Supplementary Materials to "A Parallel Algorithm for Large-scale Nonconvex Penalized Quantile Regression"

Liqun Yu*, Nan Lin* and Lan Wang †

In the following supplementary mat

1 Derivation of the QPADM updates

We first demonstrate that problem (9) follows the standard ADMM form (3). Define $A = [A_1 \ A_2]$ with

$$A_1 = -\begin{bmatrix} I_p & \dots & I_p & 0_{p\times n}\end{bmatrix}^T \in \mathbb{R}^{(Mp+n)\times p},$$

$$A_2 = \begin{bmatrix} 0_{n\times Mp} & -I_n\end{bmatrix}^T \in \mathbb{R}^{(Mp+n)\times n},$$

and define

$$B = \begin{bmatrix} I_p & & & -X_1^T & & \\ & \ddots & & & \ddots & \\ & & I_p & & & -X_M^T \end{bmatrix}^T \in \mathbb{R}^{(Mp+n)\times Mp},$$

and

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{y} \end{bmatrix} \in \mathbb{R}^{Mp+n}, \ \boldsymbol{x} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{r} \end{pmatrix}, \ \boldsymbol{z} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_M \end{pmatrix},$$

*Liqun Yu is a graduate student and Nan Lin is Associate Professor, Department of Mathematics, Washington University in St. Louis, St. Louis, MO 63130.

†Lan Wang is Professor, School of Statistics, University of Minnesota, Minneapolis, MN 55455.

Then

$$
A\boldsymbol{x} + B\boldsymbol{z} = \begin{bmatrix} \boldsymbol{\beta}_1 - \boldsymbol{\beta}, \\ \vdots \\ \boldsymbol{\beta}_M - \boldsymbol{\beta}, \\ \\ -\boldsymbol{r}_1 - X_1\boldsymbol{\beta}_1, \\ \vdots \\ -\boldsymbol{r}_M - X_M\boldsymbol{\beta}_M \end{bmatrix},
$$

where $(\boldsymbol{r}_1^T, \ldots, \boldsymbol{r}_M^T)^T = \boldsymbol{r}$. Then the constraints in (9) can be combined as a single constraint $A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}$. As a consequence, problem (9) can be written exactly as (3) with $f(\boldsymbol{x}) = \rho_\tau(\boldsymbol{r}) + P_\lambda(\boldsymbol{\beta})$ and $g(\boldsymbol{z}) = 0$.

Next, we show that the standard ADMM updates as in (4) applied to (9) results in the updates (10) in the main article. Applying (4) to (9) with A, B, $\boldsymbol{c}$, $\boldsymbol{x}$, $\boldsymbol{z}$ defined above, and separating the dual variable $\boldsymbol{u}$ corresponding to the constraint $A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}$ as

$$
\boldsymbol{u} = (\boldsymbol{\eta}_1^T, \ldots, \boldsymbol{\eta}_M^T,\ \boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_M^T)^T,
$$

where each $\boldsymbol{\eta}_b$ corresponds to the constraint $\boldsymbol{\beta}_b - \boldsymbol{\beta}$ and each $\boldsymbol{u}_b$ corresponds to the constraint $-\boldsymbol{r}_b - X_b\boldsymbol{\beta}_b = -\boldsymbol{y}_b$, we have

$$
\begin{aligned}
\begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{r} \end{pmatrix}^{k+1} &:= \arg\min_{(\boldsymbol{\beta},\boldsymbol{r})} \sum_{b=1}^{M} \rho_\tau(\boldsymbol{r}_b) + P_\lambda(\boldsymbol{\beta}) + \sum_{b=1}^{M} (\boldsymbol{\eta}_b^{k+1})^T(\boldsymbol{\beta}_b^{k+1} - \boldsymbol{\beta}) + \sum_{b=1}^{M} (\boldsymbol{u}_b^{k+1})^T(\boldsymbol{y}_b - X_b\boldsymbol{\beta}_b^{k+1} - \boldsymbol{r}_b) \\
&\qquad + \sum_{b=1}^{M} \frac{\gamma}{2}\|\boldsymbol{\beta}_b^{k+1} - \boldsymbol{\beta}\|_2^2 + \sum_{b=1}^{M} \frac{\gamma}{2}\|\boldsymbol{y}_b - X_b\boldsymbol{\beta}_b^{k+1} - \boldsymbol{r}_b\|_2^2, \\
\begin{pmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_M \end{pmatrix}^{k+1} &:= \arg\min_{(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_M)} \sum_{b=1}^{M} \|\boldsymbol{\beta}_b - \boldsymbol{\beta}^{k+1} + \boldsymbol{\eta}^k/\gamma\|_2^2 + \sum_{b=1}^{M} \|\boldsymbol{y}_b - X_b\boldsymbol{\beta}_b - \boldsymbol{r}_b^{k+1} - \boldsymbol{u}_b^k/\gamma\|_2^2, \\
\begin{pmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_M \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_M \end{pmatrix}^{k+1} &:= \begin{pmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_M \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_M \end{pmatrix}^{k} + \gamma \begin{pmatrix} \boldsymbol{\beta}_1^{k+1} - \boldsymbol{\beta}^{k+1} \\ \vdots \\ \boldsymbol{\beta}_M^{k+1} - \boldsymbol{\beta}^{k+1} \\ \boldsymbol{y}_1 - X_1\boldsymbol{\beta}_1^{k+1} - \boldsymbol{r}_1^{k+1} \\ \vdots \\ \boldsymbol{y}_M - X_M\boldsymbol{\beta}_M^{k+1} - \boldsymbol{r}_M^{k+1} \end{pmatrix}.
\end{aligned}
$$

Notice that the $\boldsymbol{\beta}$-update and the $\boldsymbol{r}_b$-updates, and the M $\boldsymbol{\beta}_b$-updates are all separable. Then the updating rules (10) in the main article follow immediately. The closed-form solutions of the $\boldsymbol{\beta}_b$-updates in (10) in the main article follow from the quadratic form of the objective functions as seen in the second update above.

2 Additional simulation results

2.1 Simulation results for the MCP penalty with $M = 1$

In Tables 1-3 of the main article, the performance of the QPADM for model (16) with SCAD penalty were shown. In the following, we show the performance of the QPADM with the MCP penalty.

Method	Quantile	Size	P1	P2	AE	Time (Sec)
QPADM	$\tau = 0.3$	5.80(1.56)	100%	94%	0.048(0.023)	1.65(0.31)
	$\tau = 0.5$	4.31(0.64)	100%	0%	0.036(0.022)	1.54(0.29)
	$\tau = 0.7$	6.80(1.42)	100%	93%	0.043(0.024)	1.67(0.33)
QICD	$\tau = 0.3$	7.56(3.82)	100%	92%	0.050(0.026)	0.99(1.13)
	$\tau = 0.5$	4.24(0.59)	100%	0%	0.040(0.020)	1.51(1.30)
	$\tau = 0.7$	6.80(3.62)	100%	93%	0.049(0.026)	1.46(1.59)

Table 4: Comparison of QPADM and QICD with $n = 300,\ p = 1,000$.

Method	Quantile	Size	P1	P2	AE	Time (Sec)
QPADM	$\tau = 0.3$	5.00(0.00)	100%	100%	0.0040(0.0016)	45.09(1.55)
	$\tau = 0.5$	4.00(0.00)	100%	0%	0.0042(0.0019)	47.16(1.68)
	$\tau = 0.7$	5.00(0.00)	100%	100%	0.0037(0.0017)	44.81(1.57)
QICD	$\tau = 0.3$	5.02(0.14)	100%	100%	0.0031(0.0016)	99.37(11.46)
	$\tau = 0.5$	4.16(0.37)	100%	0%	0.0033(0.0015)	121.47(16.35)
	$\tau = 0.7$	5.08(0.25)	100%	100%	0.0032(0.0014)	118.35(16.17)

Table 5: Comparison of QPADM and QICD with $n = 30,000,\ p = 1,000$.

Method	Quantile	Size	P1	P2	AE	Time (Sec)
QPADM	$\tau = 0.3$	5.00(0.00)	100%	100%	0.0032(0.0011)	3.43(0.56)
	$\tau = 0.5$	4.00(0.00)	100%	0%	0.0031(0.0011)	3.54(0.67)
	$\tau = 0.7$	5.00(0.00)	100%	100%	0.0030(0.0015)	3.42(0.58)
QICD	$\tau = 0.3$	5.06(0.24)	100%	100%	0.0027(0.0011)	12.21(3.33)
	$\tau = 0.5$	4.08(0.27)	100%	0%	0.0026(0.0011)	22.33(11.71)
	$\tau = 0.7$	5.02(0.15)	100%	100%	0.0026(0.0009)	25.45(19.00)

Table 6: Comparison of QPADM and QICD with $n = 30000,\ p = 100$.

3 Parallel QPADM: more results

The main article only showed the performance of parallel QPADM for the SCAD penalty with quantile level $\tau = 0.3$, while the simulation were done with $\tau =$

0.3, 0.5, 0.7 for both the SCAD and MCP penalties. We include the remaining results in Figures 2 - 6.

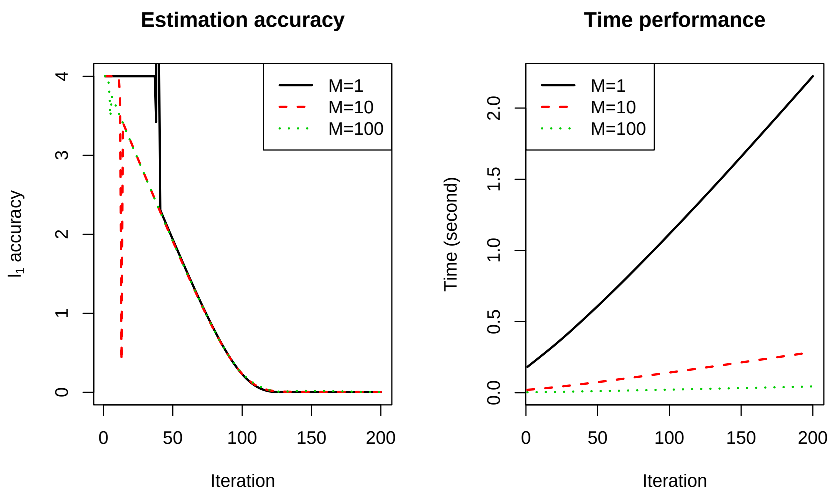

Figure 2: Comparison of QPADM with SCAD penalty for different M values at τ=0.5.

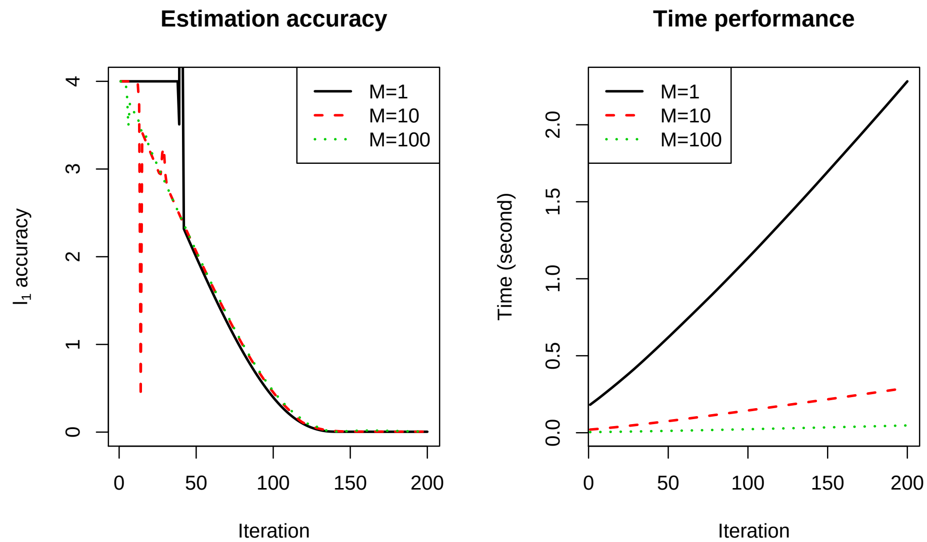

Figure 3: Comparison of QPADM with SCAD penalty for different M values at τ=0.7.

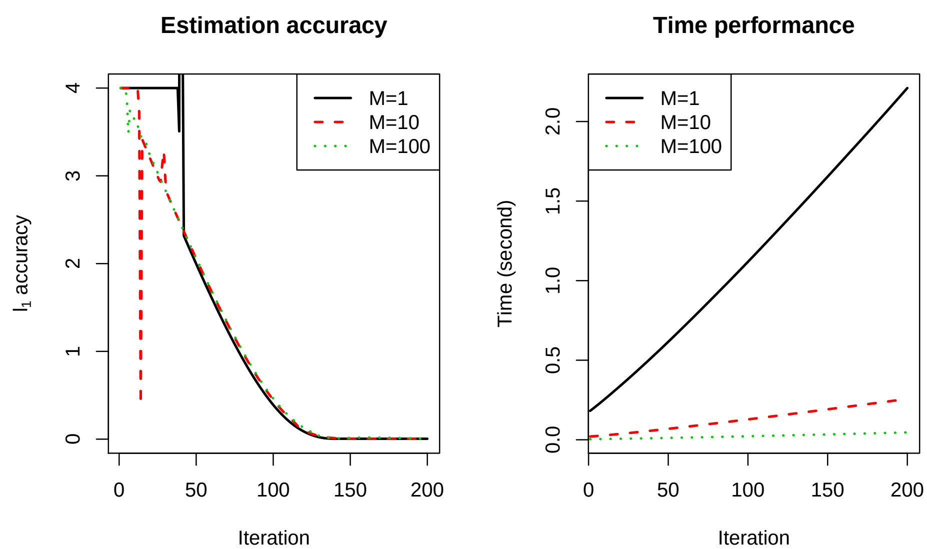

Figure 4: Comparison of QPADM with MCP penalty for different M values at τ=0.3.

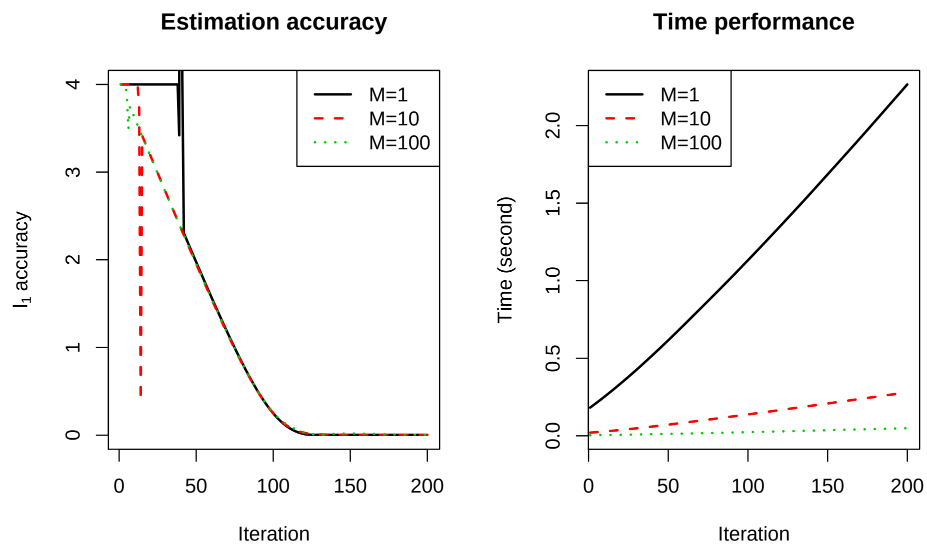

Figure 5: Comparison of QPADM with MCP penalty for different M values at τ=0.5.

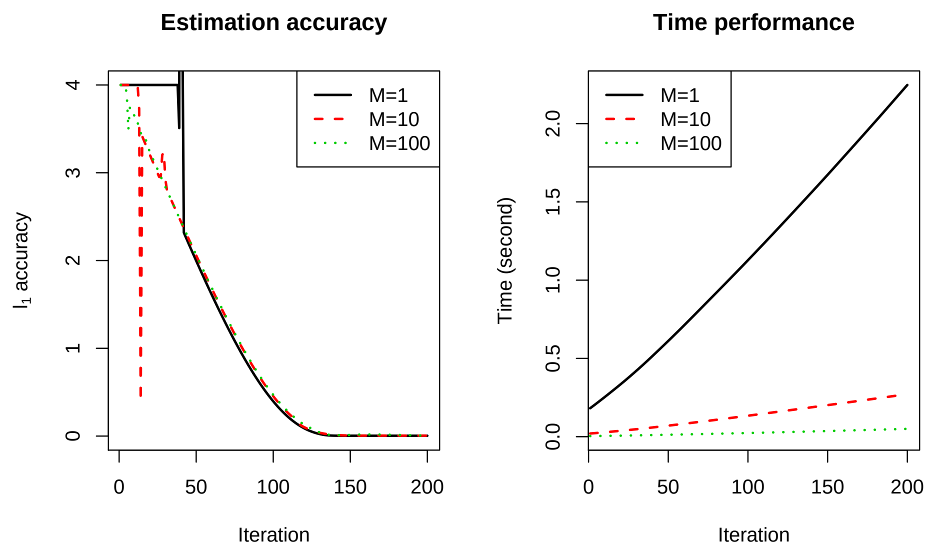

Figure 6: Comparison of QPADM with MCP penalty for different M values at τ=0.7.