

Smooth Quantile Regression in R A VIGNETTE

Installation

Like many other R packages, the simplest way to obtain SQR is to install it from github. Type the following command in R console:

```
library(devtools)
```

```
## Le chargement a nécessité le package : usethis
```

```
devtools::install_github("https://github.com/ouhourane/SQR.git")
```

```
## Skipping install of 'SQR' from a github remote, the SHA1 (cd4f6e2c) has not changed since last install.
## Use `force = TRUE` to force installation
```

The SmoothSQR algorithm combines the idea of on the approximation of the quantile objective check function, which is not differentiable at zero, by a modified check function which is differentiable at zero, with that of the Majorization Minimization (MM) principale. Then, using the the coordinate descent algorithm to update each coefficient simply and efficiently. We introduce a new R package SmoothSQR which implements a coordinate descent algorithm for computing penalized smooth quantile regression on convex penalties $l_1 + l_2$ and the nonconvex penalties SCAD (or MCP) $+l_2$. This package implements k fold cross validation as tuning parameter selection criterion.

This vignette contains only a brief introduction to utilize SmoothSQR to solve convex and non-convex penalized quantile regression under high-dimensional settings. We are given n observations on a real-valued response variable y and a set of p predictors, $x = (x_1, \dots, x_p)$, and we try to infer the relationship between y and x . We are interested in identifying and estimating the nonzero component of β when $p \gg n$.

Quantile regression (QR) was developed to analyse the relationship between the conditional quantiles of the response distribution and a set of regressors. The aim of QR is to estimate the parameter vector, $(\beta_0(\tau), \beta(\tau))$, for each quantile $\tau \in (0, 1)$, of a conditional quantile function. The penalized quantile regression estimator for β is obtained by minimizing:

$$(\hat{\beta}_0(\tau), \hat{\beta}_\tau) = \arg \min_{\beta_0(\tau), \beta_\tau} \frac{1}{n} \sum_{i=1}^n \rho_{\tau,*} \{y_i - \beta_0(\tau) - \mathbf{x}_i^\top \beta_\tau\} + P_{\lambda_1, \lambda_2}(\beta_\tau),$$

where $\rho_{\tau,*}(\cdot)$ is a smooth function to approximate the “check function” of the quantile regression and where P_{λ_1, λ_2} is the elastic net, Snet (SCAD+ l_2) or Mnet (Mcp+ l_2) penalty.

Example 1:

The diabetes dataset consists of 10 physiological variables (age, sex, weight, blood pressure) measure on 442 patients, and an indication of disease progression after one year. this dataset included in the lars package. The main function in the package SmoothSQR, it's has many option parameters.

```
#### Lasso
```

```
library(lars)
```

```
## Loaded lars 1.2
```

```

data(diabetes)
attach(diabetes)

par(mfrow = c(1, 3))
library(SQR)

## Le chargement a nécessité le package : Matrix

### Lasso penalty
fit <- SQR(x=diabetes$x,y=diabetes$y,lambda2=0.0,check="f1",taux=0.5, method = "Lasso")
plot(fit)

### Scad penalty
fit <- SQR(x=diabetes$x,y=diabetes$y,lambda2=0.0,check="f1",taux=0.5, method = "Scad")
plot(fit)

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

### Mcp penalty
fit <- SQR(x=diabetes$x,y=diabetes$y,lambda2=0.0,check="f1",taux=0.5, method = "Scad")
plot(fit)

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

```

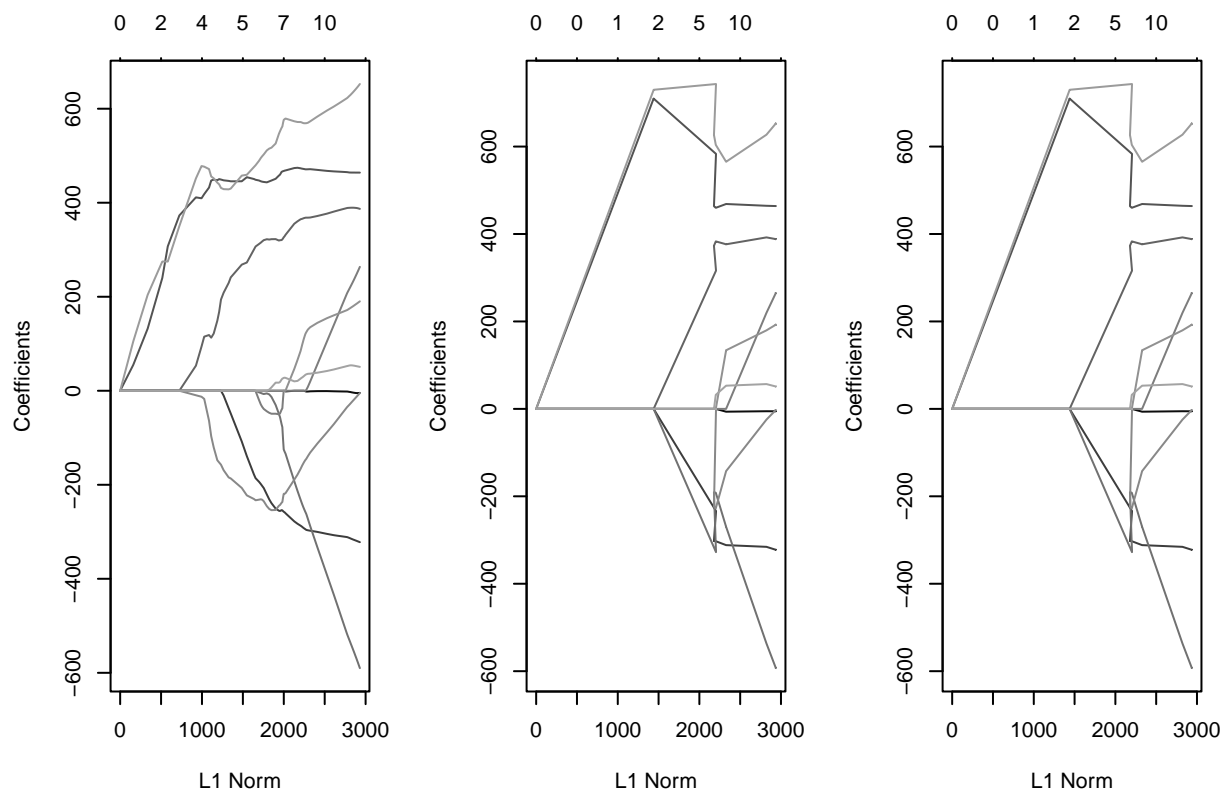


Figure1: Solution paths of the penalized quantile regression on the diabetes data with 442 observations and 10 predictors. Panels, from the left to the right, show the solution paths SmoothSQR-Elastic net, SmoothSQR-Snet and SmoothSQR-Mnet algorithms, respectively.

Example 2

To illustrate how this package works, we use the same setting in Mkhadri et al. (2018). We generated one data sets with 100 observations and 200 predictors. The jk -th elements of Σ are given as $0.5^{|j-k|}$, $\sigma = 1$ and

$$\beta = (\underbrace{3, \dots, 3}_5, \underbrace{-1.5, \dots, -1.5}_5, \underbrace{1, \dots, 1}_5, \underbrace{2, \dots, 2}_5, \mathbf{0}_{180}).$$