

Package: TFRE (via r-universe)

November 2, 2024

Type Package

Title A Tuning-Free Robust and Efficient Approach to High-Dimensional Regression

Version 0.1.0

Description Provide functions to estimate the coefficients in high-dimensional linear regressions via a tuning-free and robust approach. The method was published in Wang, L., Peng, B., Bradic, J., Li, R. and Wu, Y. (2020), "A Tuning-free Robust and Efficient Approach to High-dimensional Regression", Journal of the American Statistical Association, 115:532, 1700-1714(JASA's discussion paper), <doi:10.1080/01621459.2020.1840989>. See also Wang, L., Peng, B., Bradic, J., Li, R. and Wu, Y. (2020), "Rejoinder to "A tuning-free robust and efficient approach to high-dimensional regression". Journal of the American Statistical Association, 115, 1726-1729, <doi:10.1080/01621459.2020.1843865>; Peng, B. and Wang, L. (2015), "An Iterative Coordinate Descent Algorithm for High-Dimensional Nonconvex Penalized Quantile Regression", Journal of Computational and Graphical Statistics, 24:3, 676-694, <doi:10.1080/10618600.2014.913516>; Cléménçon, S., Colin, I., and Bellet, A. (2016), "Scaling-up empirical risk minimization: optimization of incomplete u-statistics", The Journal of Machine Learning Research, 17(1):2682-2717; Fan, J. and Li, R. (2001), "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties", Journal of the American Statistical Association, 96:456, 1348-1360, <doi:10.1198/016214501753382273>.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp (>= 1.0.9),RcppParallel

LinkingTo Rcpp,RcppEigen,RcppParallel

RoxygenNote 7.3.1

Repository <https://yunanwu123.r-universe.dev>

RemoteUrl <https://github.com/yunanwu123/tfre>

RemoteRef HEAD

RemoteSha 5dcee11988d5881ce8b20cc31007619895cad4ef

Contents

coef.TFRE	2
est_lambda	3
plot.TFRE	5
predict.TFRE	6
TFRE	7

Index	12
--------------	-----------

coef.TFRE	<i>Extract coefficients from a 'TFRE' object</i>
-----------	--

Description

Extract the coefficient vector from a fitted TFRE Lasso, SCAD or MCP model.

Usage

```
## S3 method for class 'TFRE'
coef(object, s, ...)
```

Arguments

object	Fitted "TFRE" model object.
s	Regression model to use for coefficient extraction. Should be one of "1st" and "2nd". See more details in "Details".
...	Not used. Other arguments to extract coefficients.

Details

If `object$second_stage = "none"`, `s` cannot be "2nd". If `object$second_stage = "none"` and `s = "2nd"`, the function will return the coefficient vector from the TFRE Lasso regression. If `object$second_stage = "scad"` or `"mcp"`, and `s = "2nd"`, the function will return the coefficient vector from the TFRE SCAD or MCP regression with the smallest HBIC.

Value

The coefficient vector from the fitted TFRE model, with the first element as the intercept.

Author(s)

Yunan Wu and Lan Wang
 Maintainer: Yunan Wu <yunan.wu@utdallas.edu>

References

Wang, L., Peng, B., Bradic, J., Li, R. and Wu, Y. (2020), *A Tuning-free Robust and Efficient Approach to High-dimensional Regression*, *Journal of the American Statistical Association*, 115:532, 1700-1714, doi:10.1080/01621459.2020.1840989.

See Also

[TFRE](#), [predict.TFRE](#), [plot.TFRE](#)

Examples

```
n <- 20; p <- 50
beta0 <- c(1.5, -1.25, 1, -0.75, 0.5, rep(0, p-5))
eta_list <- 0.1*6:15*sqrt(log(p)/n)
X <- matrix(rnorm(n*p), n)
y <- X %*% beta0 + rt(n, 4)
```

```
Obj_TFRE_Lasso <- TFRE(X, y, second_stage = "none", const_incomplete = 5)
coef(Obj_TFRE_Lasso, "1st")[1:10]
coef(Obj_TFRE_Lasso, "2nd")[1:10]
```

```
Obj_TFRE_SCAD <- TFRE(X, y, eta_list = eta_list, const_incomplete = 5)
coef(Obj_TFRE_SCAD, "1st")[1:10]
coef(Obj_TFRE_SCAD, "2nd")[1:10]
```

```
Obj_TFRE_MCP <- TFRE(X, y, second_stage = "mcp", eta_list = eta_list, const_incomplete = 5)
coef(Obj_TFRE_MCP, "1st")[1:10]
coef(Obj_TFRE_MCP, "2nd")[1:10]
```

est_lambda

Estimate the tuning parameter for a TFRE Lasso regression

Description

Estimate the tuning parameter of the TFRE Lasso regression given the covariate matrix X.

Usage

```
est_lambda(X, alpha0 = 0.1, const_lambda = 1.01, times = 500)
```

Arguments

X	Input matrix, of dimension n_obs x n_vars; each row is an observation vector.
alpha0	The level to estimate the tuning parameter. Default value is 0.1. See more details in "Details".

const_lambda	The constant to estimate the tuning parameter, should be greater than 1. Default value is 1.01. See more details in "Details".
times	The size of simulated samples to estimate the tuning parameter. Default value is 500.

Details

In TFRE Lasso regressions, the tuning parameter can be estimated independent of errors. In Wang *et al.* (2020), the following tuning parameter is suggested:

$$\lambda^* = \text{const_lambda} * G_{\|\mathbf{S}_n\|_\infty}^{-1}(1 - \text{alpha0})$$

, where $\mathbf{S}_n = -2[n(n-1)]^{-1} \sum_{j=1}^n \mathbf{x}_j[2r_j - (n+1)]$, r_1, \dots, r_n follows the uniform distribution on the per-mutations of the integers $\{1, \dots, n\}$, and $G_{\|\mathbf{S}_n\|_\infty}^{-1}(1 - \text{alpha0})$ denotes the $(1 - \text{alpha0})$ -quantile of the distribution of $\|\mathbf{S}_n\|_\infty$.

Value

The estimated tuning parameter of the TFRE Lasso regression given X.

Author(s)

Yunan Wu and Lan Wang
 Maintainer: Yunan Wu <yunan.wu@utdallas.edu>

References

Wang, L., Peng, B., Bradic, J., Li, R. and Wu, Y. (2020), *A Tuning-free Robust and Efficient Approach to High-dimensional Regression*, *Journal of the American Statistical Association*, 115:532, 1700-1714, doi:10.1080/01621459.2020.1840989.

See Also

[TFRE](#)

Examples

```
n <- 20; p <- 50
X <- matrix(rnorm(n*p), n)
est_lambda(X)
```

`plot.TFRE`*Plot the second stage model curve for a 'TFRE' object*

Description

Plot the HBIC curve and the model size curve as a function of the eta values used, from a fitted TFRE SCAD or MCP model.

Usage

```
## S3 method for class 'TFRE'  
plot(x, ...)
```

Arguments

<code>x</code>	A fitted "TFRE" model object. It should contain a second stage model.
<code>...</code>	Not used. Other arguments to be passed through plotting functions.

Details

In the output plot, the red line represents the HBIC curve as a function of eta values, the blue line represents the number of nonzero coefficients as a function of eta values, and the purple vertical dashed line denotes the model selected with the smallest HBIC.

This function cannot plot the object if `object$second_stage = "none"`.

Value

No return value, called for side effects.

Author(s)

Yunan Wu and Lan Wang
Maintainer: Yunan Wu <yunan.wu@utdallas.edu>

References

Wang, L., Peng, B., Bradic, J., Li, R. and Wu, Y. (2020), *A Tuning-free Robust and Efficient Approach to High-dimensional Regression*, *Journal of the American Statistical Association*, 115:532, 1700-1714, doi:[10.1080/01621459.2020.1840989](https://doi.org/10.1080/01621459.2020.1840989).

See Also

[TFRE](#), [predict.TFRE](#), [coef.TFRE](#)

Examples

```
n <- 20; p <- 50
beta0 <- c(1.5,-1.25,1,-0.75,0.5,rep(0,p-5))
eta_list <- 0.1*6:15*sqrt(log(p)/n)
X <- matrix(rnorm(n*p),n)
y <- X %*% beta0 + rt(n,4)

Obj_TFRE_SCAD <- TFRE(X, y, eta_list = eta_list, const_incomplete = 5)
plot(Obj_TFRE_SCAD)

Obj_TFRE_MCP <- TFRE(X, y, second_stage = "mcp", eta_list = eta_list, const_incomplete = 5)
plot(Obj_TFRE_MCP)
```

predict.TFRE	<i>Make predictions from a 'TFRE' object</i>
--------------	--

Description

Make predictions for new X values from a fitted TFRE Lasso, SCAD or MCP model.

Usage

```
## S3 method for class 'TFRE'
predict(object, newX, s, ...)
```

Arguments

object	Fitted "TFRE" model object.
newX	Matrix of new values for X at which predictions are to be made.
s	Regression model to use for prediction. Should be one of "1st" and "2nd". See more details in "Details".
...	Not used. Other arguments to predict.

Details

If `object$second_stage = "none"`, `s` cannot be "2nd". If `object$second_stage = "none"` and `s = "2nd"`, the function will return the predictions based on the TFRE Lasso regression. If `object$second_stage = "scad"` or `"mcp"`, and `s = "2nd"`, the function will return the predictions based on the TFRE SCAD or MCP regression with the smallest HBIC.

Value

A vector of predictions for the new X values given the fitted TFRE model.

Author(s)

Yunan Wu and Lan Wang
 Maintainer: Yunan Wu <yunan.wu@utdallas.edu>

References

Wang, L., Peng, B., Bradic, J., Li, R. and Wu, Y. (2020), *A Tuning-free Robust and Efficient Approach to High-dimensional Regression*, *Journal of the American Statistical Association*, 115:532, 1700-1714, doi:10.1080/01621459.2020.1840989.

See Also

[TFRE](#), [coef.TFRE](#), [plot.TFRE](#)

Examples

```
n <- 20; p <- 50
beta0 <- c(1.5, -1.25, 1, -0.75, 0.5, rep(0, p-5))
eta_list <- 0.1*6:15*sqrt(log(p)/n)
X <- matrix(rnorm(n*p), n)
y <- X %*% beta0 + rt(n, 4)
newX <- matrix(rnorm(10*p), 10)

Obj_TFRE_Lasso <- TFRE(X, y, second_stage = "none", const_incomplete = 5)
predict(Obj_TFRE_Lasso, newX, "1st")
predict(Obj_TFRE_Lasso, newX, "2nd")

Obj_TFRE_SCAD <- TFRE(X, y, eta_list = eta_list, const_incomplete = 5)
predict(Obj_TFRE_SCAD, newX, "1st")
predict(Obj_TFRE_SCAD, newX, "2nd")

Obj_TFRE_MCP <- TFRE(X, y, second_stage = "mcp", eta_list = eta_list, const_incomplete = 5)
predict(Obj_TFRE_MCP, newX, "1st")
predict(Obj_TFRE_MCP, newX, "2nd")
```

TFRE

Fit a TFRE regression model with Lasso, SCAD or MCP regularization

Description

Fit a TFRE Lasso model and/or a TFRE SCAD or MCP model. The TFRE regression models are fitted via QICD algorithm and *Incomplete U-statistics* resampling technique (optional). The tuning parameter of TFRE Lasso regression is estimated by the covariate matrix X. The TFRE SCAD / MCP regressions are computed at a grid of values for the tuning parameter eta. High dimensional BIC (HBIC) will be used as the criterion on the TFRE SCAD / MCP tuning parameter searching.

Usage

```

TFRE(
  X,
  y,
  alpha0 = 0.1,
  const_lambda = 1.01,
  times = 500,
  incomplete = TRUE,
  const_incomplete = 10,
  thresh = 1e-06,
  maxin = 100,
  maxout = 20,
  second_stage = "scad",
  a = 3.7,
  eta_list = NULL,
  const_hbic = 6
)

```

Arguments

X	Input matrix, of dimension $n_{\text{obs}} \times n_{\text{vars}}$; each row is an observation vector.
y	Response variable.
alpha0	The level to estimate the tuning parameter. Default value is 0.1. See more details in the "Details" section of est_lambda .
const_lambda	The constant to estimate the tuning parameter, should be greater than 1. Default value is 1.01. See more details in the "Details" section of est_lambda .
times	The size of simulated samples to estimate the tuning parameter. Default value is 500.
incomplete	Logical. If <code>incomplete = TRUE</code> , the <i>Incomplete U-statistics</i> resampling technique would be applied; if <code>incomplete = FALSE</code> , the complete U-statistics would be used in computation. See more details in Cl��men��on, Colin and Bellet (2016).
const_incomplete	The constant for the <i>Incomplete U-statistics</i> resampling technique. If <code>incomplete = TRUE</code> , <code>const_incomplete</code> x n_{obs} samples will be randomly selected in the coefficient estimation. Default value is 10. See more details in Cl��men��on, Colin and Bellet (2016).
thresh	Convergence threshold for QICD algorithm. Default value is $1e-6$. See more details in Peng and Wang (2015).
maxin	Maximum number of inner coordinate descent iterations in QICD algorithm; default is 100. See more details in Peng and Wang (2015).
maxout	Maximum number of outer Majorization Minimization step (MM) iterations in QICD algorithm; default is 20. See more details in Peng and Wang (2015).
second_stage	Penalty function for the second stage model. Character vector, which can be "scad", "mcp" and "none". If <code>second_stage = "scad"</code> , the TFRE SCAD re-

	gression would be fitted; if <code>second_stage = "mcp"</code> , the TFRE MCP regression would be fitted; if <code>scad = "none"</code> , only the TFRE Lasso regression outputs would be returned.
<code>a</code>	an unknown parameter in SCAD and MCP penalty functions. The default value is 3.7, suggested by Fan and Li (2001).
<code>eta_list</code>	A numerical vector for the tuning parameters to be used in the TFRE SCAD or MCP regression. Cannot be NULL if <code>second_stage = "scad"</code> or <code>"mcp"</code> .
<code>const_hbic</code>	The constant to be used in calculating HBIC in the TFRE SCAD regression. Default value is 6. See more details in "Details".

Details

Wang *et al.* (2020) proposed the TFRE Lasso estimator for high-dimensional linear regressions with heavy-tailed errors as below:

$$\hat{\beta}(\lambda^*) = \arg \min_{\beta} \frac{1}{n(n-1)} \sum \sum_{i \neq j} |(Y_i - \mathbf{x}_i^T \beta) - (Y_j - \mathbf{x}_j^T \beta)| + \lambda^* \sum_{k=1}^p |\beta_k|,$$

where λ^* is the tuning parameter estimated by `est_lambda`. The TFRE Lasso model is fitted by QICD algorithm proposed in Peng and Wang (2015). To overcome the computational barrier arising from the U-statistics structure of the aforementioned loss function, we apply the *Incomplete U-statistics* resampling technique which was first proposed in Cl  men  on, Colin and Bellet (2016). Wang *et al.* (2020) also proposed a second-stage enhancement by using the TFRE Lasso estimator $\hat{\beta}(\lambda^*)$ as an initial estimator. It is defined as:

$$\tilde{\beta}^{(1)} = \arg \min_{\beta} \frac{1}{n(n-1)} \sum \sum_{i \neq j} |(Y_i - \mathbf{x}_i^T \beta) - (Y_j - \mathbf{x}_j^T \beta)| + \sum_{k=1}^p p'_\eta(|\hat{\beta}_k(\lambda^*)|) |\beta_k|,$$

where $p'_\eta(\cdot)$ denotes the derivative of some nonconvex penalty function $p_\eta(\cdot)$, $\eta > 0$ is a tuning parameter. This function implements the second-stage enhancement with two popular nonconvex penalty functions: SCAD and MCP. The modified high-dimensional BIC criterion in Wang *et al.* (2020) is employed for selecting η . Define:

$$HBIC(\eta) = \log \left\{ \sum \sum_{i \neq j} |(Y_i - \mathbf{x}_i^T \tilde{\beta}_\eta) - (Y_j - \mathbf{x}_j^T \tilde{\beta}_\eta)| \right\} + |A_\eta| \frac{\log \log n}{n * \text{const_hbic}} \log p,$$

where $\tilde{\beta}_\eta$ denotes the second-stage estimator with the tuning parameter value η , and $|A_\eta|$ denotes the cardinality of the index set of the selected model. This function selects the value of η that minimizes $HBIC(\eta)$.

Value

An object of class "TFRE", which is a list containing at least the following components:

<code>X</code>	The input matrix used.
<code>y</code>	The response variable used.
<code>incomplete</code>	Logical. TRUE if the <i>Incomplete U-statistics</i> resampling technique is applied, and FALSE if not.

<code>beta_TFRE_Lasso</code>	The estimated coefficient vector of the TFRE Lasso regression. The first element is the estimated intercept.
<code>tfre_lambda</code>	The estimated tuning parameter of the TFRE Lasso regression.
<code>second_stage</code>	Character vector, "scad" if the TFRE SCAD regression is fitted, "mcp" if the TFRE MCP regression is fitted, "none" if only the TFRE Lasso regression is fitted.

If `second_stage = "scad"`, then the fitted TFRE object will also contain an object named as "TFRE_scad", which is a list containing the following components:

<code>Beta_TFRE_scad</code>	The estimated coefficient matrix of the TFRE SCAD regression. The dimension is $n_{\text{eta}} \times (p+1)$ with the first column to be the intercepts, where n_{eta} is the length of <code>eta_list</code> vector.
<code>df_TFRE_scad</code>	The number of nonzero coefficients (intercept excluded) for each value in <code>eta_list</code> .
<code>eta_list</code>	The tuning parameter vector used in the TFRE SCAD regressions
<code>hbic</code>	A numerical vector of HBIC values for the TFRE SCAD model corresponding to each value in <code>eta_list</code> .
<code>eta_min</code>	The eta value which yields the smallest HBIC value in the TFRE SCAD regression.
<code>Beta_TFRE_scad_min</code>	The estimated coefficient vector which employs <code>eta_min</code> as the eta value in the TFRE SCAD regression.

If `second_stage = "mcp"`, then the fitted TFRE object will also contain an object named as "TFRE_mcp", which is a list containing the following components:

<code>Beta_TFRE_mcp</code>	The estimated coefficient matrix of the TFRE MCP regression. The dimension is $n_{\text{eta}} \times (p+1)$ with the first column to be the intercepts, where n_{eta} is the length of <code>eta_list</code> vector.
<code>df_TFRE_mcp</code>	The number of nonzero coefficients (intercept excluded) for each value in <code>eta_list</code> .
<code>eta_list</code>	The tuning parameter vector used in the TFRE MCP regressions
<code>hbic</code>	A numerical vector of HBIC values for the TFRE MCP model corresponding to each value in <code>eta_list</code> .
<code>eta_min</code>	The eta value which yields the smallest HBIC value in the TFRE MCP regression.
<code>Beta_TFRE_mcp_min</code>	The estimated coefficient vector which employs <code>eta_min</code> as the eta value in the TFRE MCP regression.

Author(s)

Yunan Wu and Lan Wang
 Maintainer: Yunan Wu <yunan.wu@utdallas.edu>

References

- Wang, L., Peng, B., Bradic, J., Li, R. and Wu, Y. (2020), *A Tuning-free Robust and Efficient Approach to High-dimensional Regression*, *Journal of the American Statistical Association*, 115:532, 1700-1714, doi:10.1080/01621459.2020.1840989.
- Peng, B. and Wang, L. (2015), *An Iterative Coordinate Descent Algorithm for High-Dimensional Nonconvex Penalized Quantile Regression*, *Journal of Computational and Graphical Statistics*, 24:3, 676-694, doi:10.1080/10618600.2014.913516.
- Cléménçon, S., Colin, I., and Bellet, A. (2016), *Scaling-up empirical risk minimization: optimization of incomplete u-statistics*. *The Journal of Machine Learning Research*, 17(1):2682–2717.
- Fan, J. and Li, R. (2001), *Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties*, *Journal of the American Statistical Association*, 96:456, 1348-1360, doi:10.1198/016214501753382273.

See Also

[predict.TFRE](#), [coef.TFRE](#), [plot.TFRE](#), [est_lambda](#)

Examples

```
n <- 20; p <- 50
beta0 <- c(1.5,-1.25,1,-0.75,0.5,rep(0,p-5))
eta_list <- 0.1*6:15*sqrt(log(p)/n)
X <- matrix(rnorm(n*p),n)
y <- X %*% beta0 + rt(n,4)
```

```
Obj_TFRE_Lasso <- TFRE(X, y, second_stage = "none", const_incomplete = 5)
Obj_TFRE_Lasso$beta_TFRE_Lasso[1:10]
```

```
Obj_TFRE_SCAD <- TFRE(X, y, eta_list = eta_list, const_incomplete = 5)
Obj_TFRE_SCAD$TFRE_scad$hbic
Obj_TFRE_SCAD$TFRE_scad$df_TFRE_scad
Obj_TFRE_SCAD$TFRE_scad$Beta_TFRE_scad_min[1:10]
```

```
Obj_TFRE_MCP <- TFRE(X, y, second_stage = "mcp", eta_list = eta_list, const_incomplete = 5)
Obj_TFRE_MCP$TFRE_mcp$hbic
Obj_TFRE_MCP$TFRE_mcp$df_TFRE_mcp
Obj_TFRE_MCP$TFRE_mcp$Beta_TFRE_mcp_min[1:10]
```

Index

`coef.TFRE`, [2](#), [5](#), [7](#), [11](#)

`est_lambda`, [3](#), [8](#), [9](#), [11](#)

`plot.TFRE`, [3](#), [5](#), [7](#), [11](#)

`predict.TFRE`, [3](#), [5](#), [6](#), [11](#)

`TFRE`, [3–5](#), [7](#), [7](#)