain Sardy and Jairo Diaz

Examples

```
# Test H0:beta=0
P=200
N=20
s=1
A=P
X=matrix(rnorm(N*P),N,P)
M=100 #Leave the default or select higher value for better level.
#when H0 is not rejected
beta_scal=0
beta=c(rep(beta_scal, s), rep(0, P-s))
y=X%*%beta+rnorm(N)
zerolambda=ztf(y,X,M=M)
print(zerolambda)

#when H0 is rejected
beta_scal=10
beta=c(rep(beta_scal, s), rep(0, P-s))
y=X%*%beta+rnorm(N)
zerolambda=ztf(y,X,M=M)
print(zerolambda)
```

Index