Computation in Quantile and Composite Quantile Regression Models with or without Regularization

by

Jueyu Gao

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

STATISTICAL MACHINE LEARNING

Department of Mathematical and Statistical Sciences

University of Alberta

Abstract

Quantile, composite quantile regression with or without regularization have been widely studied and applied in the high-dimensional model estimation and variable selections. Although the theoretical aspect has been well established, the lack of efficient computation methods and publicly available programs or packages hinder the research in this area. Koenker [11] has established and implemented the interior point(IP) method in **quantreg** [12] for quantile regression with or without regularization. However, it still lacks the ability to handle the composite quantile regression with or without regularization. The same incapability also existed in Coordinate Descent (CD) algorithm that has been implemented in **CDLasso** [8]. The lack of handful programs for composite quantile regression with or without regularization motivates our research here. In this work, we implement three different algorithms including Majorize and Minimize(MM), Coordinate Descent(CD) and Alternation Direction Method of Multiplier(ADMM) for quantile and composite quantile regression with or without regularization. We conduct the simulation that compares the performance of four algorithms in time efficiency and estimation accuracy. The simulation study shows our program is time efficient when dealing with high dimensional problems. Based on the good performance of our program, we

publish the R package **cqrReg** [7], which give the user more flexibility and capability when directing various data analyses. In order to optimize the time efficiency, the package **cqrReg** [7] is coded in **C++** and linked back to R by an user-friendly interface.

Acknowledgements

First of all, I would like to express my deep gratitude to Professor Kong and Professor Gombay, my research supervisors, for their patient guidance, enthusiastic encouragement and valuable critiques of this research work.

It has been a truly wonderful experience to work with the staff in the Department of Mathematical and Statistical Sciences, whose professional services are gratefully acknowledged.

I would also like to extend my thanks to all my friends in Edmonton. They have encouraged me and helped me a lot during these days. Without the experience they shared with me, I might lost my way. In particular, I would like to thank to Miss Su, who helped me to get out of the dark days and brought the sunshine to my life.

Finally, thanks to my parents, for their unconditional encouragement, prudent advice and long-lasting support.

Contents

Abstract **ii**

Acknowledgements **iv**

List of Tables **viii**

List of Figures **ix**

1 Introduction **1**

1.1 Quantile and Composite Quantile Regression 1

1.1.1 Quantile Regression 1

1.1.2 Composite Quantile Regression 4

1.2 Four Algorithms for Convex Optimization 6

1.2.1 Interior Point (IP) algorithm 7

1.2.2 Majorize and Minimize (MM) algorithm 8

1.2.3 Coordinate Descent(CD) Algorithm 10

1.2.4 Alternating Direction Method of Multipliers (ADMM) 11

1.3 My Contribution 14

2 Algorithms **17**

2.1 Quantile Regression 17

2.1.1 IP Algorithm 17

2.1.2 MM Algorithm 19

2.1.3 CD Algorithm 21

2.1.4 ADMM Algorithm 22

2.2 Quantile Regression with Regularization 24

2.2.1 IP Algorithm 24

2.2.2 MM Algorithm 25

2.2.3 CD Algorithm 27

2.2.4 ADMM Algorithm 28

2.3 Composite Quantile Regression 31

2.3.1 IP Algorithm 31

2.3.2 MM Algorithm 32

2.3.3 CD Algorithm 32

2.3.4 ADMM Algorithm 33

2.4 Composite Quantile Regression with Regularization 35

2.4.1 IP Algorithm 35

2.4.2 MM Algorithm 36

2.4.3 CD Algorithm 37

2.4.4 ADMM Algorithm 37

3 Simulation **40**

3.1 Quantile Regression and Composite Quantile Regression . . . 41

3.2 Quantile Regression and Composite Quantile Regression with Regularization 44

4 Conclusion **49**

Bibliography **52**

List of Tables

3.1 Quantile regression with normal distributed error 41

3.2 Quantile regression with t distributed distributed error 41

3.3 Composite quantile regression with normal distributed error . 42

3.4 Composite quantile regression with t distributed error 43

3.5 Quantile regression with regularization and normal distributed error 45

3.6 Quantile regression with regularization with t distributed error. 46

3.7 Composite quantile regression with regularization and normal distributed error 47

3.8 Composite quantile regression with regularization and t distributed error. 47

List of Figures

3.1 Compare the running time of four algorithm for quantile and composite quantile regression 43

3.2 Compare the running time of four algorithm for quantile and composite quantile regression with regularization 46

Chapter 1

Introduction

1.1 Quantile and Composite Quantile Regression

1.1.1 Quantile Regression

Weisberg [22] defines a regression model as the study of dependence. For the most of statisticians, the goal of regression model is to summarize the embedded patterns that hide in the observed data. In other words, how the response variable varies as the value of the covariates changes. Denote the response variable as $\mathbf{Y} = [y_1, y_2, ..., y_n]^T$ and the covariates matrix as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$, where n is the sample size. The regression model can be denoted as $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}$, where $\boldsymbol{\beta}$ is the coefficient and $\boldsymbol{\epsilon}$ is the error term. There are two major assumptions concerning the error. First, there is no common patterns between the corresponding error and covariates, denoted as $E(\epsilon_i|X = x_i) = 0$. The second assumption is that the error are independent, meaning that the value of the error for one case gives no information about the value of the error for another cases [22]. There are many well known methods that have been implemented for regression model estimation. The most common way is to obtain the mean response variable by the ordinary least square (OLS) estimation, which minimizes the sum of squared residuals.

$$\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i, \boldsymbol{\beta}))^2. \tag{1.1}$$

Under the normality assumption of the error distribution, it is the maximum likelihood estimator. However, if the error follows heavy tailed distribution or the data itself has outliers, the estimator will be inefficient. In that case, Rud̈er Josip Bosković, a Jesuit Catholic priest from Dubrovnik [19] introduces the least absolute deviation (LAD) regression, which estimates the conditional median function by minimizing the sum of absolute value of residuals.

$$\sum_{i=1}^{n}|(y_i - f(\mathbf{x}_i, \boldsymbol{\beta}))|. \tag{1.2}$$

The least absolute deviation regression works well even when the error has a distribution without finite variance. In addition, it is the maximum likelihood estimation (MLE) if the error follows the Laplace distribution.

Generally, the least absolute deviation regression belongs to the family of quantile regression. In other words, the least absolute deviation regression is the 0.5 quantile level quantile regression. Denote the quantile level $\tau \in (0, 1)$, quantile function obtains the value from the data that there are τ data below it and $1 - \tau$ data above it. The empirical version of the principals says, a sample τ quantile u_τ of n numbers $y_1, ..., y_n$ is a minimizer of the function

$L(u) = \sum_{i=1}^{n} \rho_\tau(y_i - u)$ [13], where

$$\rho_\tau(t) = t(\tau - I(t < 0)) \tag{1.3}$$

Koenker and Basett [13] define the τth quantile regression as any vector $\boldsymbol{\beta}$ minimizing the following

$$\sum_{i=1}^{n} \rho_\tau(y_i - f(\mathbf{x}_i, \boldsymbol{\beta})). \tag{1.4}$$

Quantile regression shares the nice property of least absolute deviation regression when countering outliers in the response measurements. Moreover, the advantage of quantile regression goes beyond that. Quantile regression can provide a big scope of the relationship between response variables and covariates. Compare with the least absolute deviation regression that mainly focus on the median of residuals, the quantile regression could provide more information about how the response variable is influenced by covariates in the tail parts.

In the era of "big data", the high dimensional problem makes many classical regression method less powerful. In that case, Tibshirani [20] introduces well known technique, called the lasso, for "least absolute shrinkage and selection operator". It minimizes the residuals sum of square subject to the sum of the absolute value of the coefficients being less than a constant. The extended version is the ordinary least square estimation with absolute value penalty [20] .

$$\sum_{i=1}^{n} (y_i - f(\mathbf{x}_i, \boldsymbol{\beta}))^2 + \lambda |\boldsymbol{\beta}| \tag{1.5}$$

where $\lambda \geq 0$ is a tuning parameter that controls the level of shrinkage in estimation process. The value of λ is estimated by cross validation. The Lasso has two main advantages than ordinary least square estimation, in terms of prediction accuracy and interpretation. For predicting accuracy, the lasso shrinks some coefficients into 0 that sacrifices the bias in order to reduces the variance of the predicted model that leads the prediction more accurate. For interpretation, the less number of the predictors that have the strong effects make the interpretation more clearly [20].

1.1.2 Composite Quantile Regression

Fan and Li [5] introduce the theory of oracle model selection to lead the construction of optimal model selection procedures. Considering the following linear model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1.6}$$

The goal of variable selection is to identify the unknown set of coefficients $\beta^*_{\mathcal{A}} : \beta_j \neq 0$, in other words the significant components of β . Fan and Li suggest the situation that there is the oracle who already know the true set of $\mathcal{A}$. Denote the $LS - oracle$ as the oracle estimator derive from least square estimation. Fan and Li prove that

$$\sqrt{n}(\beta^{LS-oracle}_{\mathcal{A}} - \beta^*_{\mathcal{A}}) \xrightarrow{d} N(0, \sigma^2 \boldsymbol{C}^{-1}_{\mathcal{A}\mathcal{A}}) \tag{1.7}$$

where X is the design matrix and $\underset{n\to\infty}{lim} \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{C}$, and $\boldsymbol{C}$ is $p \times p$ positive definite matrix. Defined $\boldsymbol{C}_{\mathcal{A}\mathcal{A}}$ the sub-matrix of $\boldsymbol{C}$ with both column and row index in $\mathcal{A}$ [26]. However, the oracle "estimator" is not a real estimator due to the known information about the true parameter. Fan and Li introduce the variable selection and coefficient estimation procedure η is a $LS - oracular$ estimator, if $\hat{\beta}(\eta)$ has the following properties [26]:

- Consistent selection: $Pr(\{j : \hat{\beta}(\eta)_j \neq 0\} = \mathcal{A}) \rightarrow 1$
- Efficient estimation: $\sqrt{n}(\hat{\beta}(\eta)_{\mathcal{A}} - \beta^*_{\mathcal{A}}) \xrightarrow{d} N(0, \sigma^2 \boldsymbol{C}^{-1}_{\mathcal{A}\mathcal{A}})$

Fan and Li show that the SCAD well satisfies the oracle properties. Zou [26] elaborates the adaptive lasso also attains the oracle properties. However, the $\hat{\beta}^{LS}$ has the limitation of root-n consistency when facing the infinite error variance. Based on Fan and Li, Zou and Yuan [26] introduced a new regression method called composite quantile regression (CQR) as following:

$$(\hat{b_1}, ...\hat{b_k}, \beta^{\hat{CQR}}) = \arg \min_{b_1,...b_k,\beta} \sum_{i=1}^{k} \rho_{\tau_i}(Y - b_i - X\beta) \tag{1.8}$$

where $0 < \tau_1 < \tau_2 < \tau_3... < \tau_k < 1$, denote b_k the kth quantile intercept coefficient. The composite quantile regression simultaneously consider multiple quantile levels that combines the advantage of the multiple quantile regression model. The composite quantile regression model has the same coefficients across various quantile levels [26]. In addition, Zou and Yuan [26] present the composite quantile regression with adaptive lasso penalty, which could also obtains a $CQR - oracular$ estimator that satisfies the oracle property and durable in the infinite error variance.

$$(\hat{b_1}, ...\hat{b_k}, \hat{\beta}^{CQRL}) = \arg \min_{b_1,...b_k,\beta} \sum_{i=1}^{k} \rho_{\tau_i}(Y - b_i - X\beta) + \lambda \frac{|\beta|}{|\beta^{CQR}|^2} \tag{1.9}$$

Besides the same setting for composite quantile regression, the β^{CQR} is derived from the composite quantile regression and λ is the tuning parameter.

Moreover, Zou and Yuan [26] show the $CQR - oracle$ has two noteworthy properties:

- It has 70% more relative efficiency than $LS - oracle$.
- Even if the error does not follow Gaussian distribution, its relative efficiency could be arbitrarily large.

Inspired by Zou and Yuan, Wu and Liu [25] prove the oracle property of quantile regression with adaptive lasso penalty. The oracle theorem for quantile regression with adaptive lasso penalty depends on the following conditions:

- The regression error is independent and identically distributed , with τth quantile zero and a continuous, positive density in a neighborhood of zero.
- The design matrix X is a deterministic sequence for which there is a positive definite matrix $\mathbf{C}$ where $\underset{n\to\infty}{lim}\frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{C}$

Based on the superior properties illustrate above, we conduct the quantile regression with regularization in the following form:

$$\hat{\beta}^{QRL} = \arg\min_{\beta}\ \rho_\tau(Y - X\beta) + \lambda\frac{|\beta|}{|\beta^{QR}|^2} \tag{1.10}$$

and composite quantile regression with regularization in the form of (1.9).

1.2 Four Algorithms for Convex Optimization

In this section, we briefly review the four popular algorithms for the convex optimization problem that our research problems belong to.

1.2.1 Interior Point (IP) algorithm

Linear programing has been well known in the history of optimization. Its formulation could be traced back to the 1930s and 1940s. Since the development of the Simplex algorithm by Dantzig in the mid 1940s, many researchers implement it in the economic, finance and engineering area [23]. Due to non-polynomial running time of Simplex algorithm, Karmarkar proposes the polynomial time algorithm called primal dual interior point method that follows the early work from Fiacco and McCornick on barrier methods for constrained optimization [11]. In this section, we briefly review the concept of linear programing and Interior Point (IP) method from Wright [23] and Koenker [11].

There are three fundamental properties of a linear programing problem [23].

- A vector of real variable that the optimal value solves the problem
- A linear objective function
- The inequality or equality linear constrains

The standard form of the linear programing problem defines as following:

$$\min \ c^T x \text{ subject to } Ax = y, \ \ x \geq 0 \tag{1.11}$$

where c and x are vectors in $\mathbb{R}^n$, y is the a vector in $\mathbb{R}^m$, and A is an $m \times n$ matrix. The standard form of linear programing is also called the primal problem that distinguishes from the dual problem, which is defined as following:

$$\max \ y^T d \text{ subject to } A^T d \leq c \tag{1.12}$$

where d is a vector in $\mathbb{R}^m$. The components of d are called the dual variable. The dual problem consists of the same objects of primal problem that are arranged in a different way. The duality theory explains the relationship between the primal and dual problem. For instance, given the vector x satisfies the constrains $Ax = y, \;\; x \geq 0$ and d for (1.12), we have

$$y^T d \leq c^T x \tag{1.13}$$

In other words, the primal solution gives an upper bound on the dual, and the dual objective gives a lower bound on the primal problem. The basic idea of interior point method is using variants of Newton's method to derive the optimal solution that satisfies the Karush-Kuhn-Tucker(KKT) condition of the primal-dual problem. The details of duality theory and KKT condition could be found in the reference book that is written by Wright [23]. Since our research problem could be formulated as the linear programming problem, we implement the interior point for the quantile regression and composite quantile regression with or without regularization. We give details about how we translate the our research problem into linear programing problem in Chapter 2. It is worth noticing that we use the interior point method that is implemented in the **Rmosek** [6].

1.2.2 Majorize and Minimize (MM) algorithm

Ortega and Rheiboldt [16] introduce the technique of implementing the majorize function of the original function to estimate the minimization point of the original problem. The most well known example of MM algorithm is the EM algorithm. It is widely used in estimating the maximal likelihood estimator. Although the basic idea is the same, EM stands for minorized-maximized instead of majorize-minimize. Many application of MM algorithm could be found, for example, Heiser[9] and Lange [15]. The MM algorithm breaks down the relatively hard part of optimization problem into easier sub problems. In addition, the solution of substitute problems always converges to a solution of the original problems. In that case, MM algorithm is an ideal solver for non-differentiable regression model like quantile and composite quantile regression with or without regularization. Concerning the optimization problem is to minimize the objective function $L(\theta) : \mathbb{R}^p \to \mathbb{R}$. Majorize and Minimize algorithm proceed the estimation in two step. First, we establish a surrogate function $Q(\theta|\theta^k) : \mathbb{R}^p \to \mathbb{R}$ satisfying

$$Q(\theta^k|\theta^k) = L(\theta^k) \tag{1.14}$$

$$Q(\theta|\theta^k) \geq L(\theta) \quad \text{for all} \quad \theta \tag{1.15}$$

The surrogate function $Q(\theta|\theta^k)$ is called the majorize function of $L(\theta)$ at θ^k. The second step of the MM algorithm is to derive the estimator θ^{k+1} that minimize the majorize function $Q(\theta|\theta^k)$. The estimation process repeats the above two steps until the estimator are converged. In other words, for each iteration the algorithm creates the new surrogate function $Q(\theta|\theta^{k+1})$ and looks for the local minimum. The challenge of MM algorithm is to derive the suitable majorize function that could be easily minimized [10]. In this thesis, we illustrate the procedure of implementing the MM algorithm for quantile, composite quantile regression with or without regularization in Chapter 2.

1.2.3 Coordinate Descent(CD) Algorithm

Wu and Lange [24] introduce the Coordinate Descent(CD) algorithm for LAD that is based on greedy coordinate descent and Edgeworth's [4] algorithm for ordinary LAD regression. This section briefly introduce the basic idea of greedy coordinate descent algorithm and how Wu and Lange [24] generate the new version LAD coordinate descent algorithm.

In the past ten years, due to the explosion of the information, many classical regression methods are losing their strength. The inner reason is the standard regression method always involves matrix inversion, matrix diagonalization that need enormous numbers of arithmetic operations as the cube of the number of predictors. The challenge motivates many researchers like Park and Hastie [17], Wang [21]. In that circumstance, Wu and Lange introduce the coordinate descent algorithm, that has nice properties such as simplicity, time efficiency and stability. However, the concept of coordinate descent is full of history. In 1887, Edgeworth introduced the algorithm in L_1 regression that become the main competitor of least square regression. In 1997, Portnoy and Koenker [18] followed the algorithm from Boscovich to Laplace to Edgeworth. Recently, Wu and Lange propose the greedy coordinate descent based on the Edgeworth's algorithm from Claerbout and Muir [3]. To expose the nature of coordinate descent algorithm, consider minimizing the single variable L_1 regression model $L(\boldsymbol{\beta}) = |\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - u|$, where u is the intercept. The first step is to update u by the sample median of $z = |\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}|$ that will drive the $L(\boldsymbol{\beta})$ downhill. We could rewrite the loss function.

$$L(\beta) = \sum_{i=1}^{n} |x_{ik}||\frac{y_i - u - \sum_{j \neq k} x_i \beta}{x_i} - \beta| \tag{1.16}$$

Before the algorithm goes to second step, Edgeworth [4] defines the weighted median of series of numbers $(x_1, \ldots, x_n)$ as $\{x_k | S_{k-1} < S_n/2 \leq S_k, k \in n\}$, where the cumulative sum of numbers $S_n = \sum\limits_{1}^{n} x_i$. In the second step, the CD algorithm sorts the numbers $z_i = \frac{y_i - u}{x_i}$, and then updates parameter β by the order statistic $z_{[i]}$ whose index i satisfies following condition (in other words weighted median)

$$\sum_{j=1}^{i-1} w_{[j]} < \frac{1}{2}\sum_{j=1}^{n} w_{[j]} \qquad \sum_{j=1}^{i} w_{[j]} \geq \frac{1}{2}\sum_{j=1}^{n} w_{[j]} \tag{1.17}$$

where $w_i = |x_i|$ is corresponding to the z_i. Moreover, the CD algorithm also works for the lasso penalty. The lasso penalty function $\lambda|\beta|$ could be regarded as the term of absolute value pseudo-residuals.

$$\lambda|\beta| = |y_{n+1} - x_{n+1}\beta| \tag{1.18}$$

where $y_{n+1} = 0$, $x_{n+1} = \lambda$. In other words, we add one more row in the design matrix equals the value of λ, at the same time add one more response variable $y_{n+1} = 0$. Wu and Lange [24] demonstrate the weak consistency of penalized L_1 regression. In addition, the simulation study they have done proves the CD algorithm is time efficient and stable. In Chapter 2, the thesis comprehensibly explains how coordinate descent(CD) algorithm could be implemented in quantile, composite quantile regression with or without regularization.

1.2.4 Alternating Direction Method of Multipliers (ADMM)

This section briefly reviews the powerful convex optimization algorithm that is well suited to the research problem of the thesis. It is called Alternating Direction Method of Multiplier(ADMM), and introduced by Boyd [1]. However, the algorithm itself is not invented by Boyd, it was first introduced in 1970s, and well established in 1990s. It follows the decomposition-coordination procedure, in which the global optimization problem is solved by coordinating the solution of local sub problems. ADMM could be regarded as the mixture of the dual decomposition and augmented Lagrangian methods for constrained optimization [1]. It is implemented in many fields, such as iterative algorithms for L_1 problems in signal process by Bregman [2]. We implement the ADMM in this work, since quantile, composite quantile regression with or without regularization are typically convex optimization problems. Firstly, we review the basic idea about ADMM in the following part of this section. In Chapter 2, the thesis illustrates the procedure about how we implement the ADMM for our problem. First, the general form of ADMM problem is:

$$\begin{aligned} \min \quad & f(x) + g(z) \\ \text{subject to} \quad & Ax + Bz = c \end{aligned} \tag{1.19}$$

where f and g are convex. The Lagrangian function can be wrote as:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)||Ax + Bz - c||_2^2 \tag{1.20}$$

where ρ is the constant number called penalty parameter. The ADMM dual variables for each iteration k are:

$$\begin{aligned}
x^{k+1} &= \arg\min_x L_\rho(x, z^k, y^k) \\
z^{k+1} &= \arg\min_z L_\rho(x^{k+1}, z, y^k) \\
y^{k+1} &= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \qquad (1.21)
\end{aligned}$$

Combined the linear and the quadratic term in augmented Lagrangian, equation (1.20) can write as

$$\begin{aligned}
L_\rho(x, z, y) &= f(x) + g(z) + y^T(Ax + Bz - c) + (\rho/2)||Ax + Bz - c||_2^2 \\
&= f(x) + g(z) + (\rho/2)||Ax + Bz - c + u||_2^2 - (\rho/2)u^2, \qquad (1.22)
\end{aligned}$$

with $u = (1/\rho)y$ and $u^k = (1/\rho)y^k$. Then the ADMM dual variables (scaled dual form) are:

$$\begin{aligned}
x^{k+1} &= \arg\min_x (f(x) + (\rho/2)||Ax + Bz^k - c + u^k||_2^2) \\
z^{k+1} &= \arg\min_z (g(z) + (\rho/2)||Ax^{k+1} + Bz - c + u^k||_2^2) \\
u^{k+1} &= u^k + (Ax^{k+1} + Bz^{k+1} - c) \qquad (1.23)
\end{aligned}$$

The two forms (1.21 and 1,23) are clearly equivalent, but the formulas in the scaled form of ADMM are often shorter and more convenient to solve than in the unscaled form. At each iteration, we update the termination threshold variables. For instance, we define primal residual $r^{k+1} = Ax^{k+1} + Bz^{k+1} - c$, and dual residual $s^{k+1} = \rho A^T B(z^{k+1} - z^k)$. The program terminates if the following conditions are satisfied

- $||r^k||_2 \leq \epsilon^{pri} = \sqrt{p}\epsilon^{abs} + \epsilon^{rel}max\{||Ax^k||_2, ||Bz^k||_2, ||c||_2\}$

- $||s^k||_2 \leq \epsilon^{dual} = \sqrt{n}\epsilon^{abs} + \epsilon^{rel}||A^Ty^k||_2$

where $\sqrt{p}, \sqrt{n}$ is the square root of the dimension of c, A^Ty^k respectively. We define $\epsilon^{abs} = 10^{-2}$ and $\epsilon^{rel} = 10^{-4}$ in our program for computational time efficiency.

1.3 My Contribution

As the sparse linear regression model estimation becomes more important, the least square regression with lasso penalty gets the attention from many computing scientists and statisticians. The concept of model selection oracle theory illustrates the optimal behavior during the model selection procedure that is introduced by Fan and Li [5]. However the limitation of least square oracle theory occurs when the error variance is infinite, in which case Zou and Yuan [26] establish the composite quantile regression with adaptive lasso penalty that maintains the oracle model selection theory. The superior property of composite quantile regression with adaptive lasso penalty attracts many researchers. However, the non-differentiable loss function and regularization keep many researchers away from this area. **R** package such as **quantreg** [12] is developed and well known to estimate and determine the significant parameters of the quantile regression with or without lasso penalty. However, it does not handle the composite quantile regression with or without regularization. The same happened for the **R** package **CDLasso** [8], which implements the Coordinated Descent (CD) method for LAD and LAD lasso. The lack of program nor package for composite quantile regression with or without regularization motivates our research here.

In this thesis, we implement three algorithms and use Interior Point algorithm for quantile, composite quantile regression with or without regularization

1. Interior Point (IP) [14] method, which starts from interior point towards minimum. It is worth noticing that we transform quantile, composite quantile regression with or without regularization problem into the linear programing format. In that case, we use the linear programing solver called **Rmosek** [6] for the estimation. We briefly introduce the IP algorithm in Section 1.2 but details in Chapter 2.

2. Majorize and Minimize (MM) [10], which majorize both loss function and penalty function into differentiable smooth function and then minimize them through Newton's method. We give a brief introduction of MM algorithm in the Section 1.2 but details in Chapter 2.

3. Coordinate Descent (CD) [24], which is the greedy algorithm for each parameters. The basic idea of CD algorithm could be found in Section 1.2 but details in Chapter 2.

4. Alternating Direction Method of Multiplier (ADMM) [1], which is well known as convex optimization algorithm. We give a brief introduction of ADMM algorithm in Section 1.2 but details in Chapter 2.

The global convergence result is proved by the convexity of the loss function and regularization. In Chapter 3, we conduct the simulation to compare the performances of each algorithms in time efficiency and estimation accuracy. Due to superior performance of our program, we present a fresh **R** package named **cqrReg** [7] (Composite Quantile Regression with regularization), which is coded in **C++** and linked back to R by an user-friendly interface. The numerical simulation result shows that **cqrReg** [7] package is efficient and consistent in large data sets.

Chapter 2

Algorithms

2.1 Quantile Regression

We recall the definition of quantile regression as deriving any vector $\beta \in \mathbb{R}^P$ that minimizes the loss function

$$L(\beta) = \rho_\tau(Y - X\beta - u)$$

for easy notation, we define $r = Y - X\beta - u$.

2.1.1 IP Algorithm

The quantile regression problem could be formulated as the linear program:

$$\min_{u,v} \; \{\tau e^T u + (1-\tau)e^T v \;\mid\; Y = X\beta + u - v, \;\; (u,v) \in \mathbb{R}_+^{2n}\} \tag{2.1}$$

which has the dual form

$$\max_{d} \ \{Y^T d | X^T d = 0, \ \ d \in [\tau - 1, \tau]^n\} \tag{2.2}$$

where e is $N \times 1$ vector of one. We use the **Rmosek** [6] package for solving linear programing problem in the following format

$$\begin{aligned} \text{min} \quad & \{c^T x + c_0\} \\ \text{subject \ to} \quad & l^c \leq Ax \leq u^c \\ & l^x \leq x \leq u^x \end{aligned}$$

where $x \in \mathbb{R}^n$, the constraint matrix A $\in \mathbb{R}^{m \times n}$, the objective coefficients c $\in \mathbb{R}^n$, objective constant $c_0 \in \mathbb{R}$, lower and upper constraint bounds $l^c \in \mathbb{R}^m$and $u^c \in \mathbb{R}^m$, and lower and upper variable bounds $l^x \in \mathbb{R}^n$ and $u^x \in \mathbb{R}^n$. We formulate the quantile regression into the following linear programing form:

$$\begin{aligned} \text{min} \quad & \{\tau e^T u + (1 - \tau) e^T v\} \\ \text{subject \ to} \quad & Y = X\beta + u - v \\ \text{having \ \ bounds} & -\infty \leq \beta \leq \infty \\ & 0 \leq u \leq \infty \\ & 0 \leq v \leq \infty \end{aligned}$$

In order to match the input format of package, we define $x = (\beta, u, v)$, $c = (0_{1\times p}, \tau_{1\times n}, (1-\tau)_{1\times n})$, $A = (X, I_{n\times n}, -I_{n\times n})$, $l^c = u^c = y$, $l^x = (-\infty_{1\times p+1}, 0_{1\times 2n})$, $u^x = \infty_{1\times(p+2n+1)}$.

2.1.2 MM Algorithm

Based on the idea of MM algorithm, we approximate the piecewise quantile function into the function $\rho_\tau^\epsilon(r)$, which less than $L(\beta)$ for all β. We denote the perturbation parameter $\epsilon > 0$.

$$\rho_\tau^\epsilon(r) = \rho_\tau(r) - \frac{\epsilon}{2}\ln(\epsilon + |r|) \tag{2.3}$$

In the second step, we majorize the approximation function at r^k by the quadratic function

$$\zeta_\tau^\epsilon(r|r^k) = \frac{1}{4}[\frac{(r)^2}{\epsilon + |r^k|} + (4q-2)r + c] \tag{2.4}$$

The MM algorithm operates by minimizing the majorizer

$$Q_\epsilon(\beta|\beta^k) = \zeta_\tau^\epsilon(r|r^k) \tag{2.5}$$

Before continuing to the estimation step of MM algorithm, we briefly introduce the solver for linear convex optimization problem called Newton-Raphson method. Generally, Newton-Raphson method is the high dimensional case of Newton's method, which is well known for locating the root of a function. The basic idea of Newton's method is based on the gradient of the function. In one dimensional case, for any differentiable linear function $f(x)$, we look for $\{x^*|f'(x^*) = 0\}$ where f' is the gradient function. First we pick the starting point x_k in the domain, then defined the gradient function $g_k(x) = ax + b$ at

$f(x_k)$. Newton approximates $g_k(x)$ by

$$g_k(x) \approx g_k(x_k)'x + g_k(x_k) - g_k(x_k)x_k$$

where $a \approx g_k(x_k)'$, $b \approx g_k(x_k) - g_k(x_k)x_k$. For $g_k(x_{k+1}) = 0$, we have

$$\begin{aligned} x_{k+1} &= x_k - g_k(x_k)/g_k(x_k)' \\ &= x_k - f'(x_k)/f''(x_k) \end{aligned}$$

where $f''(x_k)$ is the 2nd derivative at x_k or Hessian matrix in in high dimensional case. It has been proved that for differentiable function the Newton's method converge to the local minimum. Based on the Newton-Raphson method, we derive the iterative update schema of our problem as follow:

$$\beta^{k+1} = \beta^k + \Delta_\epsilon^k \tag{2.6}$$

where Δ_ϵ^k is the step direction

$$\begin{aligned} \Delta_\epsilon^k &= -H(f(\beta))^{-1}G(f(\beta)) & (2.7) \\ &= -[df(\beta^k)'W_\epsilon(\beta^k)df(\beta^k)]^{-1}df(\beta^k)'v_\epsilon(\beta^k)' & (2.8) \end{aligned}$$

where H is the Hessian matrix of the optimization function, G is the gradient of the optimization function. Given the first differential of $Q_\epsilon(\beta|\beta^k)$ (gradient)

$$\begin{aligned} dQ_\epsilon(\beta|\beta^k) &= \frac{1}{2}(\frac{r}{\epsilon + |r^k|} + 2q - 1)dr(\beta) \\ &= \frac{1}{2}v_\epsilon(\beta^k)df(\beta) \end{aligned}$$

where

$$v_\epsilon(\beta^k) = (1 - 2q - \frac{r_1(\beta)}{\epsilon + |r_1(\beta^k)|}, ...(1 - 2q - \frac{r_n(\beta)}{\epsilon + |r_n(\beta^k)|})$$

Given the second derivative of $Q_\epsilon(\beta|\beta^k)$ (Hessian matrix)

$$d^2 Q_\epsilon(\beta|\beta^k) \approx \frac{1}{2} df(\beta^k)' W_\epsilon(\beta^k) df(\beta^k)$$

where $df(\beta)$ is the $n \times p$ matrix with entry $\frac{\partial}{\partial \beta_j} f_i(\beta)$ in ith row and jth column. In the linear regression, we have $df(\beta) = X$. For easy notation, we define $W_\epsilon(\beta^k)$ as a n by n diagonal matrix with ith diagonal entry $[\epsilon + |r_i(\beta^k)|]^{-1}$. In each iteration, the algorithm updates β by the updated scheme until satisfy the convergence criteria, which Δ_ϵ^k is less than 10^{-4}.

2.1.3 CD Algorithm

By the general idea of CD algorithm, we replace u in residuals $r = Y - X\beta - u$ for fixed β by the sampling quantile value of the numbers $z = Y - X\beta$ at τ level, which will drive $L(\beta)$ downhill. Then we could rewrite the loss function

$$L(\beta) = \sum_{i=1}^{n} |x_{ik}| |\frac{y_i - u - \sum_{j \neq k} x_{ij}\beta_j}{x_{ik}} - \beta_k| \cdot \Theta \tag{2.9}$$

we define the quantile weight vector $\Theta = \{\tau_{r>0}, (1-\tau)_{r<0}\}$ that corresponds to the quantile function. In that case, the quantile loss function transfers to the L_1 loss function that matches the original problem of CD algorithm. The second step, we sort the numbers $z_i = \frac{y_i - u - \sum_{j \neq k} x_{ij}\beta_j}{x_{ik}}$ and update parameter β by the order statistic $z_{[i]}$ whose index i satisfies (in other words weighted median)

$$\sum_{j=1}^{i-1} w_{[j]} < \frac{1}{2}\sum_{j=1}^{n} w_{[j]} \qquad \sum_{j=1}^{i} w_{[j]} \geq \frac{1}{2}\sum_{j=1}^{n} w_{[j]}$$

where $w_i = |x_{im}| \cdot \Theta_i$ is corresponded to the z_i. The algorithm starts from β_1 to β_p. In each iteration, we greedily update β_i by the optimal value that is derived from the update schema. At the end of each iteration, we check the convergence of predictor β by the average absolute value difference threshold $\phi = 10^{-4}$.

2.1.4 ADMM Algorithm

We could rewrite the quantile regression into ADMM form

$$\begin{aligned} \min \quad & \rho_\tau(z) \\ \text{subject} \quad \text{to} \quad & Y = X\beta + z \end{aligned}$$

where $f(x) = 0, g(z) = \rho_\tau(z)$. Based on the general iteration steps of ADMM, we have the iteration steps for quantile regression :

$$\begin{aligned} \beta^{k+1} &= \arg\min_{\beta} \ (\rho/2)||Y - X\beta - z^k + u^k/\rho||_2^2 \\ &= (X^T X)^{-1} X^T (y - z^k + u^k/\rho) \\ z^{k+1} &= \arg\min_{z} \ \rho_\tau(z) + (\rho/2)||Y - X\beta^{k+1} - z + u^k/\rho||_2^2 \\ &= S_{\frac{1}{\rho}}(c - \frac{2\tau - 1}{\rho}) \\ u^{k+1} &= u^k + \rho(Y - X\beta^{k+1} - z^{k+1}) \end{aligned}$$

where $c = Y - X\beta^{k+1} + u^k/\rho$, $S_a(v) = (v-a)_+ - (-v-a)_+$. In each iteration, the program terminates if the primal residual $||r^k||_2 \leq \epsilon^{pri}$ and dual residual

$||s^k||_2 \leq \epsilon^{dual}$, where

$$
\begin{aligned}
r^{k+1} &= Y - X\beta^{k+1} - z^{k+1} \\
s^{k+1} &= \rho X^T(z^{k+1} - z^k) \\
\epsilon^{pri} &= \sqrt{n}\epsilon^{abs} + \epsilon^{rel}\max\{||X\beta^{k+1}||_2^2, ||z^{k+1}||_2^2, ||Y||_2^2\} \\
\epsilon^{dual} &= \sqrt{p}\epsilon^{abs} + \epsilon^{rel}||X^T u^{k+1}||_2^2
\end{aligned}
$$

2.2 Quantile Regression with Regularization

We recall the definition of quantile regression with regularization as deriving the vector $\beta \in \mathbb{R}^P$, which minimizes the sum of the loss function and the adaptive lasso penalty function.

$$L(\beta) = \rho_\tau(Y - X\beta) + \lambda \frac{|\beta|}{|\beta^{QR}|^2}$$

where the β^{QR} is derived from the general quantile regression. For easy notation, we define the residual $r(\beta) = Y - X\beta$.

2.2.1 IP Algorithm

The quantile regression with regularization could be formulated as the linear programing problem:

$$\begin{aligned}
\text{min} \quad & \{\tau e^T u + (1-\tau) e^T v + \lambda \beta^* / \beta^{QR^2}\} \\
\text{subject to} \quad & Y = X\beta + u - v \\
& \beta \leq \beta^* \\
& -\beta \leq \beta^* \\
\text{having} \quad & \text{bounds} - \infty \leq \beta \leq \infty \\
& 0 \leq u \leq \infty \\
& 0 \leq v \leq \infty \\
& 0 \leq \beta^* \leq \infty
\end{aligned}$$

In order to match the format for **Rmosek** package, we define $x = (\beta, u, v, \beta^*)$, $c = (0_{1\times p}, \tau_{1\times n}, (1-\tau)_{1\times n}, \lambda/\beta^{QR^2})$, $A_1 = (X, I_{n\times n}, -I_{n\times n}, 0_{n\times p})$, $A_2 = (0_{p\times 1}, I_{p\times p}, 0_{p\times(2n)}, I_{p\times p})$,

$A = (A_1, A_2)^t$, $l^x = (-\infty_{1\times p+1}, 0_{1\times 2n+p},)$, $l^c = u^c = y$, $u^x = \infty_{1\times(p+2n+1)}$.

2.2.2 MM Algorithm

Recalled the approximation function $\rho_\tau(r)$, which we define in (2.1.2)

$$\rho_\tau^\epsilon(r) = \rho_\tau(r) - \frac{\epsilon}{2}\ln(\epsilon + |r|)$$

We could rewrite the original problem into approximation loss function with adaptive lasso penalty function

$$L_\epsilon(\beta) = \rho_\tau^\epsilon[Y - X\beta] + \lambda\frac{|\beta|}{|\beta^{QR}|^2}$$

Defined $p(\beta) = |\beta|$ as penalty function. Suppose that we are given an initial value β^0. If β_j^0 is very closs to 0,then set $\hat{\beta}_j = 0$; other wise, the penalty function is locally approximated by a quadratic function using:

$$\frac{\partial}{\partial\beta_j}\{p(|\beta_j|)\} = p'(|\beta_j|+)sgn(\beta_j) \approx \frac{p'(|\beta_j^0|+)}{|\beta_j^0 + \epsilon'|}\beta_j$$

when $\beta_j^0 \neq 0$. In other words,

$$p(|\beta_j|) \approx p(|\beta_j^0|) + \frac{[\beta_j^2 - {(\beta_j^{(0)})}^2]p'(|\beta_j^0|+)}{2|\beta_j^0 + \epsilon'|}$$

Majorize at r^k by the quadratic function

$$\zeta_\tau^\epsilon(r|r^k) = \frac{1}{4}[\frac{(r)^2}{\epsilon + |r^k|} + (4q-2)r + c] + p(\beta)$$

The MM algorithm operates by minimizing the majorizer

$$Q_\epsilon(\beta|\beta^k) = \zeta_\tau^\epsilon(r|r^k) + \frac{\lambda}{|\beta^{QR}|^2}\{p(|\beta^0|) + \frac{[\beta^2 - {(\beta^{(0)})}^2]p'(|\beta^0|+)}{2|\beta^0 + \epsilon'|}\}$$

Based on the Newton-Raphson method, we derive the iterative update schema as follow:

$$\beta^{k+1} = \beta^k + \Delta_\epsilon^k$$

where Δ_ϵ^k is the Gaussian step direction

$$\Delta_\epsilon^k = -H(f(\beta))^{-1}G(f(\beta)) \tag{2.10}$$

$$= -[df(\beta^k)'W_\epsilon(\beta^k)df(\beta^k) - E_t]^{-1}\{v_\epsilon(\beta^k)df(\beta^k) - E_t\beta^k\}' \tag{2.11}$$

where H is the Hessian matrix of the optimization function, G is the gradient of the optimization function. Given the first differential of $Q_\epsilon(\beta|\beta^k)$ (gradient)

$$dQ_\epsilon(\beta|\beta^k) = \frac{1}{2}(\frac{r}{\epsilon + |r^k|} + 2q - 1]dr(\beta) + E_t\beta^k$$

$$= -\frac{1}{2}v_\epsilon(\beta^k)df(\beta) + E_t\beta^k$$

$$v_\epsilon(\beta^k) = (1 - 2q - \frac{r_1(\beta)}{\epsilon + |r_1(\beta^k)|}, ...(1 - 2q - \frac{r_n(\beta)}{\epsilon + |r_n(\beta^k)|})$$

Given the second derivative (Hessian matrix)

$$d^2Q_\epsilon(\beta|\beta^k) \approx -\frac{1}{2}df(\beta)'W_\epsilon(\beta^k)df(\beta) + E_t$$

where E_t is the diagonal matrix with ith entry $p\prime(|\beta_j^k|+)/(\epsilon\prime + |\beta_j^k|)/|\beta_j^{QR}|^2 = sign(\beta_j^k)/(\epsilon\prime + |\beta_j^k|)/|\beta_j^{QR}|^2$. The approximation parameter $\epsilon\prime = \phi \cdot p/2$ where ϕ

is the tolerance or termination threshold value. And $df(\beta)$ is the $n \times p$ matrix with entry $\frac{\partial}{\partial \beta_j} f_i(\beta)$ in row i and column j, where in linear regression $df(\beta)$ is X. $W_\epsilon(\beta^k)$ is an n dimensional diagonal matrix with ith diagonal entry $[\epsilon + |r_i(\beta^k)|]^{-1}$. In each iteration, the algorithm updates β by following the updated scheme until satisfies the convergence criteria, which Δ_ϵ^k is less than termination threshold.

2.2.3 CD Algorithm

Based on the CD algorithm, we define $r = y - X\beta - u$. In each iteration, we replace u for fixed β by the sampling quantile of the numbers $z = Y - X\beta$, which will drives $L(\beta)$ downhill. Then we could rewrite the loss function for each β_m

$$L(\beta) = \sum_{i=1}^{n} |x_{im}|| \frac{y_i - u - \sum_{j \neq m} x_{ij}\beta_j}{x_{im}} - \beta_m| \cdot \Theta + \frac{\lambda}{|\beta_m^{QR}|^2} |0 - \beta_m| \qquad (2.12)$$

where is $\Theta = \{\tau_{r>0}, (1-\tau)_{r<0}\}$ Sort the numbers $\{z_i = \frac{y_i - u - \sum_{j \neq k} x_{ij}\beta_j}{x_{ik}}, 0\}$ update parameter β by replace β by the order statistic $z_{[i]}$ whose index i satisfies (in other words weighted median)

$$\sum_{j=1}^{i-1} w_{[j]} < \frac{1}{2} \sum_{j=1}^{n} w_{[j]} \qquad \sum_{j=1}^{i} w_{[j]} \geq \frac{1}{2} \sum_{j=1}^{n} w_{[j]}$$

where $w_i = \{|x_{im}| \cdot \Theta, \frac{\lambda}{|\beta_m^{QR}|^2}\}$ is corresponding to the z_i. The iteration schema is the same as for quantile regression.

2.2.4 ADMM Algorithm

Before we illustrate the details of implementing ADMM algorithm for quantile regression with regularization, we need to present the ADMM algorithm for ordinary least square regression with lasso penalty function. Because the update step of quantile regression with regularization involves with the lasso problem. we recall the ordinary least square regression with lasso penalty function as deriving the vector $\beta \in \mathbb{R}^P$, which minimizes the following function

$$L(\beta) = ||X\beta - Y||_2^2 + \lambda||\beta||_1 \tag{2.13}$$

we could rewrite the it into ADMM form,

$$\begin{aligned} \text{min} \quad & (1/2)||X\beta - Y||_2^2 + \lambda||z||_1 \\ \text{subject to} \quad & \beta - z = 0 \end{aligned}$$

Iteration step:

$$\begin{aligned} \beta^{k+1} &= (X^T X + \rho I_*)^{-1}(X^T Y + \rho z_*^k - u_*^k) \\ z^{k+1} &= S_{\frac{\lambda}{\rho}}(\beta_*^{k+1} + u^k/\rho) \\ u^{k+1} &= u^k + \rho(\beta_*^{k+1} - z^{k+1}) \end{aligned}$$

where β_* is β without intercept, $z_* = (0, z)$, $I_* = (0, I)$, and $u_* = (0, u)$

Termination criteria: In each iteration, the program terminates if the primal residual $||r^k||_2 \leq \epsilon^{pri}$ and dual residual $||s^k||_2 \leq \epsilon^{dual}$

where

$$
\begin{aligned}
&r^{k+1} = \beta^{k+1} - z^{k+1} \\
&s^{k+1} = \rho(-1)(z^{k+1} - z^{k}) \\
&\epsilon^{pri} = \sqrt{n}\epsilon^{abs} + \epsilon^{rel} \max\{||\beta_*^{k+1}||_2^2, || - z^{k+1}||_2^2\} \\
&\epsilon^{dual} = \sqrt{p}\epsilon^{abs} + \epsilon^{rel}||u^{k+1}||_2^2
\end{aligned}
$$

After we present the ADMM for LSE Lasso, we could rewrite the quantile regression with regularization into ADMM form

$$
\begin{array}{cl}
\text{min} & \rho_\tau(r) + \lambda \frac{|\beta_*|}{|\beta^{QR}|^2} \\
\text{subject to} & r + X\beta_* + b = Y
\end{array}
$$

Iteration step:

$$
\begin{aligned}
r^{k+1} &= \arg\min_z \ \rho_\tau(r) + (\rho/2)||Y - r - X\beta^k + u^k/\rho||_2^2 \\
&= S_{\frac{1}{\rho}}(c - \frac{2\tau - 1}{\rho}) \\
\beta^{k+1} &= \arg\min_\beta \ \lambda \frac{|\beta_*|}{|\beta^{QR}|^2} + (\rho/2)||Y - r^{k+1} - X\beta + u^k/\rho||_2^2 \\
&= 1/2||Y - X\beta - r^{k+1} + u^k/\rho||_2^2 + \lambda \frac{|\beta_*|}{|\beta^{QR}|^2\rho} \\
&= \text{LSE}(X, Y - r^{k+1} + u^k/\rho, \lambda \frac{|\beta_*|}{|\beta^{QR}|^2\rho}) \\
u^{k+1} &= u^k + \rho(Y - r^{k+1} - X\beta^{k+1})
\end{aligned}
$$

where $c = Y - X\beta^k + u^k/\rho$, β_* is β without intercept, LSE means we translate the problem into least square estimation with lasso penalty as we describe earlier in this section. In each iteration, the program terminates if the primal residual $||r^k||_2 \leq \epsilon^{pri}$ and dual residual $||s^k||_2 \leq \epsilon^{dual}$

where

$$\begin{aligned}
p^{k+1} &= Y - r^{k+1} - X\beta^{k+1} \\
s^{k+1} &= \rho X_*(\beta_*^{k+1} - \beta_*^k) \\
\epsilon^{pri} &= \sqrt{n}\epsilon^{abs} + \epsilon^{rel} \max\{|| - r^{k+1}||_2^2, || - X_*\beta_*^{k+1}||_2^2, || - Y + b||_2^2\} \\
\epsilon^{dual} &= \sqrt{n}\epsilon^{abs} + \epsilon^{rel}|| - u^{k+1}||_2^2
\end{aligned}$$

2.3 Composite Quantile Regression

We consider composite quantile regression as following

$$(\hat{b_1}, ...\hat{b_k}, \hat{\beta}^{CQR}) = \arg\min_{\beta} \sum_{i=1}^{k} \rho_{\tau_i}(Y - b_k - X\beta)$$

where $0 < \tau_1 < \tau_2 < \tau_3 ... < \tau_k < 1$, b_k is the kth quantile level intercept coefficient.

2.3.1 IP Algorithm

The composite quantile regression problem could be formulated as the linear program:

$$\begin{aligned}
&min \qquad \{\sum_{k=1}^{K} \tau_k e^T u_k + (1-\tau_k) e^T v_k\} \\
&\text{subject to} \quad Y = X\beta + u_k - v_k + b_k \\
&having\ bounds - \infty \leq \beta \leq \infty \\
&\qquad\qquad\qquad\quad 0 \leq u_k \leq \infty \\
&\qquad\qquad\qquad\quad 0 \leq v_k \leq \infty \\
&\qquad\qquad\qquad -\infty \leq b_k \leq \infty
\end{aligned}$$

In order to match the format in **Rmosek** package, we define $x = (b_1, ..., b_K, \beta, u_1, ...u_K, v_1, ..., v_K)$, $u_i = (u_{i1}, ..., u_{in})$, $v_i = (v_{i1}, ..., v_{in})$, $c = (0^{1\times(p+K)}, \tau_1^{1\times n}, ..., \tau_K^{1\times n}, (1-\tau_1)^{1\times n}, ..., (1-\tau_K)^{1\times n})$, $A_i = (0^{n\times(i-1)}, e^{n\times 1}, 0^{n\times(K-i)}, X, I_1^{n\times n}, -I_2^{n\times n})$, $A = (A_1, ..., A_K)^t$, $l^c = u^c = (y, \underset{K}{...}, y)$, $l^x = (-\infty_{1\times(p+K)}, 0_{1\times 2nK})$, $u^x = \infty_{1\times(p+2nK+K)}$.

2.3.2 MM Algorithm

Since the majorize step is the same as quantile regression, we only illustrate the iterative update schema here. In each iteration, we update

$$\beta^{k+1} = \beta^k + \Delta_\epsilon^k$$

where Δ_ϵ^k is the Gaussian step direction

$$\begin{aligned} \Delta_\epsilon^k &= -H(f(\beta))^{-1}G(f(\beta)) \\ &= -[\sum_{j=1}^{K} df(\beta^k)' W_\epsilon^j(\beta^k) df(\beta^k)]^{-1}\{\sum_{j=1}^{K} v_\epsilon^j(\beta^k) df(\beta^k)\}' \\ v_\epsilon^j(\beta^k) &= (1 - 2\tau_j - \frac{r_{1j}(\beta)}{\epsilon + |r_{1j}(\beta^k)|}, ..., (1 - 2\tau_j - \frac{r_{nj}(\beta)}{\epsilon + |r_{nj}(\beta^k)|})) \end{aligned}$$

where H is the Hessian matrix of the optimization function, G is the gradient of the optimization function. We define $W_\epsilon^j(\beta^k)$ as a n by n diagonal matrix with ith diagonal entry $[\epsilon + |r_i^j(\beta^k)|]^{-1}$ where the $r_i^j(\beta^k)$ is the ith residual for quantile level τ_j. And $df(\beta)$ is the $n * p$ matrix with entry $\frac{\partial}{\partial \beta_j} f_i(\beta)$ in ith row and jth column. In the linear regression, we have $df(\beta) = X$. For easy notation, In each iteration, the algorithm updates β follow the updated scheme until satisfy the convergence criteria, which Δ_ϵ^k is less than 10^{-6}.

2.3.3 CD Algorithm

Based on the CD algorithm, we define $r = Y - X\beta - u$. In each iteration, we replace b_k for fixed β by the sampling quantile levels of the numbers $z = Y - X\beta$, which will drives loss function downhill. Then we could rewrite the loss function

$$L(b_1, \ldots, b_k, \beta) = \sum_{l=1}^{k} \sum_{i=1}^{n} |x_{im}|| \frac{y_i - b_l - \sum_{j \neq m} x_{ij}\beta_j}{x_{im}} - \beta_m| \cdot \Theta_l$$

where $\Theta_l = \{\tau^l_{r>0}, (1-\tau)^l_{r<0}\}$. We sort the numbers $z_{il} = \frac{y_i - b_l - \sum_{j \neq m} x_{ij}\beta_j}{x_{im}}$, which are $n \cdot k$ numbers. In the 2nd step, we update the parameter β by the order statistic $z_{[i]}$ whose index i satisfies (in other words weighted median)

$$\sum_{j=1}^{i-1} w_{[j]} < \frac{1}{2} \sum_{j=1}^{n} w_{[j]} \qquad \sum_{j=1}^{i} w_{[j]} \geq \frac{1}{2} \sum_{j=1}^{n} w_{[j]}$$

where $w_{il} = |x_{im}| \cdot \Theta_{il}$ is corresponding to the z_{il}. The iteration schema is the same as for quantile regression.

2.3.4 ADMM Algorithm

we could rewrite the composite quantile regression into ADMM form,

$$\begin{aligned} \text{min} \quad & \rho_\tau(z) \\ \text{subject to} \quad & X^*\beta + z = Y^* \end{aligned}$$

where $f(x) = 0, g(z) = \rho_\tau(z)$,

$$X^*_{(n \cdot k) \times (p+k)} = \begin{bmatrix} (1\,0\,0 \cdots 0)_{n \cdot k}\ X \\ (0\,1\,0 \cdots 0)_{n \cdot k}\ X \\ (0\,0\,1 \cdots 0)_{n \cdot k}\ X \\ \vdots\ \vdots\ \vdots\ \vdots \quad \vdots \quad \vdots \\ (0\,0\,0\ \ldots\ 1)_{n \cdot k}\ X \end{bmatrix} \quad Y^*_{n \cdot k} = \begin{bmatrix} Y \\ Y \\ Y \\ \vdots \\ Y \end{bmatrix}$$

where $X^*_{(n\cdot k)\times(p+k)}$ is k number of original design matrix stack up with independent intercept coefficients, and $Y^*_{n\cdot k}$ is k number of Y stack up. In that case, we have the same iterative update schema and termination criteria as quantile regression.

2.4 Composite Quantile Regression with Regularization

We recall the definition of composite quantile regression with regularization as estimation procedure for $\beta \in \mathbb{R}^P$.

$$(\hat{b_1}, ...\hat{b_k}, \hat{\beta}^{CQRL}) = \arg\min_{\beta} \sum_{i=1}^{k} \rho_{\tau_i}(Y - b_k - X\beta) + \lambda \frac{|\beta|}{|\beta^{CQR}|^2} \tag{2.14}$$

where $0 < \tau_1 < \tau_2 < \tau_3 ... < \tau_k < 1$ and b_k is the kth quantile intercept coefficient.

2.4.1 IP Algorithm

The composite quantile regression with regularization could be formulated as the linear program:

$$\begin{aligned}
\text{min} \qquad & \{\sum_{k=1}^{K}(\tau_k e^T u_k + (1-\tau_k)e^T v_k) + \lambda|\beta^*/\beta^{CQR^2}|\} \\
\text{subject to} \qquad & Y = X\beta + u_k - v_k + b_k \\
& \beta \leq \beta^* \\
& -\beta \leq \beta^* \\
\text{having} \quad & \text{bounds} - \infty \leq \beta \leq \infty \\
& -\infty \leq b_k \leq \infty \\
& 0 \leq u \leq \infty \\
& 0 \leq v \leq \infty \\
& 0 \leq \beta^* \leq \infty
\end{aligned}$$

In order to match the format in **Rmosek** package, we define

$x = (b_1, ..., b_K, \beta, u_1, u_2, ...u_K, v_1, v_2, ..., v_K, \beta^*)$, $u_i = (u_{i1}, ..., u_{in})$, $v_i = (v_{i1}, ..., v_{in})$,

$c = (0^{1\times(p+K)}, \tau_1^{1\times n}, ..., \tau_K^{1\times n}, (1-\tau_1)^{1\times n}, ..., (1-\tau_K)^{1\times n}, \lambda/\beta^{QR^2})$,

$A_i = (0^{n\times(i-1)}, e^{n\times 1}, 0^{n\times(K-i)}, X, I_1^{n\times n}, -I_2^{n\times n}, 0^{n\times p})$, $A_{k+1} = (0^{p\times K}, I^{p\times p}, 0_{p\times 2n}, -I^{p\times p})$,

$A_{k+2} = (0^{p\times K}, -I^{p\times p}, 0_{p\times 2n}, -I^{p\times p})$, $A = (A_1, ..., A_{K+1}, A_{K+2})^t$, $l^c = (y, ..., \underset{K}{y,} -\infty_{1\times 2p})$,

$u^c = (y, ..., \underset{K}{y,} 0_{1\times 2p})$ $l^x = (-\infty_{1\times(p+K)}, 0_{1\times 2nK+p})$, $u^x = \infty_{1\times(2p+2nK+K)}$.

2.4.2 MM Algorithm

Since the majorize step for loss function and penalty function is the same as for quantile regression with regularization, we only show iterative update schema here. In each iteration, we update

$$\beta^{k+1} = \beta^k + \Delta_\epsilon^k$$

where Δ_ϵ^k is the Gaussian step direction

$$\begin{aligned}
\Delta_\epsilon^k &= -H(f(\beta))^{-1}G(f(\beta)) \\
&= -[\sum_{j=1}^{K} df(\beta^k)'W_\epsilon^j(\beta^k)df(\beta^k) - E_t]^{-1}\{\sum_{j=1}^{K} v_\epsilon^j(\beta^k)df(\beta^k) - E_t\beta^k\}' \\
v_\epsilon^j(\beta^k) &= (1 - 2\tau_j - \frac{r_{1j}(\beta)}{\epsilon + |r_{1j}(\beta^k)|}, ..., (1 - 2\tau_j - \frac{r_{nj}(\beta)}{\epsilon + |r_{nj}(\beta^k)|}))
\end{aligned}$$

where H is the Hessian matrix of the optimization function, G is the gradient of the optimization function. We define E_t as the diagonal matrix with ith entry $p\prime(|\beta_j^k|+)/(\epsilon\prime + |\beta_j^k|)/|\beta_j^{CQR}|^2 = sign(\beta_j^k)/(\epsilon\prime + |\beta_j^k|)/|\beta_j^{CQR}|^2$, the approximation term for the penalty function $\epsilon\prime = \phi \cdot p/2$ where ϕ is the converge tolerance. $df(\beta)$ is the $n \cdot p$ matrix with entry $\frac{\partial}{\partial \beta_j} f_i(\beta)$ in row i and column j, where in linear regression $df(\beta)$ is X and $W_\epsilon^j(\beta^k)$ is an n dimensional diagonal matrix with ith diagonal entry $[\epsilon + |r_i^j(\beta^k)|]^{-1}$. In each iteration, the algorithm updates β follow the updated scheme until satisfy the convergence criteria, which Δ_ϵ^k is less than 10^{-6}.

2.4.3 CD Algorithm

We define the kth quantile residual $r(\beta) = Y - b_k - X\beta$. Then we replace b_k for fixed β by the sampling quantiles of the numbers $z = Y - X\beta$, which will drives $L(\beta)$ downhill. Then we could rewrite the loss function

$$L(b_i, \ldots, b_k, \beta) = \sum_{l=1}^{k} \sum_{i=1}^{n} |x_{im}|| \frac{y_i - b_l - \sum_{j \neq m} x_{ij}\beta_j}{x_{im}} - \beta_m| \cdot \Theta_l + \frac{\lambda}{|\beta_m^{CQR}|^2} |0 - \beta_m|$$

where is $\Theta_l = \{\tau_{r>0}^l, (1-\tau)_{r<0}^l\}$. We sort the numbers $\{z_{il} = \frac{y_i - b_l - \sum_{j \neq m} x_{ij}\beta_j}{x_{im}}, 0\}$, which are $n \cdot k$ numbers. In the 2nd step, we update the parameter β by replace β by the order statistic $z_{[i]}$ whose index i satisfies (in other words weighted median)

$$\sum_{j=1}^{i-1} w_{[j]} < \frac{1}{2} \sum_{j=1}^{n} w_{[j]} \qquad \sum_{j=1}^{i} w_{[j]} \geq \frac{1}{2} \sum_{j=1}^{n} w_{[j]} \tag{2.15}$$

where $w_{il} = \{|x_{im}| \cdot \Theta_{il}, \frac{\lambda}{|\beta_m^{CQR}|^2}\}$ is corresponding to the z_{il}. The iteration schema is the same as for quantile regression.

2.4.4 ADMM Algorithm

We could rewrite the composite quantile regression into ADMM form,

$$\begin{array}{ll} \min & \rho_\tau(r) + \lambda \frac{|\beta|}{|\beta^{CQR}|^2} \\ \text{subject to} & r + X^*\beta_* + b^* = Y^* \end{array}$$

where $f(x) = 0, g(z) = \rho_\tau(z)$,

$$X^*_{(n\cdot k)\times(p+k)} = \begin{bmatrix} (1\,0\,0\cdots 0)_{n\cdot k}\; X \\ (0\,1\,0\cdots 0)_{n\cdot k}\; X \\ (0\,0\,1\cdots 0)_{n\cdot k}\; X \\ \vdots\;\vdots\;\vdots\;\vdots\quad\vdots\quad\vdots \\ (0\,0\,0\;...\;1)_{n\cdot k}\; X \end{bmatrix} \; Y^*_{n\cdot k}= \begin{bmatrix} Y \\ Y \\ Y \\ \vdots \\ Y \end{bmatrix} \; b^*_{n\cdot k}= \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_k \end{bmatrix} \; \tau^*_{n\cdot k}= \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \vdots \\ \tau_k \end{bmatrix}$$

where $X^*_{(n\cdot k)\times(p+k)}$ is k number of original design matrix stack up with independent intercept coefficients, and $Y^*_{n\cdot k}$ is k number of Y stack up. And b_i and τ_i is the $n \times 1$ vector. We conduct the iterative update schema as follow

$$\begin{aligned} r^{k+1} &= \arg\min_r \; \rho_\tau(r) + (\rho/2)||Y^* - r - X^*\beta^k + u^k/\rho||_2^2 \\ &= S_{\frac{1}{\rho}}(c - \frac{2\tau^* - 1}{\rho}) \\ \beta^{k+1} &= \arg\min_\beta \; \frac{\lambda}{|\beta^{CQR}|^2}||\beta_*||_1 + (\rho/2)||Y^* - r^{k+1} - X^*\beta + u^k/\rho||_2^2 \\ &= 1/2||Y^* - X^*\beta - r^{k+1} + u^k/\rho||_2^2 + \frac{\lambda}{|\beta^{CQR}|^2\cdot\rho}||\beta_*||_1 \\ &= NLSE(X^*, Y^* - r^{k+1} + u^k/\rho, \frac{\lambda}{|\beta^{CQR}|^2\cdot\rho}) \\ u^{k+1} &= u^k + \rho(Y - r^{k+1} - X\beta^{k+1}) \end{aligned}$$

where $c = Y^* - X^*\beta^k + u^k/\rho$, β_* is β without intercept. NLSE is the least square error regression with lasso penalty that has the same formulation as LSE excepts we do not shrink the first k element of β. In each iteration, the program terminates if the primal residual $||r^k||_2 \leq \epsilon^{pri}$ and dual residual

$||s^k||_2 \leq \epsilon^{dual}$, where

$$r^{k+1} = Y - X\beta^{k+1} - r^{k+1}$$

$$s^{k+1} = \rho X_*(\beta_*^{k+1} - \beta_*^k)$$

$$\epsilon^{pri} = \sqrt{n * k}\epsilon^{abs} + \epsilon^{rel} \max\{|| - X_*\beta_*^{k+1}||_2^2, || - r^{k+1}||_2^2, || - Y + b^*||_2^2\}$$

$$\epsilon^{dual} = \sqrt{n * k}\epsilon^{abs} + \epsilon^{rel}||u^{k+1}||_2^2$$

Chapter 3

Simulation

In evaluating the performance of four different algorithms, we mainly focus on the following model,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where Y is the response variable vector with dimension n, and X is the design matrix with dimension $n \times p$. And $\boldsymbol{\beta}$ is the coefficient vector with dimension n and $\boldsymbol{\epsilon}$ is the error term. In the simulation study, we propose two kinds of random error ϵ, which either follow $N(0, 1)$ or t distribution with degree of freedom 3. In the following Section 3.1 and 3.2, we compare the performance of four algorithm for quantile regression and composite quantile regression with or without regularization.

3.1 Quantile Regression and Composite Quantile Regression

We set the dimension of design matrix $n = 200, 400, 600, 800, 1000, 2000$, and $p = 5$. We independently generate each row of the design matrix from multivariate normal distribution $N(0, I)$. In every simulation the true value of β is generated from the random sample of uniform variable in $(-1, 1)$. We conduct the quantile regression with $\tau = 0.3$. The performance is compared by the average absolute value difference between the estimate coefficients and true parameters, which denotes as $Error$. The average running time denote as $Time$ in seconds over 50 replications.

Table 3.1: Quantile regression with normal distributed error

(n,p)	IP		MM		CD		ADMM	
	Error	*Time*	*Error*	*Time*	*Error*	*Time*	*Error*	*Time*
(200,5)	0.08	0.002	0.060	0.0002	0.036	0.002	0.063	0.002
(400,5)	0.052	0.0022	0.051	0.0004	0.046	0.003	0.055	0.0038
(600,5)	0.043	0.0029	0.033	0.0005	0.043	0.0416	0.042	0.005
(800,5)	0.037	0.0048	0.031	0.0005	0.034	0.0046	0.035	0.006
(1000,5)	0.0336	0.0053	0.026	0.0006	0.031	0.0064	0.031	0.008
(2000,5)	0.0213	0.01	0.018	0.001	0.022	0.0096	0.022	0.013

Table 3.2: Quantile regression with t distributed distributed error

(n,p)	IP		MM		CD		ADMM	
	Error	*Time*	*Error*	*Time*	*Error*	*Time*	*Error*	*Time*
(200,5)	0.082	0.0018	0.09	0.0002	0.08	0.0018	0.087	0.002
(400,5)	0.06	0.0028	0.059	0.0004	0.06	0.004	0.065	0.003
(600,5)	0.05	0.003	0.047	0.0005	0.045	0.0038	0.053	0.005
(800,5)	0.045	0.003	0.045	0.0005	0.038	0.005	0.044	0.007
(1000,5)	0.039	0.004	0.038	0.0026	0.033	0.006	0.04	0.029
(2000,5)	0.03	0.008	0.028	0.001	0.026	0.009	0.03	0.014

Table 3.1 and 3.2 report the performance of four algorithm for the quantile regression with the error from $N(0,1)$ or t distribution with degree of freedom 3, respectively. In the following figure 3.1, we could clearly find out the four algorithm we implemented are stable in time consumption and estimation accuracy. The accuracy is increased as the number of observations in the design matrix increase. Although, the scale of two plots are different, we still could conclude the running time has the positive linear relationship with the number of events in design matrix. And MM algorithm has the superior performance in time consumption.

Table 3.3 and 3.4 reports the performance of four algorithm for the composite quantile regression model with normal distributed error from $N(0,1)$ and t distributed error with degree of freedom 3, respectively. We use the same setting for the design matrix in quantile regression, besides the composite quantile regression includes 9 quantile levels $\tau = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$. The simulation result matches our expectation in variable estimation accuracy, time consumption and durability to heavy tailed error distribution.

Table 3.3: Composite quantile regression with normal distributed error

(n,p)	IP		MM		CD		ADMM	
	Error	*Time*	*Error*	*Time*	*Error*	*Time*	*Error*	*Time*
(200,5)	0.058	0.009	0.057	0.0008	0.058	0.008	0.057	0.029
(400,5)	0.043	0.021	0.047	0.001	0.04	0.011	0.043	0.057
(600,5)	0.035	0.03	0.034	0.0012	0.039	0.017	0.034	0.088
(800,5)	0.029	0.047	0.029	0.0014	0.031	0.018	0.029	0.122
(1000,5)	0.025	0.064	0.028	0.0015	0.024	0.025	0.024	0.16
(2000,5)	0.077	0.14	0.017	0.0026	0.018	0.044	0.017	0.36

Table 3.4: Composite quantile regression with t distributed error

(n,p)	IP		MM		CD		ADMM	
	Error	*Time*	*Error*	*Time*	*Error*	*Time*	*Error*	*Time*
(200,5)	0.06	0.01	0.07	0.0009	0.074	0.009	0.07	0.029
(400,5)	0.046	0.021	0.054	0.0016	0.051	0.016	0.048	0.059
(600,5)	0.038	0.035	0.041	0.0019	0.043	0.023	0.041	0.09
(800,5)	0.037	0.047	0.037	0.0027	0.033	0.027	0.037	0.12
(1000,5)	0.032	0.067	0.032	0.0031	0.029	0.037	0.031	0.16
(2000,5)	0.023	0.159	0.021	0.0046	0.021	0.06	0.02	0.36

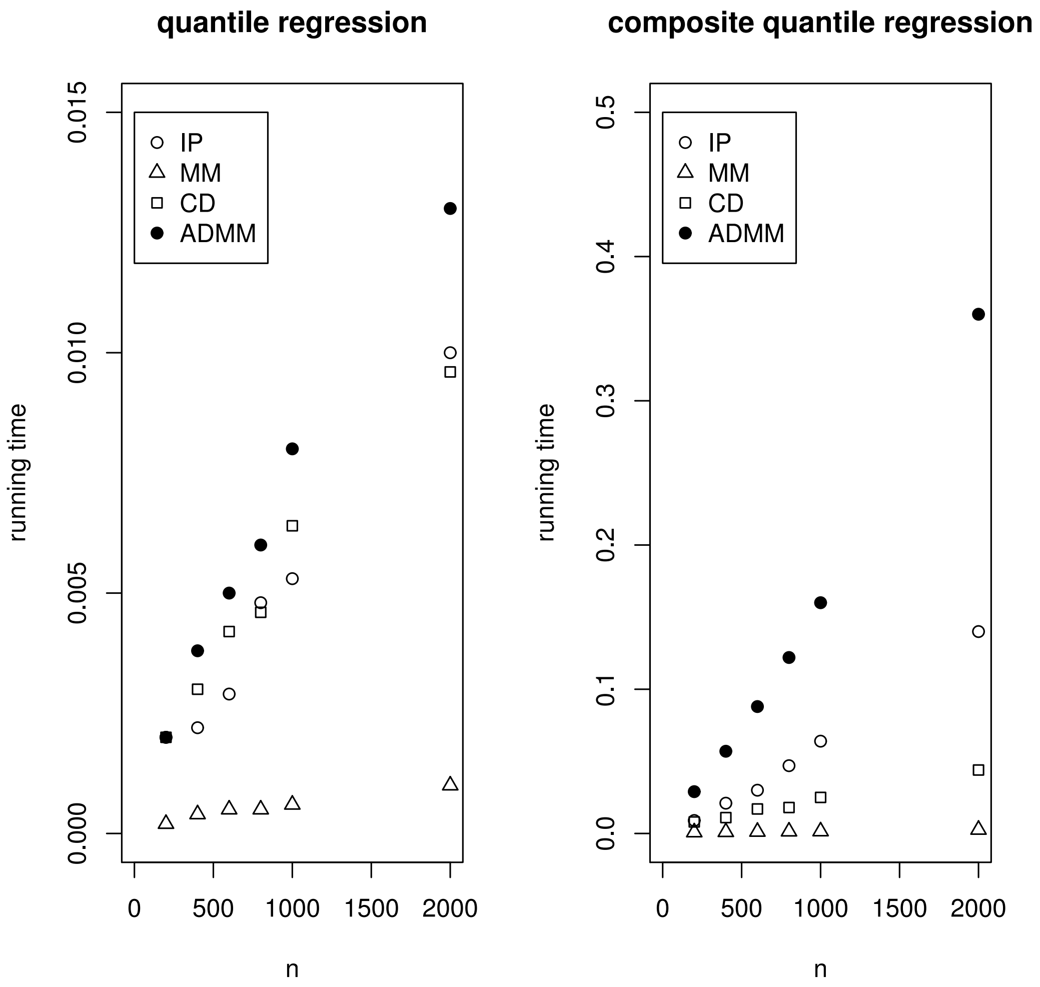

Figure 3.1: Compare the running time of four algorithm for quantile and composite quantile regression

3.2 Quantile Regression and Composite Quantile Regression with Regularization

In this section, we conduct the simulation in order to compare the performance of four algorithm in variable selection. In that case, we set $n = 100, 200, 500$, and vary p from $1.5n$ to $5n$. The regression model is generated by the same way as for quantile, composite quantile regression. In every simulation the true value of $\beta = (0, 4, 0, 6, 8, 0, 10, 0, ..., 0)$. The convergence threshold of each simulation is 10^{-4}. We set 10^{-3} threshold for non zero values, which means if the absolute value of non zero coefficient is less than 10^{-3} we set them to zero. The performance of each algorithm is compare by three index. The first one is the average numbers of false predictors selected as N_{false}, which means the estimator has non-zero value at the index that suppose to be zero. The second is the N_{true} denotes as the number of true parameters selected. The last one is the average computing time in seconds over 100 replications.

Table 3.5 and 3.6 report the quantile regression with adaptive lasso penalty, which the regression model with the error from $N(0, 1)$ or t distributed error with degree of freedom 3, respectively.

Table 3.7 and 3.8 report the composite quantile regression model with adaptive lasso penalty with $\tau = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ quantile levels. We conduct the simulation at the same setting as for quantile regression with regularization, except we did not include the interior point method. Because the **Rmosek** package is hard to install in many operation systems, and we do not include it in our package. The simulation result matcesh our expectation in variable selection accuracy, time consumption and durability to heavy tailed error distribution. However, the MM algorithm seems a little bit slow in variable selection. We believe the inversion of high dimensional design matrix is the main reason, since it take a lots floating point operations.

Table 3.5: Quantile regression with regularization and normal distributed error

(n,p)	MM			IP		
	Time	N_{true}	N_{false}	Time	N_{true}	N_{false}
(100,200)	0.1	4	0.1	0.074	4	0
(100,300)	0.25	4	0	0.024	4	0
(100,500)	0.812	3.9	0	0.98	4	0
(200,400)	0.58	4	0	0.627	4	0
(200,600)	1.64	4	0	1.96	4	0
(200,1000)	6.23	4	0	8.85	4	0
(500,750)	4.09	4	0	5.1	4	0
(500,1000)	10.3	4	0	11	4	0
(500,1500)	24	4	0	38	4	0
(n,p)	CD			ADMM		
	Time	N_{true}	N_{false}	Time	N_{true}	N_{false}
(100,200)	0.014	4	0	0.017	4	0
(100,300)	0.02	4	0	0.041	4	0
(100,500)	0.035	4	0	0.152	4	0
(200,400)	0.048	4	0	0.088	4	0
(200,600)	0.054	4	0	0.161	4	0
(200,1000)	0.11	4	0	0,791	4	0
(500,750)	0.18	4	0	0.522	4	0
(500,1000)	0.24	4	0	0.852	4	0
(500,1500)	0.36	4	0	2.41	4	0

In the following figure 3.2, we set $n = 100$, and $p = (200, 400, 500, 800, 1000)$. We compare the performance of four algorithm for composite quantile regression with regularization in quantile levels $\tau = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$. As the plot shows, the running time has a posive linear relationship to the dimension p. And the ADMM algorithm and CD algorithm have the superior time efficiency both in quantile and composite quantile regression with regularization. In addition, for composite quantile regression with regularization, the ADMM run faster than CD algorithm for $p \leq 1000$.

Table 3.6: Quantile regression with regularization with t distributed error.

(n,p)	MM			IP		
	Time	N_{true}	N_{false}	Time	N_{true}	N_{false}
(100,200)	0.11	4	0.1	0.072	4	0
(100,300)	0.23	4	0	0.026	4	0
(100,500)	0.782	3.9	0	1.01	4	0
(200,400)	0.62	4	0	0.64	4	0
(200,600)	1.46	4	0	1.83	4	0
(200,1000)	6.41	4	0	8.95	4	0
(500,750)	4.01	4	0	5.3	4	0
(500,1000)	10.5	4	0	10.4	4	0
(500,1500)	23	4	0	37.4	4	0
(n,p)	CD			ADMM		
	Time	N_{true}	N_{false}	Time	N_{true}	N_{false}
(100,200)	0.015	4	0	0.018	4	0
(100,300)	0.022	4	0	0.039	4	0
(100,500)	0.037	4	0	0.162	4	0
(200,400)	0.051	4	0	0.091	4	0
(200,600)	0.056	4	0	0.154	4	0
(200,1000)	0.12	4	0	0,781	4	0
(500,750)	0.16	4	0	0.546	4	0
(500,1000)	0.23	4	0	0.843	4	0
(500,1500)	0.37	4	0	2.34	4	0

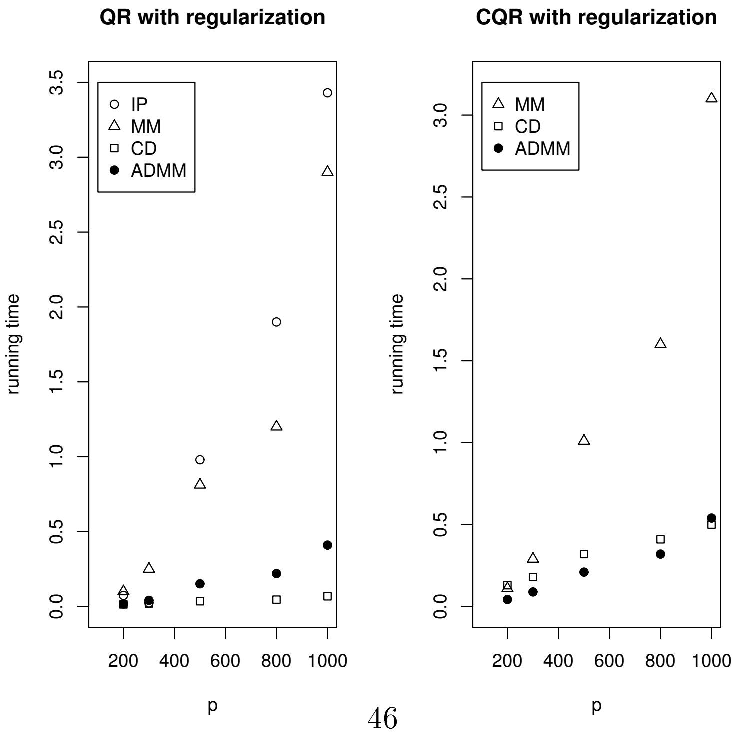

Figure 3.2: Compare the running time of four algorithm for quantile and composite quantile regression with regularization

Table 3.7: Composite quantile regression with regularization and normal distributed error

(n,p)	MM			CD			ADMM		
	Time	N_{true}	N_{false}	Time	N_{true}	N_{false}	Time	N_{true}	N_{false}
(100,200)	0.11	4	0.74	0.13	4	0	0.043	4	0
(100,300)	0.29	4	0.62	0.18	4	0	0.089	4	0
(100,500)	1.01	4	0.52	0.32	4	0	0.21	4	0
(200,400)	0.75	4	0.64	0.47	4	0	0.24	4	0
(200,600)	1.9	4	0.61	0.676	4	0	0.463	4	0
(200,1000)	7.4	4	0.25	0.615	4	0	1.52	4	0
(500,750)	5.4	4	0.6	2.4	4	0	1.56	4	0
(500,1000)	9.8	4	0.74	2.6	4	0	2.38	4	0
(500,1500)	27.5	4	0	3.6	4	0	5.67	4	0

Table 3.8: Composite quantile regression with regularization and t distributed error.

(n,p)	MM			CD			ADMM		
	Time	N_{true}	N_{false}	Time	N_{true}	N_{false}	Time	N_{true}	N_{false}
(100,200)	0.12	4	0.8	0.12	4	0	0.052	4	0
(100,300)	0.27	4	0.3	0.17	4	0	0.085	4	0
(100,500)	0.891	4	0.67	0.33	4	0	0.21	4	0
(200,400)	0.635	4	0.54	0.474	4	0	0.22	4	0
(200,600)	1.9	4	0.72	0.546	4	0	0.452	4	0
(200,1000)	7.52	4	0.25	0.621	4	0	1.41	4	0
(500,750)	5.28	4	0.8	2.38	4	0	1.52	4	0
(500,1000)	10.3	4	0.8	2.63	4	0	2.43	4	0
(500,1500)	28.5	4	0	3.8	4	0	5.86	4	0

Based on the simulation result of both quantile and composite quantile regression with or without regularization, we reach the conclusion that for large n and small p data set, MM algorithm and CD algorithm is preferred for quantile and composite quantile regression. And for high dimensional date set, CD algorithm have superior performance in time consumption, however it depends on the good initial point. In that case, the ADMM algorithm is recommend for its stability and time efficiency. It is worth noticing that in the simulation we use the IP algorithm from **quantreg** [12], because **Rmosek** is hard to install in many operation systems.

Chapter 4

Conclusion

Quantile, composite quantile regression with or without regularization have been widely studied and applied in the high-dimensional model estimation and variable selection. Fan and Li [26] propose the SCAD method to derive the oracle model estimator. However, the SCAD method has the limitation on the infinite variance of error distribution. In that case, Wu and Liu [25], Zou and Yuan [26], introduce the quantile, composite quantile regression with adaptive lasso penalty and prove its nice oracle properties, respectively. Although the theory of quantile, composite quantile regression with or without regularization have been well established, the inefficiency of the computational program leaves the challenge to many researchers. The tough part of the estimation is the non-differentiable quantile loss function and adaptive lasso function. There are four widely used algorithms for convex optimization problems including Interior Point (IP), Majorize-Minimize(MM), Coordinate Descent (CD) and Alternating Direction Method of Multipliers(ADMM). The Interior Point (IP) method has been implemented in **R** package **quantreg** [12] by Koenker [12], but the package does not include the composite quantile regression with regularization. The coordinate descent method also has been implemented in **R** package **CDLasso** [8], however it only focuses on the LAD regression with or without regularization. Unfortunately, there is no available program that has implemented neither Majorize-Minimize (MM) nor Alternating Direction Method of Multipliers(ADMM) for quantile and composite quantile regression with or without regularization.

In this thesis, we implement the above three algorithms and use Interior Point algorithm for quantile and composite quantile regression with or without regularization. The basic idea of each algorithm is covered in the Section 1.2 but the details of each algorithms in Chapter 2. In Chapter 3, we conduct the simulation to compare the performance of four algorithms in time efficiency and estimation accuracy. The simulated data is independently generated from multivariate normal distributed random samples with random errors from $N(0,1)$ or heavy tailed t distribution with degree of freedom 3. According to the simulation result, the four algorithms match our expectation both in time efficiency and estimation accuracy. The CD and ADMM have the superior performance both in variable selections and the time consumption. The simulation results of MM seems to be slow and slightly inaccurate in variable selection model. It may cause by the matrix inversion and selected approximation parameter. However, in the quantile regression and composite quantile regression with large n observation, MM has the superior performance in time consumption when compares with other algorithms. Due to the outstanding performance of our program, we publish the **R** package named **cqrReg**[7], which fulfills the need for efficient solvers for quantile, composite quantile regression with or without regularization.

Nonetheless, there still are some limitations that we need to resolve in the future work. For instance, in regularization model, we do not include the method for choosing the tuning parameter, however the user could easily conduct the cross validation to derive the suitable tuning parameter. We will continue working on implementing faster numerical methods for matrix inversion in MM. We will publish the paper of our work in the future. Overall, the **cqrReg** [7] package provides four reliable and efficient algorithms for quantile and composite quantile regression with or without regularization. The **cqrReg** [7] package is an efficient supplement to the **quantreg** [12] and **CDLasso** [8] in large data sets. We will continue maintaining and developing the package in the future.

Bibliography

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Machine Learning*, 3(1):1–122, 2010.

[2] L. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

[3] J. Claerbout and F. Muir. Robust modeling with erratic data. *Geophysics*, 38(5):826–844, 1973.

[4] F. Edgeworth. On observations relating to several quantities. *Hermathena*, 6(13):279–285, 1887.

[5] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[6] H. Friberg. *Rmosek: The R-to-MOSEK Optimization Interface*, 2014. R package version 1.2.5.1.

[7] J. Gao and L. Kong. *cqrReg: Quantile, Composite Quantile Regression and Regularized Versions*, 2015. R package version 1.2.

[8] E. Grant, K. Lange, and T. Wu. *CDLasso: Coordinate Descent Algorithms for Lasso Penalized L1, L2, and Logistic Regression*, 2013. R package version 1.1.

[9] W. Heiser. Convergent computation by iterative majorization: theory and applications in multidimensional data analysis. In *Recent advances in descriptive multivariate analysis (Exeter, 1992/1993)*, volume 2 of *Roy. Statist. Soc. Lecture Note Ser.*, pages 157–189. Oxford Univ. Press, New York, 1995.

[10] D. Hunter and K. Lange. Quantile regression via an MM algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60–77, 2000.

[11] R. Koenker. *Quantile Regression.* Cambridge University Press, 32 Avenue of the Americas, USA, 2005.

[12] R. Koenker. *quantreg: Quantile Regression*, 2015. R package version 5.11.

[13] R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

[14] R. Koenker and B. Park. An interior point algorithm for nonlinear quantile regression. *Journal of Econometrics*, 71(1):265–283, 1996.

[15] K. Lange. A gradient algorithm locally equivalent to the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 57(2):425–437, 1995.

[16] J. Ortega and W. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30 of *Classics in Applied Mathematics.* Society

for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1970 original.

[17] M. Park and T. Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

[18] S. Portnoy and R. Koenker. The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators. *Statist. Sci.*, 12(4):279–300, 1997.

[19] S. Stigler. Studies in the history of probability and statistics XL Boscovich, Simpson and a 1760 manuscript note on fitting a linear relation. *Biometrika*, 71(3):615–620, 1984.

[20] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

[21] L. Wang, M. Gordon, and J. Zhu. Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 690–700. IEEE, 2006.

[22] S. Weisberg. *Applied linear regression.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, fourth edition, 2014.

[23] S. Wright. *Primal-dual interior-point methods.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[24] T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.*, 2(1):224–244, 2008.

[25] Y. Wu and Y. Liu. Variable selection in quantile regression. *Statist. Sinica*, 19(2):801–817, 2009.

[26] H. Zou and M. Yuan. Composite quantile regression and the oracle model selection theory. *Ann. Statist.*, 36(3):1108–1126, 2008.