ADMM for High-Dimensional Sparse Penalized Quantile Regression

Qinqin Zou and Qian Tang

University of Iowa

May 8, 2023

Quantile Regression

- τth quantile: $\tau \in (0,1)$, $Q_Y(\tau) = \inf_y\{y : F(y) \geq \tau\}$.
- **Quantile Regression** (Koenker and Basssett, 1978):

$$\min_{b,\beta} \sum_{i=1}^{n} \rho_\tau \left(y_i - b - x_i'\beta\right); \text{ **check loss**: } \rho_\tau(t) = t(\tau - I(t < 0)).$$

- Median regression ($\tau = 0.5$):

$$\min_{b,\beta} \sum_{i=1}^{n} \left|y_i - b - x_i'\beta\right|$$

Sparse Penalized Quantile Regression

- Sparse penalized quantile regression:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau \left(y_i - \mathrm{x}_i^{\mathrm{T}} \boldsymbol{\beta} \right) + \sum_{j=1}^{n} p_\lambda \left(|\beta_j| \right)$$

  - $p_\lambda(\cdot), \lambda > 0$ is the penalty function: lasso penalty
- Weighted L_1-penalized quantile regression:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau \left(y_i - \mathrm{x}_i^{\mathrm{T}} \boldsymbol{\beta} \right) + \lambda \| \mathrm{w} \circ \boldsymbol{\beta} \|_1$$

  - lasso penalized quantile regression: $\mathrm{w} = 1_p$
  - adaptive lasso penalized quantile regression: $w_j = \left(\left| \hat{\beta}_j^{\text{lasso}} \right| + 1/n \right)^{-1}$, $j = 1, \ldots, p$

Weighted L_1-penalized quantile regression

- Weighted L_1-penalized quantile regression:

$$\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} \rho_\tau \left(y_i - \mathrm{x}_i^{\mathrm{T}} \boldsymbol{\beta} \right) + \lambda \| \mathrm{w} \circ \boldsymbol{\beta} \|_1 \tag{1}$$

- problem (1) is equivalent to:

$$\begin{aligned} &\min_{\beta, \mathrm{z}} \mathbb{Q}_\tau(\mathrm{z}) + \lambda \| \mathrm{w} \circ \boldsymbol{\beta} \|_1 \\ &\text{subject to } \mathrm{X}\boldsymbol{\beta} + \mathrm{z} = \mathrm{y}. \end{aligned} \tag{2}$$

where $\mathbb{Q}_\tau(\mathrm{z}) = (1/n) \sum_{i=1}^{n} \rho_\tau (z_i)$ and $\mathrm{z} = \mathrm{y} - \mathrm{X}\boldsymbol{\beta}$

ADMM Algorithm

Fix $\sigma > 0$ and the augmented Lagarangian function of (2) is:

$$\mathcal{L}_\sigma(\boldsymbol{\beta}, \mathsf{z}, \boldsymbol{\theta}) := \mathbb{Q}_\tau(\mathsf{z}) + \lambda\|\mathsf{w} \circ \boldsymbol{\beta}\|_1 - \langle \boldsymbol{\theta}, \mathsf{X}\boldsymbol{\beta} + \mathsf{z} - \mathsf{y}\rangle + \frac{\sigma}{2}\|\mathsf{X}\boldsymbol{\beta} + \mathsf{z} - \mathsf{y}\|_2^2$$

Following Boyd et al. (2011), the iterations for the standard ADMM algorithm are given by

$$\boldsymbol{\beta} \textit{ step}: \quad \boldsymbol{\beta}^{k+1} := \arg\min_{\boldsymbol{\beta}} \lambda\|\mathsf{w} \circ \boldsymbol{\beta}\|_1 - \left\langle \boldsymbol{\theta}^k, \mathsf{X}\boldsymbol{\beta} \right\rangle + \frac{\sigma}{2}\left\|\mathsf{X}\boldsymbol{\beta} + \mathsf{z}^k - \mathsf{y}\right\|_2^2$$

$$\mathsf{z} \textit{ step}: \quad \mathsf{z}^{k+1} := \arg\min \mathbb{Q}_\tau(\mathsf{z}) - \left\langle \boldsymbol{\theta}^k, \mathsf{z} \right\rangle + \frac{\sigma}{2}\left\|\mathsf{z} + \mathsf{X}\boldsymbol{\beta}^{k+1} - \mathsf{y}\right\|_2^2$$

$$\boldsymbol{\theta} \textit{ step}: \quad \boldsymbol{\theta}^{k+1} := \boldsymbol{\theta}^k - \sigma\left(\mathsf{X}\boldsymbol{\beta}^{k+1} + \mathsf{z}^{k+1} - \mathsf{y}\right).$$

ADMM Algorithm

- the proximal mapping: $\text{Prox}_{\rho_\tau}[\xi, \alpha] := \arg\min_{u \in \mathbb{R}} \rho_\tau(u) + \frac{\alpha}{2}(u - \xi)^2$
- For $i = 1, \ldots, n$, we have

$$\begin{aligned} z_i^{k+1} &:= \arg\min_{z_i} \frac{1}{n}\rho_\tau(z_i) - \theta_i^k z_i + \frac{\sigma}{2}\left(z_i + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{k+1} - y_i\right)^2 \\ &= \text{Prox}_{\rho_\tau}\left[y_i - \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \frac{1}{\sigma}\theta_i^k, n\sigma\right] \end{aligned} \tag{3}$$

Lemma 1

Given $\tau \in (0, 1)$ and $\alpha > 1$, the proximal mapping $\text{Prox}_{\rho_\tau}[\xi, \alpha] := \arg\min_{u \in \mathbb{R}} \rho_\tau(u) + \frac{\alpha}{2}(u - \xi)^2$ has explicit expression,

$$\text{Prox}_{\rho_\tau}[\xi, \alpha] = \begin{cases} \xi - \frac{\tau}{\alpha}, & \text{if} \xi > \frac{\tau}{\alpha} \\ 0, & \text{if} \frac{\tau - 1}{\alpha} \leq \xi \leq \frac{\tau}{\alpha} \\ \xi - \frac{\tau - 1}{\alpha} & \text{if} \xi < \frac{\tau - 1}{\alpha}. \end{cases}$$

ADMM Algorithm

- $\boldsymbol{\beta}$ *step* : $\boldsymbol{\beta}^{k+1} := \arg\min_{\boldsymbol{\beta}} \lambda \|\mathsf{w} \circ \boldsymbol{\beta}\|_1 - \left\langle \boldsymbol{\theta}^k, \mathsf{X}\boldsymbol{\beta} \right\rangle + \frac{\sigma}{2} \left\| \mathsf{X}\boldsymbol{\beta} + \mathsf{z}^k - \mathsf{y} \right\|_2^2$
- Augmented $\boldsymbol{\beta}$ step;

$$\begin{aligned} \boldsymbol{\beta}^{k+1} :=& \arg\min_{\beta} \lambda \|\mathsf{w} \circ \boldsymbol{\beta}\|_1 - \left\langle \boldsymbol{\theta}^k, \mathsf{X}\boldsymbol{\beta} \right\rangle \\ &+ \frac{\sigma}{2} \left\| \mathsf{X}\boldsymbol{\beta} + \mathsf{z}^k - \mathsf{y} \right\|_2^2 \\ &+ \frac{1}{2} \left\| \boldsymbol{\beta} - \boldsymbol{\beta}^k \right\|_{\mathsf{S}}^2 \end{aligned}$$

- $\|\mathsf{v}\|_{\mathsf{s}}^2 := \langle \mathsf{v}, \mathsf{S}\mathsf{v} \rangle$
- $\mathsf{S} = \sigma \left(\eta \mathsf{I}_p - \mathsf{X}^\top \mathsf{X} \right)$ with $\eta \geq \Lambda_{\max} \left(\mathsf{X}^\top \mathsf{X} \right)$

Soft-thresholding

For the problem,

$$\min_{\boldsymbol{x}} \|\boldsymbol{X} - B\|_2^2 + \lambda \|\boldsymbol{X}\|_1$$

the solution can be written as

$$\hat{x}_i = \text{shrink}[b_i, \lambda/2] = \text{sgn}(b_i)\max(|b_i| - \lambda/2, 0).$$

$$\begin{aligned}
\boldsymbol{\beta}^{k+1} =& \arg\min_{\boldsymbol{\beta}} \lambda \|\mathrm{w} \circ \boldsymbol{\beta}\|_1 \\
&+ \frac{\sigma\eta}{2} \left\| \boldsymbol{\beta} - \frac{\sigma\eta\boldsymbol{\beta}^k + \mathrm{X}^{\mathrm{T}}\left(\boldsymbol{\theta}^k + \sigma\mathrm{y} - \sigma\mathrm{X}\boldsymbol{\beta}^k - \sigma\mathrm{z}^k\right)}{\sigma\eta} \right\|_2^2 \\
&= \left(\text{Shrink}\left[\beta_j^k + \frac{1}{\sigma\eta} X_j^{\mathrm{T}} \left(\boldsymbol{\theta}^k + \sigma\mathrm{y} - \sigma\mathrm{X}\boldsymbol{\beta}^k - \sigma\mathrm{z}^k \right), \frac{\lambda w_j}{\sigma\eta} \right] \right)_{1 \le j \le p}
\end{aligned} \tag{4}$$

pADMM Algorithm

Algorithm 1 pADMM – Proximal ADMM algorithm for solving the weighted L_1-penalized quantile regression.

1. Initialize the algorithm with $(\boldsymbol{\beta}^0, \mathrm{z}^0, \boldsymbol{\theta}^0)$.
2. For $k = 0, 1, 2, \ldots,$ repeat steps (2.1)–(2.3) until the convergence criterion is met.

(2.1) Update $\boldsymbol{\beta}^{k+1} \leftarrow$

$$\left(\mathrm{Shrink}\left[\beta_j^k + \frac{1}{\sigma\eta} X_j^{\mathrm{T}}(\boldsymbol{\theta}^k + \sigma\mathbf{y} - \sigma\mathbf{X}\boldsymbol{\beta}^k - \sigma\mathbf{z}^k), \frac{\lambda w_j}{\sigma\eta}\right]\right)_{1\le j\le p}.$$

(2.2) Update $\mathbf{z}^{k+1} \leftarrow \left(\mathrm{Prox}_{\rho_\tau}\left[y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{k+1} + \sigma^{-1}\theta_i^k, n\sigma\right]\right)_{1\le i\le n}.$

(2.3) Update $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k - \gamma\sigma(\mathbf{X}\boldsymbol{\beta}^{k+1} + \mathbf{z}^{k+1} - \mathbf{y})$.

- $\eta \geq \Lambda_{\max}\left(\mathrm{X}^\top \mathrm{X}\right)$
- $\boldsymbol{\beta}$ step : $\boldsymbol{\beta}^{k+1} := \arg\min_{\boldsymbol{\beta}} \lambda\|\mathrm{w}\circ\boldsymbol{\beta}\|_1 - \left\langle\boldsymbol{\theta}^k, \mathrm{X}\boldsymbol{\beta}\right\rangle + \frac{\sigma}{2}\left\|\mathrm{X}\boldsymbol{\beta} + \mathrm{z}^k - \mathrm{y}\right\|_2^2$

scdADMM Algorithm

Algorithm 2 scdADMM – Sparse coordinate descent ADMM algorithm for solving the weighted L_1-penalized quantile regression with coordinate descent steps.

1. Initialize the algorithm with $(\boldsymbol{\beta}^0, \mathbf{z}^0, \boldsymbol{\theta}^0)$.
2. For $k = 0, 1, 2, \ldots,$ repeat Steps (2.1)–(2.3) until the convergence criterion is met.

(2.1) Carry out the coordinate descent Steps (2.1.1)–(2.1.3).

(2.1.1) Initialize $\boldsymbol{\beta}^{k,0} = \boldsymbol{\beta}^k$.

(2.1.2) For $m = 0, 1, 2, \ldots,$ repeat Step (2.1.2.1) until convergence.

(2.1.2.1) For $j = 1, \ldots, p$, update

$$\beta_j^{k,m+1} \leftarrow \frac{\text{Shrink}\left[\sum_{i=1}^n x_{ij}\left\{\theta_i^k + \sigma\left(y_i - z_i^k - \sum_{t\neq j} x_{it}\beta_t^{k,m+I(t<j)}\right)\right\}, \lambda w_j\right]}{\sigma\|X_j\|_2^2}.$$

(2.1.3) Set $\boldsymbol{\beta}^{k+1} \leftarrow \boldsymbol{\beta}^{k,m+1}$.

(2.2) Update $\mathbf{z}^{k+1} \leftarrow \left(\text{Prox}_{\rho_\tau}\left[y_i - \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^{k+1} + \sigma^{-1}\theta_i^k, n\sigma\right]\right)_{1\le i\le n}$.

(2.3) Update $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k - \sigma\left(\mathbf{X}\boldsymbol{\beta}^{k+1} + \mathbf{z}^{k+1} - \mathbf{y}\right)$.

Convergence Theory

Theorem 1

For given $\lambda > 0, \sigma > 0, 0 < \tau < 1, 0 < \gamma < (\sqrt{(}5) + 1)/2$ and a component-wisely nonnegative weight vector w, let $\{(\beta^k, z^k, \theta^k)\}$ be generated by the pADMM algorithm as described in Algorithm 1. Then, the sequence $\{(\beta^k, z^k), k = 0, 1, 2, \ldots\}$ converges to an optimal solution (β^*, z^*) to (2) and $\{\theta^k, k = 0, 1, 2, \ldots\}$ converges to an optimal soluiton θ^* to the dual problem of (2). Equivalently, $\{\beta^k, k = 0, 1, 2, \ldots\}$ converges to a global minimizer of problem (1).

Note that the convergence of the scdADMM algorithm (Algorithm 2) can be directly obtained from Boyd et al. (2011).

Implementation Details

- R package: `FHDQR`
- λ sequence: $\lambda_k = \lambda_{\max}^{\frac{M-k}{M-1}} \lambda_{\min}^{\frac{k-1}{M-1}}, k = 1, \ldots, M$, $\lambda_{\min} = \delta \lambda_{\max}$
- Implementation skill: warm-start
- Stop criterion:

$$\left\| \mathrm{X}\boldsymbol{\beta}^k + \mathrm{z}^k - \mathrm{y} \right\|_2 \leq \sqrt{n}\epsilon_1 + \epsilon_2 \max\left\{ \left\| \mathrm{X}\boldsymbol{\beta}^k \right\|_2, \left\| \mathrm{z}^k \right\|_2, \|\mathrm{y}\|_2 \right\},$$

$$\sigma \left\| \mathrm{X}^{\mathrm{T}} \left(\mathrm{z}^k - \mathrm{z}^{k-1} \right) \right\|_2 \leq \sqrt{p}\epsilon_1 + \epsilon_2 \left\| \mathrm{X}^{\mathrm{T}} \boldsymbol{\theta}^k \right\|_2,$$

Data generating models

- Example 1:

$$Y = \sum_{j=1}^{p} X_j \beta_j + k \cdot \varepsilon$$

  - $(X_1, \ldots, X_p)^{\mathrm{T}} \sim N(0, \Sigma)$ with $\Sigma = (\alpha + (1-\alpha) I(i=j))_{p \times p}$
  - $\alpha \in \{0, 0.1, 0.2, 0.5, 0.9, 0.95\}$
  - $\beta_j = (-1)^j \exp(-(2j-1)/20)$
  - $\varepsilon \sim N(0,1)$
  - $k = 3$
  - $n = 100$, $p = 1000$

- Example 2:

$$y_i = \mathrm{x}_i^{\mathrm{T}} \boldsymbol{\beta}^* + \varepsilon_i, \quad \mathrm{x}_i \sim N(0, \Sigma_{\mathrm{x}}), \quad i = 1, \ldots, n$$

  - $\boldsymbol{\beta}^* = \left(2, 0, 1.5, 0, 0.8, 0, 0, 1, 0, 1.75, 0, 0, 0.75, 0, 0, 0.3, 0_{p-16}^{\mathrm{T}}\right)^{\mathrm{T}}$
  - n=200, p=1000

Simulation setting

- Timing Comparisions:
  - Penalty function: lasso
  - Solvers: `FHDQR`, `hqreg`, `quantreg`
- Finite-Sample Performance:
  - Penalized quantile regression vs. penalized least squares regression
  - Penalty function: lasso, adaptive lasso and SCAD

Example 1: Computation time

	Correlation (α)					
	0.00	0.10	0.20	0.50	0.90	0.95
			$\tau = 0.25$			
quantreg	300+	300+	300+	300+	300+	300+
hqreg	9.52	9.32	9.82	9.86	7.05	5.63
pADMM	0.57	4.38	5.85	12.14	18.43	11.62
scdADMM	1.41	1.37	1.36	1.23	1.00	0.94
			$\tau = 0.50$			
quantreg	300+	300+	300+	300+	300+	300+
hqreg	6.88	6.64	6.89	7.92	8.65	5.29
pADMM	0.62	5.36	7.85	15.47	30.34	21.66
scdADMM	1.26	1.20	1.26	1.18	1.19	0.91
			$\tau = 0.75$			
quantreg	300+	300+	300+	300+	300+	300+
hqreg	8.65	8.34	8.81	8.87	8.37	5.76
pADMM	0.55	4.45	6.26	12.12	21.49	16.40
scdADMM	1.42	1.39	1.48	1.34	1.15	1.20

- Different algorithms all yield the same (numerically speaking) objective funtion values.

Example 2: Computation time($\tau = 0.5$)

Covariance	Method	Error			
		$N(0,2)$	Laplace	$\sqrt{2} \times t_4$	Cauchy
$\Sigma_x = I$	quantreg	400+	400+	400+	400+
	hqreg	10.45	10.45	13.95	32.96
	scdADMM	3.03	2.88	3.39	5.88
	pADMM	1.52	1.47	1.46	0.46
$AR_{0.5}$	quantreg	400+	400+	400+	400+
	hareg	11.19	10.48	14.11	24.56
	scdADMM	3.76	3.47	4.11	5.85
	pADMM	1.83	1.76	1.77	0.55
$CS_{0.5}$	quantreg	400+	400+	400+	400+
	hqreg	13.70	10.11	14.32	15.04
	scdADMM	7.11	6.77	7.68	8.49
	pADMM	19.91	17.96	20.12	5.65

- $AR_{0.5} : \Sigma_x = \left(0.5^{|i-j|}\right)$
- $CS_{0.5} : \Sigma_x = (0.5 + (1 - 0.5)I(i = j))$

Example 2: Finite-Sample Performance(L_1, L_2 losses)

		$\Sigma = (0.5 + 0.5I(i = j))$		
		Lasso	Alasso	SCAD
$N(0, 2)$	LS	4.229, 0.959	1.717, 0.610	1.286, 0.570
	QR(0.50)	5.744, 1.177	1.710, 0.718	1.681, 0.674
	QR(0.75)	5.630, 1.236	2.016, 0.835	2.170, 0.824
Laplace	LS	4.217, 0.973	1.753, 0.620	1.294, 0.578
	QR(0.50)	4.120, 0.887	1.110, 0.494	0.991, 0.408
	QR(0.75)	5.207, 1.103	1.667, 0.700	1.609, 0.666
$\sqrt{2} \times t_4$	LS	6.027, 1.342	3.706, 1.114	2.265, 0.975
	QR(0.50)	5.941, 1.276	2.045, 0.842	1.900, 0.780
	QR (0.75)	6.485, 1.456	2.819, 1.090	2.959, 1.112
Cauchy	LS	20.931, 4.650	32.018, 8.487	472.891, 46.890
	QR(0.50)	5.905, 1.324	2.497, 0.978	2.257, 0.868
	QR(0.75)	8.193, 1.904	4.563, 1.598	4.904, 1.684

A Real Data Example

- Goal: study how the expression of *TRIM32* depends on the expressions of 3000 genes.
- n=120, p=3000
- Timing results:

τ	0.25	0.50	0.75
quantreg	5000+	5000+	5000+
hqreg	4.97	4.09	4.56
pADMM	351.93	401.76	347.89
scdADMM	1.68	1.15	1.09

A Real Data Example

- Prediction error $=(1/40)\sum_{i\in \text{ validation}} \rho_\tau\left(y_i - \hat{\beta}_0 - \mathrm{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}\right)$
- 50 random partitions
- All the results are averaged over 50 runs for the random partition columns.

τ	All data #genes	Random partition	
		Ave. #genes	Prediction error
0.25	14	15.00(1.26)	0.0351(0.0014)
0.50	23	24.16(2.38)	0.0395(0.0010)
0.75	14	11.22(1.07)	0.0671(0.0196)

Thank you