Package 'L2E'

January 7, 2022

Title Robust Structured Regression via the L2 Criterion

Description An implementation of a computational framework for performing robust structured regression with the L2 criterion from Chi and Chi (2021+).

Version 1.0

Depends R (>= 3.5.0), isotone, cobs

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://jocelynchi.com/l2e/

License GPL (>= 2)

LazyData true

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Jocelyn Chi [aut, cre],
    Eric Chi [aut]

Maintainer Jocelyn Chi <jocetchi@gmail.com>

Repository CRAN

Date/Publication 2022-01-07 13:32:55 UTC

R topics documented:

---

bank | *Bank data*

Description

Data from an Italian bank on 1,949 customers. The response y is the amount of money made over a year. The 13 covariates are possible macroservices the customers can sign up for.

Format

A data frame with 1949 rows and 14 variables

References

Marco Riani, Andrea Cerioli, and Anthony C. Atkinson (2014). Monitoring robust regression. Electronic Journal of Statistics, Volume 8, 646-677. https://projecteuclid.org/euclid.ejs/1400592267.

L2E | *L2E*

Description

The L2E package is an R implementation of a user-friendly computational framework for performing a wide variety of robust structured regression methods via the L2 criterion.

Details

Please refer to the vignette for examples of how to use this package.

---

l2e_regression | *LTE multivariate regression*

Description

l2e_regression Performs L2E regression via inexact coordinate descent

Usage

```
l2e_regression(y, X, tau, b, max_iter = 100, tol = 1e-04)
```

Arguments

y	Response vector
X	Design matrix
tau	Initial precision estimate
b	Initial vector of regression coefficients
max_iter	Maximum number of iterations
tol	Relative tolerance

Value

Returns a list object containing the estimates for beta (vector) and tau (scalar)

Examples

```
# Bank data example
y <- bank$y
X <- as.matrix(bank[,1:13])
X0 <- as.matrix(cbind(rep(1,length(y)), X))
tauinit <- 1/mad(y)
binit <- matrix(0, 14, 1)#'
sol <- l2e_regression(y, X0, tauinit, binit)
r <- y - X0 %*% sol$beta
ix <- which(abs(r) > 3/sol$tau)
l2e_fit <- X0 %*% sol$beta
plot(y, l2e_fit, ylab='Predicted values', pch=16, cex=0.8)
points(y[ix], l2e_fit[ix], pch=16, col='blue', cex=0.8)
```

---

l2e_regression_convex *LTE convex regression*

Description

l2e_regression_convex Performs robust convex regrsesion using the L2 criterion

Usage

```
l2e_regression_convex(y, b, tau, max_iter = 100, tol = 1e-04)
```

Arguments

y	response
b	initial vector of regression coefficients
tau	initial precision estimate
max_iter	maximum number of iterations
tol	relative tolerance

Value

Returns a list object containing the estimates for beta (vector) and tau (scalar)

Examples

```
set.seed(12345)
n <- 200
tau <- 1
x <- seq(-2, 2, length.out=n)
f <- x^4 + x
y <- f + (1/tau) * rnorm(n)

## Clean data example
plot(x, y, pch=16, cex.lab=1.5, cex.axis=1.5, cex.sub=1.5)
lines(x, f, col='blue', lwd=3)

tau <- 1
b <- y
sol <- l2e_regression_convex(y,b,tau)

plot(x, y, pch=16, cex.lab=1.5, cex.axis=1.5, cex.sub=1.5)
lines(x, f, col='blue', lwd=3)
cvx <- fitted(cobs::conreg(y, convex=TRUE))
lines(x, cvx, col='red', lwd=3)
lines(x, sol$beta, col='green', lwd=3)

## Contaminated data example
ix <- 0:9
y[45 + ix] <- 14 + rnorm(10)

plot(x, y, pch=16, cex.lab=1.5, cex.axis=1.5, cex.sub=1.5)
lines(x, f, col='blue', lwd=3)

tau <- 1
b <- y
sol <- l2e_regression_convex(y, b, tau)

plot(x, y, pch=16, cex.lab=1.5, cex.axis=1.5, cex.sub=1.5)
lines(x, f, col='blue', lwd=3)
cvx <- fitted(cobs::conreg(y, convex=TRUE))
lines(x, cvx, col='red', lwd=3)
lines(x, sol$beta, col='green', lwd=3)
```

---

l2e_regression_isotonic

LTE isotonic regression

Description

`l2e_regression_isotonic` Performs L2E isotonic regression via inexact coordinate descent.

Usage

```
l2e_regression_isotonic(y, b, tau, max_iter = 100, tol = 1e-04)
```

Arguments

y	Response vector
b	Initial vector of regression coefficients
tau	Initial precision estimate
max_iter	Maximum number of iterations
tol	Relative tolerance

Value

Returns a list object containing the estimates for beta (vector) and tau (scalar)

Examples

```
set.seed(12345)
n <- 200
tau <- 1
x <- seq(-2.5, 2.5, length.out=n)
f <- x^3
y <- f + (1/tau)*rnorm(n)

# Clean Data
plot(x, y, pch=16)
lines(x, f, col='blue', lwd=3)

tau <- 1
b <- y
sol <- l2e_regression_isotonic(y, b, tau)

plot(x, y, pch=16)
lines(x, f, col='blue', lwd=3)
iso <- gpava(1:n, y)$x
lines(x, iso, col='red', lwd=3)
lines(x, sol$beta, col='green', lwd=3)

# Contaminated Data
ix <- 0:9
y[45 + ix] <- 14 + rnorm(10)

plot(x, y, pch=16)
lines(x, f, col='blue', lwd=3)
```

```
tau <- 1
b <- y
sol <- l2e_regression_isotonic(y, b, tau)

plot(x, y, pch=16)
lines(x, f, col='blue', lwd=3)
iso <- gpava(1:n, y)$x
lines(x, iso, col='red', lwd=3)
lines(x, sol$beta, col='green', lwd=3)
```

---

update_beta_convex   *Beta update in LTE convex regression*

Description

update_beta_convex Function for updating beta in LTE convex regression

Usage

```
update_beta_convex(y, b, tau, max_iter = 100, tol = 1e-04)
```

Arguments

y	Response vector
b	Current estimate for beta
tau	Current estimate for tau
max_iter	Maximum number of iterations
tol	Relative tolerance

Value

Returns a list object containing the new estimate for beta (vector) and the number of iterations (scalar) the update step utilized

---

update_beta_isotonic   *Beta update in LTE isotonic regression*

Description

update_beta_isotonic Function for updating beta in LTE isotonic regression

Usage

```
update_beta_isotonic(y, b, tau, max_iter = 100, tol = 1e-04)
```

Arguments

y	Response vector
b	Current estimate for beta
tau	Current estimate for tau
max_iter	Maximum number of iterations
tol	Relative tolerance

Value

Returns a list object containing the new estimate for beta (vector) and the number of iterations (scalar) the update step utilized

---

update_beta_qr	*Beta update in LTE multivariate regression*

Description

update_beta_qr Function for updating beta in LTE multivariate regression via a QR solve

Usage

```
update_beta_qr(y, X, QRF, tau, b, max_iter = 100, tol = 1e-04)
```

Arguments

y	Response vector
X	Design matrix
QRF	QR factorization object for X (obtained via 'QRF=qr(X)')
tau	Current estimate for tau
b	Current estimate for beta
max_iter	Maximum number of iterations
tol	Relative tolerance

Value

Returns a list object containing the new estimate for beta (vector) and the number of iterations (scalar) the update step utilized

update_tau_R	*Tau update function*

Description

`update_tau_R` Function for updating tau

Usage

```
update_tau_R(r, tau, sd_y, max_iter = 100, tol = 1e-10)
```

Arguments

r	Residual vector
tau	Current estimate for tau
sd_y	Standard deviation of y
max_iter	Maximum number of iterations
tol	Relative tolerance

Value

Returns a list object containing the new estimate for tau (scalar) and the number of iterations (scalar) the update step utilized

Index