Learning Multiple Quantiles with Neural Networks

Sang Jun Moon[1], Jong-June Jeon[1], Jason Sang Hun Lee[2], Yongdai Kim[3]

November, 9, 2019

[1]Department of Statistics, University of Seoul
[2]Department of Physics, University of Seoul
[3]Department of Statistics, Seoul National University

Introduction

- The quantile regression model is widely used to learn the relationship between predictors and response variables.
- Estimating quantiles is considered as an alternative of learning distribution because a set of quantiles is an informative summarization of the distribution.
- Numerous models have been presented to capture the nonlinearity between quantiles and associated predictors.
- The non-linearity provides the flexibility of modeling for multiple quantiles, but a crossing problem exists in estimation.

Related works

- Cannon (2018) proposed the monotone composite quantile regression neural network (MCQRNN) in which non-crossing quantile estimates can be obtained.
- It employs a special structure of networks that satisfies the monotonicity across quantiles.
- However, we found that the MCQRNN is sensitive to the selection of the number of layers and overfitting problem frequently occur for complex patterns of conditional quantiles without weight decay.

Our contributions

- The purpose of this paper is to develop a neural network model and computation algorithm for non-crossing quantile regression based on another approach from Cannon's.
- The non-crossing multiple quantiles regression with neural networks adopts the concept of the non-crossing support vector regression with linear constraints.

Quantile regression model

- Let $Y \in \mathbb{R}$ and $\mathbf{X} \in \mathbb{R}^p$ be random vectors of response and predictor and $(y_i, \mathbf{x}_i),\ i = 1, \cdots, n$ be random samples from the distribution of $(Y, \mathbf{X})$
- Koenker(1994) proposed the M-estimation method for a linear model with $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)^\top \in \mathbb{R}^p$ that minimizes the empirical risk defined by

$$L_\tau(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i^\top \boldsymbol{\beta})$$

  where $\rho_\tau(u) = u(\tau - I(u < 0))$ which is called check function or tilted absolute value function.

- Let $0 < \tau_1 < \cdots < \tau_K < 1$ then the simultaneous estimation of multiple quantiles are estimated through minimizing

$$L(\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_K) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k}(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}_k)$$

where $\boldsymbol{\beta}_k = (\beta_{k1}, \cdots, \beta_{kp})^\top$ for $k = 1, \cdots, K$.

Non-crossing property

- By the definition of quantiles, conditional quantile functions should not be crossing on the support on $\mathbf{X}$.
- Thus, the non-crossing quantile regression model is estimated by M-estimation method, minimizing the quantile risk with constraints:

$$\begin{aligned} \min \quad & L(\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_K) \\ \text{subject to} \quad & \mathbf{x}^\top \boldsymbol{\beta}_1 \leq \cdots \leq \mathbf{x}^\top \boldsymbol{\beta}_K \ \text{ for all } \mathbf{x} \end{aligned}$$

Neural network approach

- We consider the τ_k-conditional quantile regression model as

$$f_{\tau_k}(\mathbf{x}) = \mathbf{z}(\mathbf{x}; \mathbf{\Theta})^\top \boldsymbol{\beta}_k$$

  for $k = 1, \cdots, K$ where $\mathbf{z}(\cdot; \mathbf{\Theta})$ is a feature map from $\mathbb{R}^p$ to $\mathbb{R}^q$ and $\mathbf{\Theta}$ is the parameter of the map, and $\boldsymbol{\beta}_k \in \mathbb{R}^q$.
- We always fix the first coordinate of the image $\mathbf{z}(\cdot; \mathbf{\Theta})$ as $z_1(\mathbf{x}; \mathbf{\Theta}) = 1$ such that β_{k1} is the intercept of the τ_k-conditional quantile regression model for each k.

- Then the M-estimation problem for non-crossing quantiles is defined by

$$\begin{aligned} \underset{\beta,\theta}{\operatorname{argmin}} \quad & L(\boldsymbol{\beta}, \boldsymbol{\Theta}) && (1) \\ \text{subject to} \quad & \mathbf{z}(\mathbf{x}; \boldsymbol{\Theta})^\top \boldsymbol{\beta}_k \le \mathbf{z}(\mathbf{x}; \boldsymbol{\Theta})^\top \boldsymbol{\beta}_{k+1} \\ & \text{for all } \mathbf{x},\ k = 1, \cdots, K-1, && (2) \end{aligned}$$

where $L(\boldsymbol{\beta}, \boldsymbol{\Theta}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k}(y_i - \mathbf{z}(\mathbf{x}_i; \boldsymbol{\Theta})^\top \boldsymbol{\beta}_k)$ and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \cdots, \boldsymbol{\beta}_K^\top)^\top \in \mathbb{R}^{qK}$.

- If $\sup_{\theta\in\Theta} \|\mathbf{z}(\mathbf{x};\boldsymbol{\Theta})\|_\infty \le 1$, the constraint (2) is represented by the polyhedron

$$\mathcal{C}_b = \{\boldsymbol{\beta} \in \mathbb{R}^{qK} : \mathbf{v}^\top \boldsymbol{\beta}_k \le \mathbf{v}^\top \boldsymbol{\beta}_{k+1} \ \forall \mathbf{v} \in \{0,1\}^q,\ k = 1, \cdots, K-1\}$$

- In addition, we can write the feasible region $\mathcal{C}_b$ as a simpler form by reparametrization of $\boldsymbol{\beta}$.
- Let $\boldsymbol{\delta}_1 = \boldsymbol{\beta}_1$, $\boldsymbol{\delta}_{k+1} = \boldsymbol{\beta}_{k+1} - \boldsymbol{\beta}_k$, $k = 1, \cdots, K-1$ and denote the jth element of $\boldsymbol{\delta}_k$ by δ_{kj}.
- Also, $\mathbf{v}^\top \boldsymbol{\beta}_k \le \mathbf{v}^\top \boldsymbol{\beta}_{k+1},\ \forall \mathbf{v} \in \{0,1\}^q$ if and only if

$$\delta_{k1} - \sum_{j=2}^{q} \max(0, -\delta_{kj}) \ge 0 \text{ for } k = 2, \cdots, K \tag{3}$$

(Bondell et al., 2010).

- The number of constraints is reduced to only $K - 1$.
- The constraints do not depend on the feature map or the parameter theta but only depend on the coefficients.
- I always be able to make the assumption holds for any x.
- In other words, it is easy to make z satisfy the assumption that the norm of z must be bounded.

- With the reparametrization of $\boldsymbol{\beta}$ to $\boldsymbol{\delta}$, the objective function (1) is also written as

$$L_r(\boldsymbol{\delta}, \boldsymbol{\Theta}) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n} \rho_{\tau_k} \left(y_i - \sum_{l=1}^{k} \mathbf{z}(\mathbf{x}_i, \boldsymbol{\Theta})^\top \boldsymbol{\delta}_l \right) \tag{4}$$

  in terms of $\boldsymbol{\delta} = (\boldsymbol{\delta}_1^\top, \cdots, \boldsymbol{\delta}_K^\top)^\top \in \mathbb{R}^{qK}$.

- Also, a feasible set of $\boldsymbol{\delta}$ is

$$\mathcal{C}_d = \left\{ \boldsymbol{\delta} \in \mathbb{R}^{qK} : \delta_{k1} - \sum_{j=2}^{q} \max(0, -\delta_{kj}) \geq 0, \text{ for } k = 2, \cdots, K \right\}. \tag{5}$$

Learning multiple quantiles

Interior-point method

- The interior-point method utilizes the barrier function in the original objective function.
- In the (5) the barrier function is given by

$$B(\boldsymbol{\delta}; M) = -\frac{1}{M}\sum_{k=2}^{K}\log(\delta_{k1} - \sum_{j=2}^{q}\max(0, -\delta_{kj}))$$

  where M is the tuning parameter.

- The following objective function is considered in the interior-point method:

$$L_B(\boldsymbol{\delta}, \boldsymbol{\Theta}; M) = L_r(\boldsymbol{\delta}, \boldsymbol{\Theta}) + B(\boldsymbol{\delta}; M) \tag{6}$$

- It is known that the duality gap of the optimal solution (6) is bounded above with $\frac{K-1}{M}$ (Boyd and Vandenberghe, 2004).
- However, the second approximation is needed in the algorithm which is almost intractable in our problem.
- As an alternative, the gradient descent algorithm can be considered as follows.

$$(\boldsymbol{\delta}^{(t+1)}, \boldsymbol{\Theta}^{(t+1)}) = (\boldsymbol{\delta}^{(t)}, \boldsymbol{\Theta}^{(t)}) - \eta_k \nabla L_B(\boldsymbol{\delta}^{(t)}, \boldsymbol{\Theta}^{(t)}; M_t)$$

- Because we can efficiently compute the gradient vector of $L_B(\boldsymbol{\delta}, \boldsymbol{\Theta}; M)$ by open-source software library such as Tensorflow and Pytorch, this method is computationally attractive.

- But, the updated solution $\delta^{(t+1)}$ should be contained in the domain of the barrier function to compute the next gradient direction, which requires delicate determination of the step size γ_t and parameters M_ts in the middle of each iteration.
- It is found that the updated solutions are frequently stuck on the boundary of the feasible set by a careless selection of γ_t and M_t, and these solutions show poor predictive performances.

Proposed computational algorithm

- We modified the original barrier function by introducing auxiliary variables. Let $\tilde{\delta}_{k1} = \max\left(\delta_{k1}, \sum_{j=2}^{q} \max(0, -\delta_{kj}) + \epsilon_\delta\right)$ with a constant $\epsilon_\delta \in (0, 1)$ and $\tilde{\delta}_{kj} = \delta_{kj}$ for $k = 2, \cdots, K$ and $j = 2, \cdots, q$ and define a new barrier function depending on auxiliary variables as

$$B(\tilde{\boldsymbol{\delta}}; M) = -\frac{1}{M} \sum_{k=2}^{K} \log\left(\tilde{\delta}_{k1} - \sum_{j=2}^{q} \max(0, -\tilde{\delta}_{kj})\right).$$

- The auxiliary variable $\tilde{\boldsymbol{\delta}}$ lies on the feasible set, and thus $B(\tilde{\boldsymbol{\delta}}; M)$ is always defined.
- In addition, we apply the l_1 penalty function to $(\boldsymbol{\delta}_k - \tilde{\boldsymbol{\delta}}_k)$ for $k = 2, \cdots, K$ to shrink the auxiliary variable $\tilde{\boldsymbol{\delta}}_k$ towards the original variable $\boldsymbol{\delta}_k$.
- The proposed objective function is written as follows:

$$
\begin{aligned}
\min_{\delta,\theta} \quad & L_r(\boldsymbol{\delta}, \boldsymbol{\Theta}) + B(\tilde{\boldsymbol{\delta}}; M) + \lambda \sum_{k=2}^{K} \|\boldsymbol{\delta}_k - \tilde{\boldsymbol{\delta}}_k\|_1 \qquad (7) \\
\text{where} \quad & \tilde{\delta}_{k1} = \max\left(\delta_{k1}, \sum_{j=2}^{q} \max(0, -\delta_{kj}) + \epsilon_\delta\right), \\
& \tilde{\delta}_{kj} = \delta_{kj} \text{ for } k = 2, \cdots, K,\ j = 2, \cdots q,
\end{aligned}
$$

where $\lambda \geq 0$ is the tuning parameter.

- Let the optimal solution of (7) be $(\boldsymbol{\delta}^*(\lambda), \tilde{\boldsymbol{\delta}}^*(\lambda), \boldsymbol{\Theta}^*(\lambda))$ and the proposed τ_k-quantile function is given by

$$f^*_{\tau_k}(\mathbf{x}) = \mathbf{z}(\mathbf{x}; \boldsymbol{\Theta}^*(\lambda))^\top \boldsymbol{\beta}^*_k(\lambda),$$

where $\boldsymbol{\beta}^*(\lambda) = (\boldsymbol{\beta}^*_1(\lambda)^\top, \cdots, \boldsymbol{\beta}^*_K(\lambda)^\top)^\top$ and $\boldsymbol{\beta}^*_k(\lambda) = \sum_{l=1}^k \tilde{\boldsymbol{\delta}}^*_l(\lambda)$.

- By definition of $\tilde{\boldsymbol{\delta}}$, the non-crossing condition is always satisfied regardless of λ, that is

$$f^*_{\tau_k}(\mathbf{x}) \le f^*_{\tau_{k+1}}(\mathbf{x}) \text{ for } k = 1, \cdots, K-1.$$

- Our proposed algorithm, Adaptive interior-point (AIP), is as follows.

AIP algorithm

1. Let $t = 0$ and set an initial $\boldsymbol{\delta}^{(t)}$ and $\boldsymbol{\Theta}^{(t)}$.
2. Repeat:
   - Update $\tilde{\boldsymbol{\delta}}^{(t)}$ with $\boldsymbol{\delta}^{(t)}$ and set $\mathcal{A}^{(t)} = \{k : \delta_{k1}^{(t)} \neq \tilde{\delta}_{k1}^{(t)}\}$
   - Update $\boldsymbol{\delta}^{(t+1)}$:
     - $\boldsymbol{\delta}_k^{(t+1)} = \boldsymbol{\delta}_k^{(t)} - \gamma_t \left(\nabla_{\delta_k} L(\boldsymbol{\delta}^{(t)}, \boldsymbol{\Theta}^{(t)}) + \nabla_{\delta_k} \phi(\boldsymbol{\delta}_k^{(t)}; M)\right)$ for $k \in \mathcal{A}^{(t)}$
     - $\boldsymbol{\delta}_k^{(t+1)} = \boldsymbol{\delta}_k^{(t)} - \gamma_t \left(\nabla_{\delta_k} L(\boldsymbol{\delta}^{(t)}, \boldsymbol{\Theta}^{(t)}) + \lambda \mathrm{sign}(\boldsymbol{\delta}_k^{(t)} - \tilde{\boldsymbol{\delta}}_k^{(t)})\right)$ for $k \notin \mathcal{A}^{(t)}$
   - Update $\boldsymbol{\Theta}^{(t+1)}$:
     - $\boldsymbol{\Theta}^{(t+1)} = \boldsymbol{\Theta}^{(t)} - \gamma_t \nabla_\theta L(\boldsymbol{\delta}^{(t)}, \boldsymbol{\Theta}^{(t)})$
     - $t \leftarrow t + 1$

Example

- To illustrate the trajectory of the updated solutions provides by the proposed algorithm we consider the following problem as a toy example:

$$\begin{aligned} \min_{\delta} \quad & (\delta_1 + 1)^2 + \frac{1}{4}(\delta_2 + 5)^2 \\ \text{subject to} \quad & \delta_1 - \max(0, -\delta_2) \geq 0, \end{aligned}$$

- Let $\tilde{\delta_1} = \max(\delta_1, \max(0, -\delta_2) + \epsilon_\delta)$ and $\tilde{\delta}_2 = \delta_2$, and $B(\tilde{\boldsymbol{\delta}}; M) = -\log(\tilde{\delta}_1 - \max(0, -\delta_2))/M$ with $\tilde{\boldsymbol{\delta}} = (\tilde{\delta}_1, \tilde{\delta}_2)^\top$.

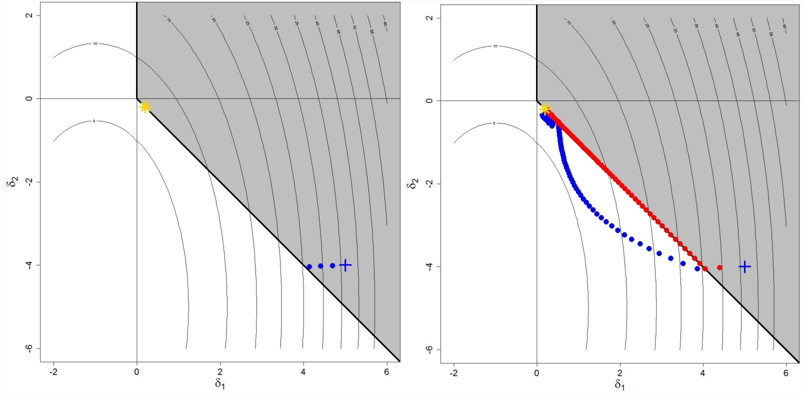

Figure 1: Yellow star denotes the optimal solution; Blue cross point denotes the initial point; Blue points denote $\boldsymbol{\delta}$ and red points denote $\tilde{\boldsymbol{\delta}}$; Left panel shows the updated solutions of interior point method; Right panel shows those of AIP algorithm

- We show that the optimal solution to the above problem for the large enough λ is same as the optimal solution from the interior-point method problem (6).

Theorem

Consider a fixed $M > 0$ and $\epsilon_\delta \in (0, 1)$. Let $(\boldsymbol{\delta}^*(\lambda), \tilde{\boldsymbol{\delta}}^*(\lambda), \boldsymbol{\Theta}^*(\lambda))$ be the optimal solution of (7) for given $\lambda > 0$. Let $\lambda_{\max} = |\max_{k=2,\cdots,K} \nabla_{\delta_{k1}} L_r(\boldsymbol{\delta}, \boldsymbol{\Theta})|$. Then $\boldsymbol{\delta}^*(\lambda) = \tilde{\boldsymbol{\delta}}^*(\lambda)$ for $\lambda > \lambda_{\max}$.

Simulation and real data analysis

- Simulation 1:
  $y = \sin(\pi \mathbf{x})/(\pi \mathbf{x}) + \epsilon,\ \mathbf{x} \sim U(-1, 1),\ \epsilon \sim N(0, \exp(1 - \mathbf{x})/10)$
- Simulation 2: $y = (-1, -2, \cdots, -p)^\top \mathbf{x} + \epsilon,\ \mathbf{x} \sim U(-2, 2)^p$,
  $\epsilon_i \sim N(0, \exp(1 + \min(\|\mathbf{x}\|_2^2 I(\|\mathbf{x}\|_2^2 \geq 1), 4)/2)$
- Simulation 3: $y = -3\mathbf{x} + \epsilon$, $\mathbf{x} \sim U(0, 4)$, $\epsilon \sim N(0, \exp(1 - \mathbf{x}))$
- Note that in the setting of the simulation 1, the true conditional quantile function is given by

$$f_\tau^*(\mathbf{x}) = \sin(\pi \mathbf{x})/(\pi \mathbf{x}) + \exp(\sin(2\pi \mathbf{x}))\Phi^{-1}(\tau)$$

  for $-1 \leq x \leq 1$, where $\Phi^{-1}(\cdot)$ is the quantile function of standard normal distribution.

- We fix the learning rates by 0.005 in the methods except the interior point method.
- The learning rate of the interior point method is carefully corrected to update the solution in the feasible set.
- The predictive performance of the trained model is evaluated by the quantile risk on the test set,

$$L(f^*) = \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m}\rho_{\tau_k}(\tilde{y}_i - f^*_{\tau_k}(\tilde{\mathbf{x}}_i))$$

  where $(\tilde{y}_i, \tilde{\mathbf{x}}_i)$ for $i = 1, \cdots, m$ are the samples of the test data set.

- Throughout all simulations we let $(\tau_1, \cdots, \tau_K) = (0.1, 0.25, 0.5, 0.75, 0.9)$ and fit the models with 200 training samples and compared the performances with 1000 test samples by 100 times repeated numerical simulations.
- In simulation 1 and 2, all models have 2 hidden layers with 4 hidden nodes per hidden layer.
- We set the number of maximum iterations to 2000 epochs, and the tuning parameter λ as 5.
- Note that the number of trials to correct the learning rate is not considered as a single epoch when the interior-point method is applied.

Models		QRNN	MCQRNN	Projection
Performance	mean	0.582	0.506	0.460
	median	0.535	0.477	0.460
	sd	0.146	0.134	0.016
Models		**Interior-point**	**AIP**	
Performance	mean	0.608	**0.457**	
	median	0.563	**0.456**	
	sd	0.124	**0.015**	
Models		**QRNN**	**MCQRNN**	**Projection**
Time	mean	1.18	3.14×10	1.34×10^4
	median	1.18	3.03×10	1.24×10^4
	sd	7.20×10^{-2}	3.62	9.04×10^3
Models		**Interior-point**	**AIP**	
Time	mean	5.61×10^3	**8.30**	
	median	4.32×10^3	**8.03**	
	sd	3.94×10^3	**1.97**	

Table 1: Results in simulation 1

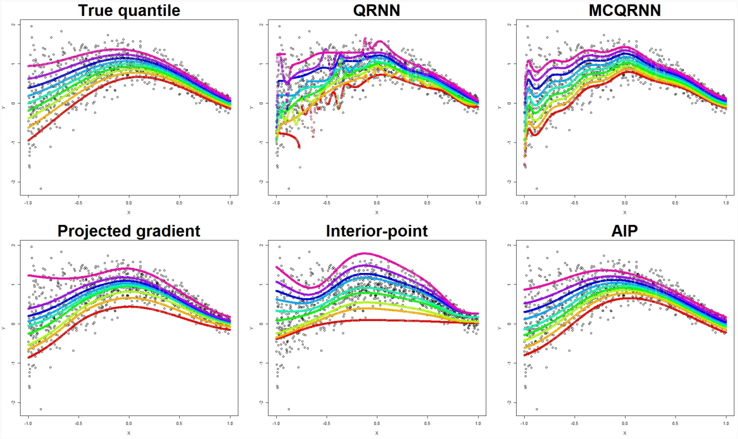

Figure 2: Illustration of simulation 1 with 1000 training samples and $\tau = (0.1, \cdots, 0.9)$

model		dimension				
		1	2	3	4	5
MCQRNN	mean	302.2	4426.3	12961.4	538.7	4617.8
	median	9.3	18.2	27.3	31.7	34.8
	sd	2774.9	43242.7	62949.7	1866.3	36942.3
AIP	mean	9.0	16.8	23.4	26.7	29.2
	median	8.9	16.7	23.4	26.4	29.1
	sd	0.8	1.2	1.6	1.8	1.8

Table 2: Mean, median and standard deviation (sd) of test losses in simulation 2

model		layer				
		1	2	3	4	5
MCQRNN	mean	1.667	1.643	3.371	4.597	6.095
	median	1.534	1.495	1.964	5.185	7.430
	sd	0.747	0.752	2.620	2.974	2.552
AIP	mean	1.013	1.000	1.012	1.041	1.146
	median	1.009	0.995	1.008	1.034	1.063
	sd	0.056	0.050	0.051	0.061	0.550

Table 3: Mean, median and standard deviation (sd) of test losses in simulation 3

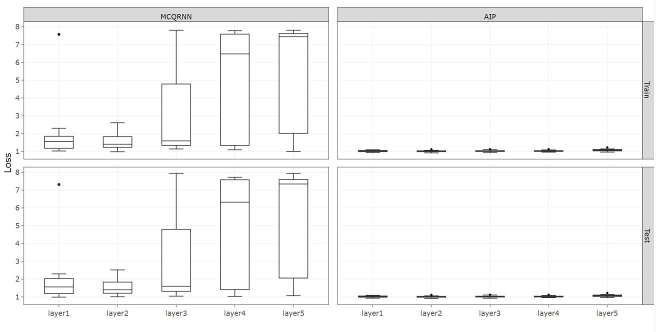

Figure 3: The boxplot of test losses in the third example

Real data analysis

- We use scaled YVRprecip dataset embedded in the qrnn package of the R program.
- The dataset is about daily precipitation totals (mm) at Vancouver int'l Airport for the period 1971-2000.
- Seasonal cycle, daily sea-level pressures (Pa), 700-hPa specific humidities (kg/kg), and 500-hPa geopotential heights (m) are included.
- We fit the all algorithms for the conditional $0.8, 0.85, 0.9$-quantile 100 times.
- The data before 1975 is used for training and the remaining years is used for testing.

Models	QRNN	MCQRNN	Projection	Interior-point	AIP
mean	4.136	4.016	**3.845**	4.163	**3.853**
median	4.112	4.026	**3.845**	4.163	**3.852**
sd	0.159	0.058	**0.037**	0.084	**0.041**

Table 4: Mean, median and standard deviation (sd) of test losses in the Real data analysis

Conclusion

- We suggest the non-crossing non-linear quantile regression using modified neural network model.
- Simulation and real data analysis show that the performance of the proposed methods is better or competitive with existing methods.
- AIP algorithm suggests a method dealing with the efficient first-order method for the optimization on a feasible set even though the feasible set consists of simple linear constraints.

Reference

- Bondell, Howard D., Brian J. Reich and Huixia Wang. "Noncrossing quantile regression curve estimation." Biometrika 97.4 (2010): 825-838.
- Cannon, Alex J. "Quantile regression neural networks: Implementation in R and application to precipitation downscaling." Computers & geosciences 37.9 (2011): 1277-1284.
- Cannon, Alex J. "Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes." Stochastic environmental research and risk assessment 32.11 (2018): 3207-3225.
- Koenker, Roger and Kevin F. Hallock. "Quantile regression." Journal of economic perspectives 15.4 (2001): 143-156.