Distributed Quantile Regression with Non-Convex Sparse Penalties

Reza Mirzaeifard*, Vinay Chakravarthi Gogineni*, Naveen K. D. Venkategowda§, Stefan Werner*
*Dept. of Electronic Systems, Norwegian University of Science and Technology-NTNU, Norway
§Department of Science and Technology, Linköping University, Sweden
E-mails: {reza.mirzaeifard, vinay.gogineni, stefan.werner}@ntnu.no, naveen.venkategowda@liu.se

***Abstract*—The surge in data generated by IoT sensors has increased the need for scalable and efficient data analysis methods, particularly for robust algorithms like quantile regression, which can be tailored to meet a variety of situations, including nonlinear relationships, distributions with heavy tails, and outliers. This paper presents a sub-gradient-based algorithm for distributed quantile regression with non-convex, and non-smooth sparse penalties such as the Minimax Concave Penalty (MCP) and Smoothly Clipped Absolute Deviation (SCAD). These penalties selectively shrink non-active coefficients towards zero, addressing the limitations of traditional penalties like the l_1-penalty in sparse models. Existing quantile regression algorithms with non-convex penalties are designed for centralized cases, whereas our proposed method can be applied to distributed quantile regression using non-convex penalties, thereby improving estimation accuracy. We provide a convergence proof for our proposed algorithm and demonstrate through numerical simulations that it outperforms state-of-the-art algorithms in sparse and moderately sparse scenarios.**

***Index Terms*—Distributed learning, quantile regression, non-convex and non-smooth penalties, weak convexity, sparse learning**

I. Introduction

Internet-of-things (IoT) and cyber-physical systems incorporate distributed devices and sensors that collect data to provide inference and decision-making capabilities. It is necessary to develop distributed solutions in such systems that do not require data transfer to a central hub. Distributed methods handle resource constraints as well as concerns such as computational power, battery power, communication bandwidth, and privacy [1]–[3].

In IoT systems, regression is a common task that involves estimating the relationship between one or more predictor variables and a response variable [4]. Regression algorithms typically estimate the conditional mean of the response variable associated with a set of observations using a mean regression technique [5]. However, the mean regression method is sensitive to outliers and cannot relate the response variable to another point in the conditional distribution, such as the median or a specific percentile. Alternatively, quantile regression, which describes regression relationships on the basis of quantiles, can be used to alleviate the sensitivity of mean-based regression [6]. It has therefore been found useful in various applications, such as estimating uncertainty in data from smart electricity meters [7] and forecasting load on smart grids [8].

In many real-world applications, the models to be estimated tend to be sparse, such as quantitative traits in genetics [9], gene selection for microarray gene expression [10], and more. Using the *a priori* information about sparsity may yield better results than the conventional quantile regression method. Therefore, sparse-penalized quantile regression has received considerable research interest [11], [12]. Using a l_1-penalty in a quantile regression model produces more accurate results when the model is highly sparse, but fails when the model is moderately or non-sparse since the l_1-penalty shrinks all coefficients uniformly. Consequently, l_1-penalized quantile regression provides poor performance and bias with reducing model sparsity. One solution to this problem is using sparse penalties, such as the minimax concave penalty (MCP) [13] and the smoothly clipped absolute deviation (SCAD) [14], that are capable of intelligently distinguishing between active and inactive coefficients. Although these penalties encourage sparse solutions, they mitigate the bias effect of the l_1-penalty [15], [16].

Solving l_1-penalized quantile regression problems has conventionally relied on linear programming algorithms [17], [18]. However, these linear programming algorithms are limited in the sense that there is no guarantee of convergence or the quality of the solution. Therefore, a new approach was introduced in [19] that uses a local linear approximation (LLA) algorithm as part of a general framework to solve folded concave penalized regression. This approach approximates the non-convex penalty function with a piece-wise linear function and solves the resulting optimization problem iteratively. Alternatively, the iterative coordinate descent algorithm (QICD), was proposed in [20] for similar problems, with established convergence and faster convergence rates than LLA. However, both the LLA and QICD algorithms can be computationally intensive and have slow convergence rates as they suffer from inner loop. To tackle this issue, a more efficient single-loop ADMM algorithm was proposed in [21]. Nevertheless, all the algorithms mentioned above are limited to centralized quantile regression, and there has been little research on using MCP or SCAD penalties in distributed quantile regression. Furthermore, it is unclear whether existing distributed algorithms for non-convex non-smooth problems, e.g., [22]–[27], can be directly applied to the quantile regression context. Thus, more

This work was supported by the Research Council of Norway.

research is needed to investigate these questions.

This paper introduces a new distributed sub-gradient-based algorithm to tackle the distributed quantile regression problem when faced with non-convex and non-smooth sparse penalties, such as MCP and SCAD. In spite of the fact that MCP and SCAD are non-convex and non-smooth, their weak convexity allows us to demonstrate the convergence of the algorithm under mild conditions, including connected and doubly stochastic networks, and non-summable and diminishing step sizes. To validate our theoretical findings, we conducted several numerical simulations. The simulation results confirm that our proposed algorithm outperforms the state-of-the-art techniques, distributed quantile regression (dQR) [28] and its l_1-penalized variant (l_1-dQR) [28], in terms of accuracy for sparse and moderately sparse settings.

Mathematical Notations: Lowercase letters represent scalars, bold lowercase letters represent column vectors, and bold uppercase letters represent matrices. The transpose of a matrix is represented by $(\cdot)^{\mathrm{T}}$, and the jth column of a matrix $\mathbf{A}$ is denoted as $\mathbf{a}_j$. Additionally, the entry in the ith row and jth column of $\mathbf{A}$ is denoted as a_{ij}. Lastly, $\partial f(u)$ represents the sub-gradient of the function $f(\cdot)$ evaluated at u.

II. Preliminaries

A brief introduction to quantile regression is provided in this section. Considering a scalar random variable Y, a P-dimensional vector of covariates $\boldsymbol{\chi}$, and $F_Y(y|\mathbf{x}) = P(Y \leq y|\boldsymbol{\chi} = \mathbf{x})$ as the conditional cumulative distribution function, the conditional quantile τ is denoted as follows:

$$Q_Y(\tau|\mathbf{x}) = \inf\{y : F_Y(y|\mathbf{x}) \geq \tau\}, \tag{1}$$

for $\tau \in (0, 1)$. In the quantile regression model, $Q_Y(\tau|\mathbf{x})$ is assumed to be linearly related to $\mathbf{x}$ as follows:

$$Q_Y(\tau|\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_\tau + q_\tau^\epsilon, \tag{2}$$

where $\boldsymbol{\beta}_\tau \in \mathbb{R}^P$ is the coefficient vector, $q_\tau^\epsilon \in \mathbb{R}$ is the τth quantile of the noise, and both are unknown and must be estimated. The optimization problem for $\boldsymbol{\beta}_\tau$ and q_τ^ϵ can be formulated as [29]:

$$\{\boldsymbol{\beta}_\tau, q_\tau^\epsilon\} = \underset{\boldsymbol{\beta}_\tau, q_\tau^\epsilon}{\arg\min}\, \mathbb{E}[\rho_\tau(y - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_\tau - q_\tau^\epsilon)], \tag{3}$$

where $\rho_\tau(\mathbf{z}) = ||\mathbf{z}||_1 - (2\tau - 1)\mathbf{1}^{\mathrm{T}}\mathbf{z}$ which is also known as check loss function. Given a data set of measurements and observations $(\mathbf{x}_j, y_j)_{j=1}^m$, where j is the sample index, and the selected value of τ, the parameters $\boldsymbol{\beta}_\tau$ and q_τ^ϵ can be estimated by optimizing the following problem [29]:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\arg\min}\, \frac{1}{m}\sum_{j=1}^{m} \rho_\tau(y_j - \bar{\mathbf{x}}_j^{\mathrm{T}}\mathbf{w}), \tag{4}$$

where m is the number of samples, $\bar{\mathbf{x}}_j = [\mathbf{x}_j^{\mathrm{T}}, 1]^{\mathrm{T}} \in \mathbb{R}^{P+1}$, and $\mathbf{w} = [\boldsymbol{\beta}_\tau^{\mathrm{T}}, q_\tau^\epsilon]^{\mathrm{T}} \in \mathbb{R}^{P+1}$.

One way to improve inference quality in quantile regression is to utilize *a priori* information about the model coefficients. This can be achieved by adding a penalty function, $P_{\lambda,\gamma}(\mathbf{w})$, to the quantile regression loss function. The optimization problem (4) takes a new form after penalizing the loss function as [21]:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\arg\min}\, \frac{1}{m}\sum_{j=1}^{m} \rho_\tau(y_j - \bar{\mathbf{x}}_j^{\mathrm{T}}\mathbf{w}) + P_{\lambda,\gamma}(\mathbf{w}). \tag{5}$$

When it comes to promoting sparsity, there are several regulation functions to choose from, but the LASSO function has gained widespread popularity. Despite its popularity, the LASSO function can lead to estimation bias and is not well-suited for group sparsity. In this paper, we present a solution that utilizes the MCP and SCAD penalty functions as $P_{\lambda,\gamma}(\mathbf{w}) = \sum_{p=1}^{P} g_{\lambda,\gamma}(w_p)$ to achieve sparsity. Due to the definitions of MCP [13] and SCAD [14] with constraints $\gamma \geq 1$ and $\gamma \geq 2$ respectively, which are given by:

$$g_{\lambda,\gamma}^{\mathrm{MCP}}(w_p) = \begin{cases} \lambda|w_p| - \frac{w_p^2}{2\gamma}, & |w_p| \leq \gamma\lambda \\ \frac{\gamma\lambda^2}{2}, & |w_p| > \gamma\lambda \end{cases} \tag{6}$$

and

$$g_{\lambda,\gamma}^{\mathrm{SCAD}}(w_p) = \begin{cases} \lambda|w_p|, & |w_p| \leq \lambda \\ -\frac{|w_p|^2 - 2a\lambda|w_p| + \lambda^2}{2(\gamma-1)}, & \lambda < |w_p| \leq \gamma\lambda \\ \frac{(\gamma+1)\lambda^2}{2}, & |w_p| > \gamma\lambda \end{cases} \tag{7}$$

these non-convex and non-smooth functions can clearly differentiate between active and non-active coefficients. Additionally, the MCP and SCAD functions are known to be weakly convex for $\rho \geq \frac{1}{\gamma}$ and $\rho \geq \frac{1}{\gamma-1}$ respectively, according to [30]. This final property is useful in demonstrating the convergence of the sub-gradient algorithm with these penalty functions [22].

III. Distributed penalized quantile regression

Let us consider a network with L agents modeled as an undirected graph $\mathcal{G}$ consisting of vertices $\mathcal{V} = \{1, \cdots, L\}$ and bidirectional communication links represented by the edge set $\mathcal{E}$. Each agent $i \in \mathcal{V}$ can communicate with those in its neighborhood $\mathcal{N}_i$ of cardinality $|\mathcal{N}_i|$. The network's weighting matrix $\mathbf{C}$ is a $L \times L$ dimensional matrix, in which each entry c_{ij} quantifies the weight assigned by node i to information received from node j. We make the following assumptions about the connectivity of the graph and the weight matrix $\mathbf{C}$:

Assumption 1. *The graph $\mathcal{G}$ is strongly connected, meaning that all agents have direct or indirect connections to one another.*

Assumption 2. *The weight matrix $\mathbf{C}$ is doubly stochastic, i.e., $\sum_j c_{i,j} = \sum_j c_{j,i} = 1$ for all i.*

Let $\mathbf{X}_i = [\bar{\mathbf{x}}_{i,1}^{\mathrm{T}}, \cdots, \bar{\mathbf{x}}_{i,M_i}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{M_i \times (P+1)}$ denote the observation matrix at agent i and $\mathbf{y}_i = [y_{i,1}, \cdots, y_{i,M_i}]^{\mathrm{T}} \in \mathbb{R}^{M_i}$ the corresponding response vector, where M_i is the number of measurements at agent i and $\sum_{i=1}^{L} M_i = m$. The parameter vector $\mathbf{w} \in \mathbb{R}^{P+1}$, consisting of $\boldsymbol{\beta}_\tau$ and q_τ^ϵ, models the linear relationship between all measurements and observations, represented by $\mathbf{X} = [\mathbf{X}_1^{\mathrm{T}}, \cdots, \mathbf{X}_L^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{y} = [\mathbf{y}_1^{\mathrm{T}}, \cdots, \mathbf{y}_L^{\mathrm{T}}]^{\mathrm{T}}$

respectively. The penalized quantile regression estimate of $\mathbf{w}$ is obtained by solving

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{1}{m}\sum_{i=1}^{L}\sum_{j=1}^{M_i}\rho_\tau(y_{i,j} - \bar{\mathbf{x}}_{i,j}^{\mathrm{T}}\mathbf{w}) + P_{\lambda,\gamma}(\mathbf{w}). \quad (8)$$

This optimization problem can be expressed as the sum of local objective functions, each defined as $f_i(\mathbf{w}) = \frac{1}{m}\sum_{j=1}^{M_i}\rho_\tau(y_{i,j} - \bar{\mathbf{x}}_{i,j}^{\mathrm{T}}\mathbf{w}) + \frac{1}{L}P_{\lambda,\gamma}(\mathbf{w})$, leading to the following global optimization problem:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{L} f_i(\mathbf{w}). \quad (9)$$

The proposed algorithm for quantile regression in a distributed setting, called distributed sub-gradient method for penalized quantile regression (DSPQ), updates each node's estimate of the actual parameter $\mathbf{w}^*$ through a diffusion step and a sub-gradient update step. Taking $\mathbf{v}_i$ as the diffusion variable for node i, at each time instant k, diffusion occurs as follows:

$$\mathbf{v}_i^{(k)} := \sum_{i=1}^{L} c_{i,j}\mathbf{w}_j^{(k)}, \quad (10)$$

where each node updates its estimate by making a linear combination of its own generated estimate and the estimates received from its neighboring nodes. This update is performed through a sub-gradient update step described by the following equation:

$$\mathbf{w}_i^{(k+1)} = \mathbf{v}_i^{(k)} - \alpha^{(k)}\mathbf{g}_i^{(k)}, \quad (11)$$

where $\alpha^{(k)}$ is step-size at iteration k and $\mathbf{g}_i^{(k)} \in \partial f_i(\mathbf{v}_i^{(k)})$ is any element of the sub-differential set of $f_i(\cdot)$. It is important to note that each function $f_i(\cdot)$ is neither convex nor smooth, and thus, the distributed sub-gradient method [22] is adopted for iteratively computing the estimate of $\mathbf{w}$ at each node.

The sub-gradient of the local penalized quantile regression loss function $f_i(\cdot)$ with respect to the coefficient $\mathbf{v}_i^{(k)}$ is computed as follows:

$$\partial\mathbf{g}_i^{(k)} = \frac{1}{m}\sum_{j=1}^{M_i}\partial\rho_\tau(y_{i,j} - \bar{\mathbf{x}}_{i,j}^{\mathrm{T}}\mathbf{v}_i^{(k)}) + \frac{1}{L}\partial P_{\lambda,\gamma}(\mathbf{v}_i^{(k)}). \quad (12)$$

Here, $\partial\rho_\tau(y_{i,j} - \bar{\mathbf{x}}_{i,j}^{\mathrm{T}}\mathbf{v}_i^{(k)})$ represents the sub-gradient of the check loss function with respect to $\mathbf{v}_i^{(k)}$ when the measurement $\bar{\mathbf{x}}_{i,j}, y_{i,j}$ is available. This function is obtained as:

$$\partial\rho_\tau(y_{i,j} - \bar{\mathbf{x}}_{i,j}^{\mathrm{T}}\mathbf{v}_i^{(k)}) = \begin{cases} -\tau\bar{\mathbf{x}}_{i,j}, & y_{i,j} - \bar{\mathbf{x}}_{i,j}^{\mathrm{T}}\mathbf{v}_i^{(k)} \geq 0, \\ (1-\tau)\bar{\mathbf{x}}_{i,j}, & y_{i,j} - \bar{\mathbf{x}}_{i,j}^{\mathrm{T}}\mathbf{v}_i^{(k)} < 0, \\ \mathbf{0}_{P+1}, & y_{i,j} - \bar{\mathbf{x}}_{i,j}^{\mathrm{T}}\mathbf{v}_i^{(k)} = 0. \end{cases} \quad (13)$$

Additionally, $\partial P_{\lambda,\gamma}(\mathbf{v}_i^{(k)})$ represents the sub-gradient of the penalty function with respect to $\mathbf{v}_i^{(k)}$, which can be derived as:

$$\partial P_{\lambda,\gamma}(\mathbf{v}_i^{(k)}) = [\partial g_{\lambda,\gamma}(v_{i,1}^{(k)}), \cdots, \partial g_{\lambda,\gamma}(v_{i,P}^{(k)}), 0]^{\mathrm{T}}. \quad (14)$$

Algorithm 1: Distributed Sub-gradient Method for Penalized Quantile Regression (DSPQ)

```
Initialize $\mathbf{w}_i^{(k)}(1)$ for each node $i$, the step-size $\alpha_1$,
  and $\gamma$ the number of iterations $K$ , the parameter $\tau$
  and the regularized parameters $\gamma$ and $\lambda$.;
for $k = 1, \cdots, K$ do
    for $i = 1, \cdots, L$ do
        Receive $\mathbf{w}_j$ from neighbors in $\mathcal{N}_i$;
        Update $\mathbf{v}_i^{(k)}$ by (10);
        The DSPQ-MCP algorithm:
            Update $\mathbf{g}_i^{(k)}$ by (12)-(15);
        The DSPQ-SCAD algorithm:
            Update $\mathbf{g}_i^{(k)}$ by (12)-(14), and (16);
        Update $\mathbf{w}_i^{(k+1)}$ by (11);
    end
end
```

For MCP the sub-gradient can be derived as [13]:

$$\partial g_{\lambda,\gamma}^{\mathrm{MCP}}(v_{i,p}) = \begin{cases} (\lambda - \frac{|v_{i,p}|}{\gamma})\mathrm{sign}(v_{i,p}), & |v_{i,p}| \leq \gamma\lambda \\ 0. & |v_{i,p}| > \gamma\lambda \end{cases} \quad (15)$$

In the same way, SCAD's subgradient can be deduced as follows [14]:

$$\partial g_{\lambda,\gamma}^{\mathrm{SCAD}}(v_{i,p}) = \begin{cases} \lambda\ \mathrm{sign}(v_{i,p}), & |v_{i,p}| \leq \lambda \\ \frac{\gamma\lambda - |v_{i,p}|}{\gamma - 1}, & \lambda < |v_{i,p}| \leq \gamma\lambda \\ 0, & |v_{i,p}| > \gamma\lambda \end{cases} \quad (16)$$

As a final requirement, the step-size $\alpha^{(k)}$ for convergence of DSPQ should meet Assumption 3, i.e., it should be non-summable and diminishing. A summary of the proposed method for solving distributed sparse penalized quantile regression can be found in Algorithm 1.

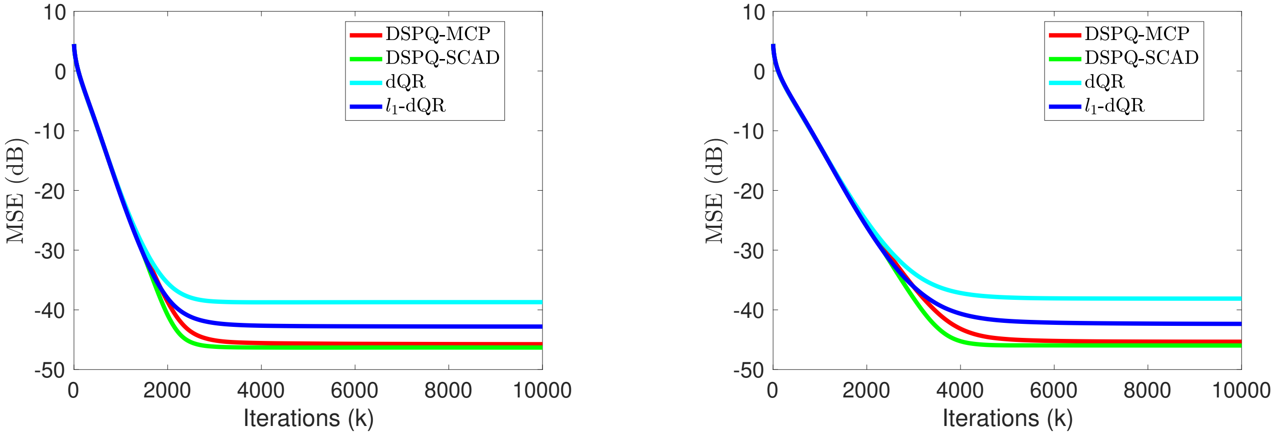
Assumption 3. *The step-size $\alpha^{(k)}$ satisfies the following constraints: $\lim_{k\to\infty}\alpha^{(k)} = 0$, $\sum_{i=1}^{\infty}\alpha^{(k)} = +\infty$, $\sum_{i=1}^{\infty}(\alpha^{(k)})^2 < +\infty$.*

Theorem 1. *Assuming the static graph $\mathcal{G}$ adheres to the connectivity assumption stated in Assumption 1, the weighted graph $\mathbf{C}$ satisfies the doubly stochasticity requirement outlined in Assumption 2, and the step-size follows the not-summable but diminishing assumption described in Assumption 3, the DSPQ algorithm will converge to a stationary point.*

Proof. The convergence of our proposed algorithm is established by ensuring that all the assumptions stated in [22, Theorem 1] are satisfied, and the objective function of the penalized quantile regression with either the MCP or SCAD penalty is weakly convex. □

IV. Simulation results

In this section, we evaluate the performance of the proposed DSPQ algorithm through simulations, comparing it to the dQR and l_1-dQR algorithms presented in [28]. The sensor network



(a) $\tau = 0.55$

(b) $\tau = 0.75$

Fig. 1: MSE versus iterations

considered consists of $L = 30$ nodes, randomly distributed over a 2.5×2.5 square area and is fully connected, with connections established between nodes if the distance between them is less than 0.8 and a minimum and maximum of 2 and 10 neighbors, respectively. The weighting matrix is generated according to the metropolis rule [31]. The measurement data, $\mathbf{x}_{i,j}$, for each node and time instance is generated from an i.i.d normal distribution $\mathcal{N}(0, \mathbf{\Sigma}_{P \times P})$ with $P = 18$, where $\Sigma_{pq} = 0.5^{|p-q|}$. We set $\gamma_{\text{SCAD}} = 3.7$, $\gamma_{\text{MCP}} = 2.4$, and $\lambda = 30 \times 1.7 \times 10^{-4}$ for penalty functions. The step-size for each iteration, $\alpha^{(k)}$, was chosen as $\frac{3.5}{k^{(0.51)}}$, ensuring it meets the requirements of Assumption 3. Moreover, the mean square error (MSE), represented by $\frac{\sum_{i=1}^{L} ||\hat{\mathbf{w}}_i - \mathbf{w}||_2^2}{L}$, serves as the performance measure and all the results were determined by averaging the results of 100 individual trials.

In the first scenario, we evaluate the accuracy of these methods using the MSE metric for two different values of τ, namely $\tau = 0.55$ and $\tau = 0.75$. We consider $y_{i,j}$ as described by the following equation:

$$y_{i,j} = \mathbf{x}_{i,j}^{\mathrm{T}} \boldsymbol{\beta} + \epsilon_{i,j}, \tag{17}$$

where $\boldsymbol{\beta}$ represents the linear parameter with $S = 3$ randomly selected active coefficients equal to one, and $\epsilon_{l,j}$ is a random noise that is generated i.i.d from $\mathcal{N}(0, 0.2)$. As a result, the quantile of error can be expressed as $q_\tau^\epsilon = 0.2 \cdot \Phi(\tau)$, where $\Phi(\cdot)$ represents the cumulative distribution function of the standard normal distribution. Fig. 1 presents the mean squared error against the number of iterations for different algorithms, plotted for $\tau = 0.55$ and $\tau = 0.75$ in the learning curves. As shown in Fig. 1, the proposed DSPQ achieves a lower MSE than other existing approaches with the same convergence rate.

In the second scenario, we evaluated the performance of the algorithms with varying levels of sparsity, as the number of active coefficients increased from 0 to P. τ was fixed at 0.75, and S coefficients were randomly set to 1 while the rest were set to 0. Fig. 2 displays the mean squared error against the number of active coefficients for each algorithm. The results demonstrate that the proposed DSPQ algorithm outperformed the others, delivering better MSE results across all sparsity levels. The dQR algorithm outperformed the l_1-dQR for $S > 12$, showing that l_1 is only effective in sufficiently sparse scenarios. These results indicate that MCP and SCAD are effective in both sparse and non-sparse models.

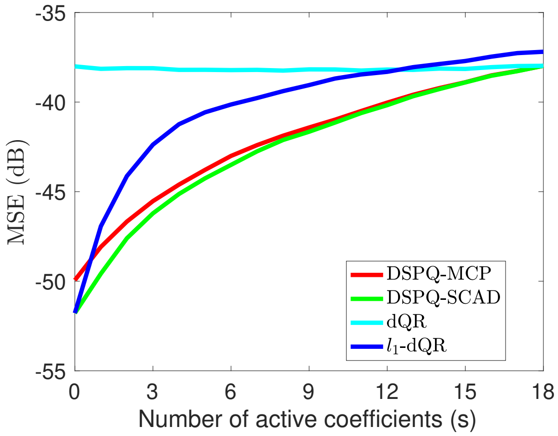


Fig. 2: Steady-state MSE versus the number of non-zero elements

V. Conclusion

This paper presented a sub-gradient algorithm for quantile regression that employed non-convex and non-smooth sparse penalties. The convergence of the proposed algorithm was rigorously established. Our simulation results demonstrated the superiority of this algorithm in terms of mean squared error compared to the conventional dQR and l_1-dQR methods. The algorithm was consistently effective, particularly in moderate and non-sparse scenarios, where other algorithms demonstrated subpar performance.

References

[1] X. Chen, W. Liu, and Y. Zhang, "Quantile regression under memory constraint," *The Annals of Statistics*, vol. 47, no. 6, pp. 3244–3273, Dec. 2019.

[2] K. M. Tan, H. Battey, and W.-X. Zhou, "Communication-constrained distributed quantile regression with optimal statistical guarantees," *Journal of Machine Learning Research*, vol. 23, pp. 1–61, Aug. 2022.

[3] X. Zhou and Y. Xiang, "ADMM-based differential privacy learning for penalized quantile regression on distributed functional data," *Mathematics*, vol. 10, no. 16, p. 2954, Aug. 2022.

[4] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and iot applications," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2017, pp. 1–8.

[5] G. A. Seber and A. J. Lee, *Linear Regression Analysis*. John Wiley & Sons, 2012, vol. 329.

[6] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of Economic Perspectives*, vol. 15, no. 4, pp. 143–156, Dec. 2001.

[7] S. Ben Taieb, R. Huser, R. J. Hyndman, and M. G. Genton, "Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2448–2455, Mar. 2016.

[8] H. Aprillia, H.-T. Yang, and C.-M. Huang, "Statistical load forecasting using optimal quantile regression random forest and risk assessment index," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1467–1480, Oct. 2021.

[9] Q. He, L. Kong, Y. Wang, S. Wang, T. A. Chan, and E. Holland, "Regularized quantile regression under heterogeneous sparsity with application to quantitative genetic traits," *Computational Statistics & Data Analysis*, vol. 95, pp. 222–239, Mar. 2016.

[10] Z. Y. Algamal, R. Alhamzawi, and H. T. M. Ali, "Gene selection for microarray gene expression classification using bayesian lasso quantile regression," *Computers in biology and medicine*, vol. 97, pp. 145–152, June 2018.

[11] Y. Wu and Y. Liu, "Variable selection in quantile regression," *Statistica Sinica*, vol. 19, no. 2, pp. 801–817, Apr. 2009.

[12] L. Xue, S. Ma, and H. Zou, "Positive-definite l1-penalized estimation of large covariance matrices," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1480–1491, Dec. 2012.

[13] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of statistics*, vol. 38, no. 2, pp. 894–942, Apr. 2010.

[14] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[15] R. Mirzaeifard, V. C. Gogineni, N. K. Venkategowda, and S. Werner, "Dynamic graph topology learning with non-convex penalties," in *30th European Signal Processing Conference*. IEEE, 2022, pp. 682–686.

[16] R. Mirzaeifard, N. K. Venkategowda, and S. Werner, "Robust phase retrieval with non-convex penalties," in *2022 56th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2022, pp. 1291–1295.

[17] A. Belloni and V. Chernozhukov, "l_1-penalized quantile regression in high-dimensional sparse models," *The Annals of Statistics*, vol. 39, no. 1, pp. 82–130, Feb. 2011.

[18] R. Koenker and P. Ng, "A frisch-newton algorithm for sparse quantile regression," *Acta Mathematicae Applicatae Sinica*, vol. 21, no. 2, pp. 225–236, May 2005.

[19] H. Zou and R. Li, "One-step sparse estimates in nonconcave penalized likelihood models," *Annals of Statistics*, vol. 36, no. 4, p. 1509, Aug 2008.

[20] B. Peng and L. Wang, "An iterative coordinate descent algorithm for high-dimensional nonconvex penalized quantile regression," *Journal of Computational and Graphical Statistics*, vol. 24, no. 3, pp. 676–694, July 2015.

[21] R. Mirzaeifard, N. K. Venkategowda, V. C. Gogineni, and S. Werner, "ADMM for sparse-penalized quantile regression with non-convex penalties," in *30th European Signal Processing Conference*. IEEE, 2022, pp. 2046–2050.

[22] S. Chen, A. Garcia, and S. Shahrampour, "On distributed nonconvex optimization: Projected subgradient method for weakly convex problems in networks," *IEEE Transactions on Automatic Control*, vol. 67, no. 2, pp. 662–675, Feb. 2021.

[23] D. Hajinezhad and M. Hong, "Perturbed proximal primal–dual algorithm for nonconvex nonsmooth optimization," *Mathematical Programming*, vol. 176, no. 1-2, pp. 207–245, Feb. 2019.

[24] V. Kungurtsev, "Distributed stochastic nonsmooth nonconvex optimization," *Operations Research Letters*, vol. 50, no. 6, pp. 627–631, Nov. 2022.

[25] R. Xin, U. A. Khan, and S. Kar, "An improved convergence analysis for decentralized online stochastic non-convex optimization," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1842–1858, Mar. 2021.

[26] B. Swenson, R. Murray, H. V. Poor, and S. Kar, "Distributed stochastic gradient descent: Nonconvexity, nonsmoothness, and convergence to local minima," *Journal of Machine Learning Research*, vol. 23, no. 328, pp. 1–62, Oct. 2022.

[27] R. Xin, S. Kar, and U. A. Khan, "Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 102–113, May 2020.

[28] H. Wang and C. Li, "Distributed quantile regression over sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 2, pp. 338–348, June 2017.

[29] R. Koenker and G. Bassett Jr, "Robust tests for heteroscedasticity based on regression quantiles," *Econometrica: Journal of the Econometric Society*, pp. 43–61, Jan. 1982.

[30] R. Varma, H. Lee, J. Kovačević, and Y. Chi, "Vector-valued graph trend filtering with non-convex penalties," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 48–62, Dec. 2019.

[31] Lin Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *42nd IEEE International Conference on Decision and Control*, vol. 5, 2003, pp. 4997–5002 Vol.5.