Package ‘robustlm’
March 22, 2021

Type Package
Title Robust Variable Selection with Exponential Squared Loss

Version 0.1.0
Date 2021-03-21
Maintainer Jin Zhu <zhuji37@mail2.sysu.edu.cn>
Imports MASS, matrixStats
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Suggests knitr, rmarkdown,
VignetteBuilder knitr
NeedsCompilation no
Author Jin Zhu [cre, aut] (<https://orcid.org/0000-0001-8550-5822>),
Borui Tang [aut],
Yunlu Jiang [aut],
Xueqin Wang [aut] (<https://orcid.org/0000-0001-5205-9950>)
Repository CRAN
Date/Publication 2021-03-22 15:40:02 UTC

R topics documented:

 coef.robustlm ................................................................. 2
 predict.robustlm .......................................................... 2
 print.robustlm .............................................................. 3
 robustlm ................................................................. 3

Index 6
coef.robustlm  
*Provides estimated coefficients from a fitted "robustlm" object.*

### Description
This function provides estimated coefficients from a fitted "robustlm" object.

### Usage
```
## S3 method for class 'robustlm'
coef(object, ...)
```

### Arguments
- `object`  
  An "robustlm" project.
- `...`  
  Other arguments.

### Value
A list consisting of the intercept and regression coefficients of the fitted model.

---

predict.robustlm  
*Make predictions from a "robustlm" object.*

### Description
Returns predictions from a fitted "robustlm" object.

### Usage
```
## S3 method for class 'robustlm'
predict(object, newx, ...)
```

### Arguments
- `object`  
  Output from the robustlm function.
- `newx`  
  New data used for prediction
- `...`  
  Additional arguments affecting the predictions produced.

### Value
The predicted responses.
**print.robustlm**

*Print method for a "robustlm" object*

#### Description

Print the primary elements of the "robustlm" object.

#### Usage

```
## S3 method for class 'robustlm'
print(x, ...)
```

#### Arguments

- `x`: A "robustlm" object.
- `...`: Additional print arguments.

#### Value

Print a robustlm object.

---

**robustlm**

*Robust variable selection with exponential squared loss*

#### Description

robustlm carries out robust variable selection with exponential squared loss. A block coordinate gradient descent algorithm is used to minimize the loss function.

#### Usage

```
robustlm(x, y, gamma = NULL, weight = NULL, intercept = TRUE)
```

#### Arguments

- `x`: Input matrix, of dimension nobs * nvars; each row is an observation vector. Should be in matrix format.
- `y`: Response variable. Should be a numerical vector or matrix with a single column.
- `gamma`: Tuning parameter in the loss function, which controls the degree of robustness and efficiency of the regression estimators. The loss function is defined as
  
  \[ 1 - \exp\left(-t^2/\gamma\right). \]

  When `gamma` is large, the estimators are similar to the least squares estimators in the extreme case. A smaller `gamma` would limit the influence of an outlier on the estimators, although it could also reduce the sensitivity of the estimators. If `gamma=NULL`, it is selected by a data-driven procedure that yields both high robustness and high efficiency.
weight

Weight in the penalty. The penalty is given by

\[ n \sum_{j=1}^{d} \lambda_{nj}|\beta_j|, \]

weight is a vector consisting of \( \lambda_{nj} \)'s. If weight=NULL (by default), it is set to be \((log(n))/(n|\tilde{\beta}_j|))\), where \( \tilde{\beta} \) is a numeric vector, which is an initial estimator of regression coefficients obtained by an MM procedure. The default value meets a BIC-type criterion (See Details).

intercept

Should intercepts be fitted (TRUE) or set to zero (FALSE)

Details

robustlm solves the following optimization problem to obtain robust estimators of regression coefficients:

\[
\text{argmin}_{\beta} \sum_{i=1}^{n} (1 - \exp(-(y_i - x_i^T \beta)^2/\gamma_n)) + n \sum_{j=1}^{d} p_{\lambda_{nj}}(|\beta_j|),
\]

where \( p_{\lambda_{nj}}(|\beta_j|) = \lambda_{nj}|\beta_j| \) is the adaptive LASSO penalty. Block coordinate gradient descent algorithm is used to efficiently solve the optimization problem. The tuning parameter \( \gamma \) and regularization parameter \( \text{weight} \) are chosen adaptively by default, while they can be supplied by the user. Specifically, the default \text{weight} meets the following BIC-type criterion:

\[
\min_{\tau_n} \sum_{i=1}^{n} [1 - \exp(-(Y_i - x_i^T \beta)^2/\gamma_n)] + n \sum_{j=1}^{d} \tau_{nj}|\beta_j|/|\tilde{\beta}_{nj}| - \sum_{j=1}^{d} \log(0.5 n \tau_{nj}) \log(n).
\]

Value

An object with S3 class "robustlm", which is a list with the following components:

beta

The regression coefficients.

alpha

The intercept.

gamma

The tuning parameter used in the loss.

weight

The regularization parameters.

loss

Value of the loss function calculated on the training set.

Author(s)

Borui Tang, Jin Zhu, Xueqin Wang

References


Examples

```r
library(MASS)
N <- 100
p <- 8
rho <- 0.2
mu <- rep(0, p)
Sigma <- rho * outer(rep(1, p), rep(1, p)) + (1 - rho) * diag(p)
ind <- 1:p
beta <- (-1)^ind * exp(-2 * (ind - 1) / 20)
lambda_seq <- seq(0.05, 5, length.out = 100)
X <- mvrnorm(N, mu, Sigma)
Z <- rnorm(N, 0, 1)
k <- sqrt(var(X %*% beta) / (3 * var(Z)))
Y <- X %*% beta + drop(k) * Z
robustlm(X, Y)
```
Index

coefficients, 2
predict, 2
print, 3
robustlm, 3