

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Robust Classification and Regression

Permalink

<https://escholarship.org/uc/item/99b2s59x>

Author

Li, Hanbo

Publication Date

2018

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Robust Classification and Regression

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Hanbo Li

Committee in charge:

Professor Jelena Bradic, Chair
Professor Ery Arias-Castro
Professor Sanjoy Dasgupta
Professor Yoav Freund
Professor Rayan Saab

2018

Copyright
Hanbo Li, 2018
All rights reserved.

The dissertation of Hanbo Li is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2018

DEDICATION

To my family. To the past five years.

EPIGRAPH

A very great deal more truth can become known than can be proven.

—Richard Feynman

TABLE OF CONTENTS

| | | |
|--|---|----------|
| Signature Page | | iii |
| Dedication | | iv |
| Epigraph | | v |
| Table of Contents | | vi |
| List of Figures | | ix |
| List of Tables | | xi |
| Acknowledgements | | xii |
| Vita | | xiv |
| Abstract of the Dissertation | | xv |
| Chapter 1 | Boosting in the Presence of Outliers: Adaptive Classification with Non-convex Loss Functions | 1 |
| 1.1 | Introduction | 1 |
| 1.1.1 | Literature Review | 3 |
| 1.2 | Methodology of the ArchBoost | 6 |
| 1.3 | Robust non-convex loss functions | 11 |
| 1.4 | Theoretical Considerations | 14 |
| 1.4.1 | Numerical convergence | 14 |
| 1.4.2 | Robustness | 16 |
| 1.5 | Numerical Experiments | 21 |
| 1.5.1 | Gaussian - Student Mixture | 21 |
| 1.5.2 | Comparison with Non-convex Gradient Boost | 23 |
| 1.5.3 | Consistency | 24 |
| 1.5.4 | The Long/Servedio problem | 24 |
| 1.5.5 | Outlier detection | 25 |
| 1.5.6 | Real data application | 26 |
| 1.6 | Proof of Theorems | 29 |
| 1.6.1 | Proof of Theorem 1 | 29 |
| 1.6.2 | Proof of Theorem 3 | 32 |
| 1.7 | Acknowledgement | 33 |

| | | |
|-----------|---|----|
| Chapter 2 | Forest-type Regression with General Losses and Robust Forest | 34 |
| | 2.1 Introduction | 34 |
| | 2.1.1 Random forest | 36 |
| | 2.1.2 Adaptive nearest neighbors | 37 |
| | 2.2 General framework for forest-type regression | 38 |
| | 2.2.1 Squared error and random forest | 38 |
| | 2.2.2 Extension to general loss | 39 |
| | 2.2.3 Quantile loss and quantile random forest | 40 |
| | 2.3 Robust forest | 41 |
| | 2.3.1 Huber loss | 41 |
| | 2.3.2 Tukey's biweight | 44 |
| | 2.4 Truncated squared loss and nearest neighbors | 45 |
| | 2.4.1 Truncated squared error | 45 |
| | 2.4.2 Random Forest Nearest Neighbors | 46 |
| | 2.5 Experiments | 47 |
| | 2.5.1 One dimensional toy example | 48 |
| | 2.5.2 Multivariate example | 48 |
| | 2.5.3 Nearest neighbors | 50 |
| | 2.5.4 Real data | 50 |
| | 2.5.5 Conclusion and discussion | 51 |
| | 2.6 Proof of Lemma 5 | 51 |
| | 2.7 Acknowledgement | 52 |
| Chapter 3 | Forest-type Quantile Regression for Random Censored Data | 57 |
| | 3.1 Introduction | 57 |
| | 3.1.1 Random forest weights | 61 |
| | 3.1.2 Forest-type regression framework | 62 |
| | 3.2 Methodology | 63 |
| | 3.2.1 Motivation | 63 |
| | 3.2.2 Full model | 65 |
| | 3.2.3 Estimation of survival function $G(q x)$ | 66 |
| | 3.3 Theory | 68 |
| | 3.3.1 Time complexity | 68 |
| | 3.3.2 Consistency | 69 |
| | 3.4 Simulation Study | 73 |
| | 3.4.1 Toy example | 74 |
| | 3.4.2 One-dimensional AFT model | 75 |
| | 3.4.3 Comparison of different conditional survival estimators | 77 |
| | 3.4.4 One-dimensional sine function model | 79 |
| | 3.4.5 Multi-dimensional AFT model results | 83 |
| | 3.4.6 Multi-dimensional complex manifold | 84 |
| | 3.4.7 Node size | 85 |
| | 3.5 Real Data | 87 |

| | | | |
|--------------|-------|---|-----|
| | 3.5.1 | Prediction Intervals | 89 |
| | 3.6 | Acknowledgement | 91 |
| Chapter 4 | | Estimation and Inference for High-dimensional Left-censored Quantiles . | 93 |
| | 4.1 | Introduction | 93 |
| | | 4.1.1 Contributions | 93 |
| | | 4.1.2 Related Work | 94 |
| | | 4.1.3 Content | 95 |
| | 4.2 | Methodology | 95 |
| | | 4.2.1 Model Description | 95 |
| | | 4.2.2 Initial Estimator | 96 |
| | | 4.2.3 Bias Correction | 98 |
| | | 4.2.4 Inverse Hessian Estimator: Nodewise Lasso | 101 |
| | 4.3 | Theoretical Considerations | 103 |
| | | 4.3.1 Distribution and Density Estimators | 103 |
| | | 4.3.2 Consistency of Initial Estimator | 104 |
| | | 4.3.3 Asymptotic Normality of One-step Penalized Estimator . . . | 107 |
| | 4.4 | Numerical Experiments and Application | 109 |
| | | 4.4.1 Further Details of Algorithm 6 and 7 | 109 |
| | | 4.4.2 Simulation Data | 111 |
| | | 4.4.3 Real Data | 119 |
| | 4.5 | Lemmas | 121 |
| | 4.6 | Proofs of Lemmas | 124 |
| | 4.7 | Proofs of Theorems | 140 |
| | 4.8 | Acknowledgement | 147 |
| Bibliography | | | 148 |

LIST OF FIGURES

| | | |
|-------------|--|----|
| Figure 1.1: | AdaBoost, LogitBoost, ArchBoost loss functions and the corresponding normalized weight updating rules. | 7 |
| Figure 1.2: | γ -robust losses, ϕ_γ and the corresponding normalized weight updating rules. | 13 |
| Figure 1.3: | Comparison of average test errors of ARB- γ , AdaBoost and LogitBoost. . . | 22 |
| Figure 1.4: | Comparisons with Non-convex Gradient Boost and Consistency | 23 |
| Figure 1.5: | Comparison of ARB- γ on Long/Servedio problem with different ϵ | 25 |
| Figure 2.1: | In the first row, we compare squared error loss $\frac{1}{2}x^2$ and pseudo-Huber loss with different δ . In the second row, we plot the scaling factor (2.12) of Huber loss. | 53 |
| Figure 2.2: | We plot the scaling factor (2.13) of Tukey's biweight. Compared to Huber scaling factor (see (2.12)), it has a hard threshold at δ | 54 |
| Figure 2.3: | One dimensional comparison of random forest, quantile random forest, Huber forest and Tukey forest. All forests are ensemble of 500 regression trees and the maximum number of points in terminal nodes is 20. | 55 |
| Figure 2.4: | The performance of k -RFNN against the number of nearest neighbors. . . . | 56 |
| Figure 3.1: | Sample loss plots when $\tau = 0.5$. In the first row, we have sample sizes 100 and 500. In the second row, we have sample sizes 1000 and 5000. The black curve is $U_1(q)$, the red curve is $U_2(q)$, the black (red) dotted vertical line is the root of $U_1(q)$ ($U_2(q)$), and the blue vertical line is $q = \tau$ | 75 |
| Figure 3.2: | One-dimensional AFT model results. In (a), black points stand for observation that are not censored; red points are observations that are censored, and the green points are the counterpart of the red points, that is, they are the latent values of those red points if they were not censored. | 78 |
| Figure 3.3: | AFT 1D box plots. | 79 |
| Figure 3.4: | Comparison of different conditional survival estimators for the one-dimensional AFT model. The censoring variable $C \sim \text{Exp}(\lambda = 0.08)$, and the average censoring rate is around 20%. From left-most column to right-most column, we plot the conditional survival estimators for four test points, $x = 0.4, 0.8, 1.2, 1.6$. | 80 |
| Figure 3.5: | Comparison of different conditional survival estimators for the one-dimensional AFT model. The censoring variable $C \sim \text{Exp}(\lambda = 0.20)$, and the average censoring rate is around 50%. From left-most column to right-most column, we plot the conditional survival estimators for four test points, $x = 0.4, 0.8, 1.2, 1.6$. | 81 |
| Figure 3.6: | One-dimensional Sine model. In (a), black points stand for observation that are not censored; red points are observations that are censored, and the green points are the counterpart of the red points, that is, they are the latent values of those red points if they were not censored. | 82 |
| Figure 3.7: | Sine 1D box plots. | 83 |
| Figure 3.8: | AFT multi-dimensional results. | 84 |
| Figure 3.9: | Quantile losses of Sine model with different node sizes. | 86 |

| | | |
|--------------|--|-----|
| Figure 3.10: | Quantile losses of 1D AFT model with different node sizes. | 86 |
| Figure 3.11: | Quantile losses of multi-dimensional AFT model with different node sizes. | 87 |
| Figure 3.12: | Quantile losses of multi-dimensional complex model with different node sizes. | 87 |
| Figure 3.13: | Quantile losses on real data. | 89 |
| Figure 3.14: | Quantile losses of different node sizes on real data. | 90 |
| Figure 3.15: | Prediction intervals of Sine. | 91 |
| Figure 3.16: | Confidence intervals with different node sizes on real data. | 91 |
| Figure 4.1: | $\tau = 0.4$ comparative boxplots of the average interval length (with true F_0 and true f_0). | 112 |
| Figure 4.2: | $\tau = 0.7$ comparative boxplots of the average interval length (with true F_0 and true f_0). | 113 |
| Figure 4.3: | $\tau = 0.4$ comparative boxplots of the average interval length (with estimated \hat{F}_n and true \hat{f}_n). | 116 |
| Figure 4.4: | $\tau = 0.7$ comparative boxplots of the average interval length (with estimated \hat{F}_n and true \hat{f}_n). | 117 |
| Figure 4.5: | Power curve of signal (left) and noise (right) variables under normal errors. | 118 |
| Figure 4.6: | Power curve of signal (left) and noise (right) variables under Student's t_4 errors. | 118 |

LIST OF TABLES

| | | |
|------------|--|-----|
| Table 1.1: | The list of commonly used loss functions and its corresponding F^* | 6 |
| Table 1.2: | The list of commonly used loss functions and their weak hypotheses h | 11 |
| Table 1.3: | Long/Servedio problem | 25 |
| Table 1.4: | Outliers detection. The x-axis stands for the index of the training points ranging from 1 to 2000, and the y-axis stands for the times a point is misclassified, ranging from 0 to 800. | 26 |
| Table 1.5: | Comparison of the average test errors and sample deviation (over 100 repetitions and using five-fold cross-validation) of four algorithms on the Wisconsin breast cancer dataset. | 27 |
| Table 1.6: | Comparison of the average test errors and sample deviation (over 100 repetitions and using five-fold cross-validation) of four algorithms on the GSE20194 gene dataset. | 28 |
| Table 1.7: | Comparison of the average test errors and sample deviation (over 100 repetitions) of four algorithms on the Sensorless drive diagnosis dataset. | 28 |
| Table 2.1: | Comparison of random forest (RF), quantile random forest (QRF), Huber forest (Huber) and Tukey forest (Tukey) on one dimensional example. | 48 |
| Table 2.2: | Comparison of the four methods in the setting (1). The average MSE is reported in first row, and average MAD in second row. | 49 |
| Table 2.3: | Comparison of the four methods in the setting (2). | 49 |
| Table 2.4: | Comparison of the four methods on two UCI repository datasets: (1) concrete compressive strength (CCS) [Yeh98]; (2) airfoil self-noise (Airfoil); and one OpenIntro dataset: Ames residential home sales (Ames). | 50 |
| Table 2.5: | Test on real data sets with extra noise. | 51 |
| Table 4.1: | $\tau = 0.4$ Coverage Probability for High-dimensional Left-censored Quantile Regression (HLQR) with True F_0 and True f_0 | 111 |
| Table 4.2: | $\tau = 0.7$ Coverage Probability for High-dimensional Left-censored Quantile Regression (HLQR) with True F_0 and True f_0 | 114 |
| Table 4.3: | $\tau = 0.4$ Coverage Probability for High-dimensional Left-censored Quantile Regression (HLQR) with Estimated \hat{F}_n and Estimated \hat{f}_n | 115 |
| Table 4.4: | $\tau = 0.7$ Coverage Probability for High-dimensional Left-censored Quantile Regression (HLQR) with Estimated \hat{F}_n and Estimated \hat{f}_n | 115 |
| Table 4.5: | Gene expressions selected by High-dimensional Left-censored Quantile Regression (HLQR) with 10% censoring in comparison with the ones selected by L_1 norm QR model in [LZ08] (L_1 QR) with no censoring | 120 |

ACKNOWLEDGEMENTS

I would like to first express my special appreciation and thanks to my advisor Professor Jelena Bradic, who has been a tremendous mentor for me. I would like to thank you for encouraging me to explore all kinds of ideas, and for helping me grow as a researcher with curiosity and independent thinking. Your advice on both research as well as on my career have been invaluable.

I would like to thank Professor Ery Arias-Castro, Professor Sanjoy Dasgupta, Professor Yoav Freund, and Professor Rayan Saab for serving as my committee members and offering meaningful discussions. I would like to thank Professor Ruth Williams for her solid and detailed instruction on probability theory. I would like to thank Professor Ian Abramson, Professor Ery Arias-Castro, Professor Jelena Bradic, Professor Dimitris Politis, and Professor Lily Xu for bringing me into the beautiful world of data and statistics. I would also like to thank Professor Sanjoy Dasgupta and Professor Lawrence Saul for their excellent teaching and intuitive explanation on fundamental machine learning concepts. I really enjoyed attending your lectures.

To the Mathematics Department, I appreciate all the resources and support provided. Thanks to Wilson Cheung, Holly Proudfoot, Scott Rollans, and Debbie Shon for all your patience and assistance. I thank my fellow office-mates, Jiaqi Guo, Jingwen Liang and Andrew Ying, for the stimulating discussions, the interesting conversations, and the delicious foods. Also I thank all my friends and fellow graduate students in the department, Yaqing Wang, Shaunak Das, Ching-Wei Ho, Marquis Hou, Fred Li, Yuchao Liu, Ran Pan, Liyu Qin, Dun Qiu, Kuang Thamrongpaioj, and Selene Xu, for all the funs.

My sincere thanks also go to Andrew Martin, David Fagnan and Imri Sofer at Zillow, and Vikas Bhardwaj and Manisha Srivastava at Amazon, who provided me an opportunity to join their teams as intern, and helped me gain industry experience.

I really enjoyed my last five years at San Diego, which is and will always be the second hometown in my heart. I am thankful to all the hot sake, whiskey and beer I consumed, to the

treadmills in Costa Verde and Mesa Apartments, and to my hot tub, which could always relieve my stress.

Finally, I must express my very profound gratitude to my parents, my grandmother, and to my partner for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Chapter 1, in full, is a reprint of the material as it appears in Journal of the American Statistical Association. Li, Hanbo; Bradic, Jelena. Boosting in the presence of outliers: adaptive classification with nonconvex loss functions, JASA, 1-15, 2018. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 2, in full, is a version of the material as it appears in International Conference on Machine Learning. Li, Hanbo; Martin, Andrew. Forest-type regression with general losses and robust forest, ICML, 2017. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 3, in full, is currently being prepared for submission for publication of the material. Li, Hanbo; Bradic, Jelena. Forest-type quantile regression for randomly censored data. The dissertation/thesis author is the primary investigator and author of this material.

Chapter 4, in full, is currently being prepared for submission for publication of the material. Bradic, Jelena; Guo, Jiaqi; Li, Hanbo. High-dimensional covariate effects on left-censored quantile event times. The dissertation/thesis author is one of the principal investigators and author of this material.

VITA

| | |
|-----------|--|
| 2013 | B. S. in Mathematics <i>with Honours, First Class</i> , the Chinese University of Hong Kong, Hong Kong |
| 2013-2018 | Graduate Teaching Assistant, University of California, San Diego |
| 2015 | C. Phil. in Mathematics, University of California, San Diego |
| 2018 | Ph. D. in Mathematics, University of California, San Diego |

PUBLICATIONS

Li, Hanbo, and Andrew Martin, “Forest-type regression with general losses and robust forest”, *International Conference on Machine Learning*, 2017.

Li, Hanbo, and Jelena Bradic, “Boosting in the presence of outliers: adaptive classification with nonconvex loss functions”, *Journal of the American Statistical Association*, 1-15, 2018.

Bradic, Jelena, Jiaqi Guo, and Hanbo Li, “High-dimensional covariate effects on left-censored quantile event times”, *Manuscript in Preparation*, 2018.

Li, Hanbo, and Jelena Bradic, “Forest-type quantile regression for randomly censored data”, *Manuscript in Preparation*, 2018.

ABSTRACT OF THE DISSERTATION

Robust Classification and Regression

by

Hanbo Li

Doctor of Philosophy in Mathematics

University of California San Diego, 2018

Professor Jelena Bradic, Chair

Recent advances in technologies for cheaper and faster data acquisition and storage have led to an explosive growth of data complexity in a variety of scientific areas. As a result, noise accumulation, experimental variation, and data inhomogeneity have become substantial. However, many classical classification and regression methods in such settings are known to pose many statistical challenges and hence call for new methods and theories.

This thesis is devoted to robust classification and regression algorithms with theoretical guarantee on important statistical properties. In Chapter 1, we present a classification framework – ArchBoost, which applies to a wide range of loss functions including nonconvex losses and is specifically designed to be robust and efficient whenever the labels are recorded with an error

or whenever the data are contaminated with outliers. In Chapter 2, we introduce a forest-type framework for regression problems, and prove that many state-of-the-art forest algorithms belong to this framework. We then propose robust forest-type regression methods by applying our proposed framework to robust loss functions. In Chapter 3, we design a novel estimating equation motivated by the framework in Chapter 2 to solve quantile regression problem on random censored data. In Chapter 4, we focus on high-dimensional left-censored quantile regression and study its inference problem. We modify the quantile loss to accommodate the left-censored nature of the problem, by extending the idea of redistribution of mass. For the inference part, asymptotic properties are carefully investigated. All the methods in aforementioned chapters are tested through extensive numerical experiments on both simulated and real data sets.

Chapter 1

Boosting in the Presence of Outliers: Adaptive Classification with Non-convex Loss Functions

1.1 Introduction

Recent advances in technologies for cheaper and faster data acquisition and storage have led to an explosive growth of data complexity in a variety of scientific areas such as high-throughput genomics, biomedical imaging, high-energy physics, astronomy and economics. As a result, noise accumulation, experimental variation and data inhomogeneity have become substantial. However, classification in such settings is known to pose many statistical challenges and hence calls for new methods and theories.

ArchBoost contributes to the literature of binary classification algorithms and boosting algorithms in particular. It applies to a wide range of loss functions including non-convex losses and is specifically designed to be robust and efficient whenever the labels are recorded with an error or whenever the data is contaminated with outliers. ArchBoost tilts or *arches* down the loss

function to adapt to the unknown and unobserved noise in the data by exploring non-convexity efficiently.

To design the new framework we will amend the drawbacks of the AdaBoost algorithm [FS97] in the contaminated data setting. AdaBoost algorithm is based on an iterative scheme, in which at each stage data is reweighed, and a new weak classifier is found by minimizing the exponential loss. A final estimate of the classification boundary is found by summing up weak classifiers throughout all iterations. AdaBoost’s sensitivity to outliers comes from the unbounded weight assignment on the misclassified observations. As outliers are more likely to be misclassified, they are very likely to be assigned large weights and will be repeatedly refitted in the following iterations. This refitting will deteriorate seriously the generalization performance, as the algorithm “learns” incorrect data distribution. To achieve robustness, the algorithm should be able to abandon observations that are on the extreme, incorrect side of the classification boundary. Here, we theoretically and computationally investigate the applicability of non-convex loss functions for this purpose. We illustrate that the best weight updating rule is to assign a weight of $-\phi'(y_i F(x_i))$ to each data point (x_i, y_i) with $F(x_i)$ denoting the current estimate of the classification boundary. This assignment is only efficient if the loss function ϕ is a non-convex loss function. We develop a tilting argument for the non-convex losses. It is shown that, if we use a non-convex loss, sufficiently tilted, i.e. $-\phi'(v)$ is small for all $v \ll 0$, then the outliers are eliminated successively. Hence, a constant “trimming” – typically used in robust statistics – is not sufficient for outlier removal in classification setting. In tilting or “arching” the loss function, we are effectively preserving as much fidelity to the data as possible, while redistributing emphasis to different observations. We propose a new *ArchBoost* framework that implements the above tilting method and adjusts for optimality by a new search of the optimal weak hypothesis. Instead of relying only on gradient descent rules (like LogitBoost [FHT00] or GradientBoost [Fri01]), ArchBoost chooses the optimal weak hypothesis that is most orthogonal to the previous weak hypothesis, therefore improving the most the accuracy of the next iteration.

We propose a sufficient set of conditions needed for a loss function to allow for good properties of the ArchBoost. We show that not every non-convex function satisfies such conditions; an example is the sigmoid loss. However, we propose a family of loss functions, γ -loss, that balances both the benefits of non-convexity and the empirical risk interpretation of boosting. Lastly, the proposed family of ARB- γ algorithms is widely applicable to a wide variety of problems related to non-Gaussian observations and data that are mis-labeled (maliciously or otherwise). We address its robustness and statistical efficiency with details. Although it is straightforward to provide such analysis for parametric linear models, computations for the nonparametric and classification boundaries are far more challenging. We provide novel influence function [Ham74] and finite sample breakdown point theory [Ham68] that fill in the gap in the existing literature on robustness of the boosting algorithms.

1.1.1 Literature Review

There have been considerable efforts focused on designing methods that adapt to the error in the data: outliers and/or mislabeling of the observations. In the existing work, algorithms of [GD04] achieve provable guarantees [NDRT13, KTEM07] when contamination model [SBH13] is known or when multiple noisy copies of the data are available [CSS02], good generalization errors in the test set are by no means guaranteed. This problem is compounded when the contamination model is unknown, where outliers need to be detected automatically. Despite progress on outlier-removing algorithms, significant practical challenges (due to exceedingly restrictive conditions imposed therein) remain. Hence, a classification method that doesn't rely on the specified model of the corruption in the observations is still unavailable.

As boosting algorithms utilize observed data distribution over iterations, they may provide a robust alternative to the existing classification methods. Among the boosting algorithms, the most famous one is AdaBoost [FS97] that averages simple estimators (classifiers) from reweighted data over a sequence of iterations. It is the first adaptive boosting algorithm because the update at

each iteration is a direct function of the classification error of the previous step. AdaBoost then attracted much attention from statistics community, and has proven to be simple and effective [ZY05]. [Bre96, Bre99] shows that AdaBoost is a gradient descent method in function space and [FHT00] views AdaBoost as a gradient-based incremental search for an additive model using the exponential loss function. By observing that the exponential criterion is equivalent to the binomial log-likelihood criterion to the second order, [FHT00] also proposed the LogitBoost algorithm. All these algorithms depend on standard convex optimization techniques like the Newton method. The descent method viewpoint then extends the usage of boosting to the context other than classification. For example, [Fri01] developed gradient boosting method for regression using squared error loss, and [MBBF99a] generalizes the boosting idea to wider families of loss functions.

Nevertheless, in the presence of the label noise and/or outliers, the existing methods face significant challenges [DW00]. AdaBoost is known to be very sensitive to noise [FS96, DW00, MO97] because of the exponential criterion it uses. The weights on repeatedly misclassified data increase exponentially fast which leads AdaBoost to overfit the noises. Algorithms like LogitBoost, MadaBoost [DW00], Log-lossBoost [CSS02] are able to better tolerate noise than AdaBoost because they use loss functions that give much slower weights growth rate than e^x . However, they are still not insensitive to outliers or provably robust. In fact, any boosting algorithm with convex loss is highly susceptible to a random label noise as pointed out by [LS10].

Boost by majority (BBM) [Fre95] follows a very different mechanism and can give up on repeatedly misclassified observations because it has a pre-assigned number of boosting iterations. Hence, the weights updating rule of BBM is non-convex. However, the non-adaptiveness of BBM prevents its practical usage because the uniform bound $1/2 - \gamma$ ($\gamma > 0$) on the errors of weak learners are hard to achieve. BrownBoost [Fre01] combines the non-convexity of BBM and the adaptiveness of AdaBoost, and RobustBoost [Fre09] is developed based on BrownBoost and further adapts to the idea of margin maximization which is believed to be the reason for the good

generalization performance of AdaBoost [FSA99, ROM01, Ser03]. However, BrownBoost hinders upon an extra tuning parameter, target error ϵ , and RobustBoost depends on both target error ϵ and maximum margin θ . These tuning parameters make both algorithms highly inconsistent with respect to minor changes in the population parameter settings. Furthermore, both BrownBoost and RobustBoost do not fit in the mainstream boosting algorithms that analytically minimize a convex loss function. They solve two differential equations for two unknowns at each iteration, and the loss function (which they call potential function) changes after every iteration and converges to the 0-1 loss. Although stable in simulations, the statistical properties and robustness are unknown. Therefore, a natural question is: how do we formally develop an adaptive, mainstream and robust boosting algorithm that has a non-convex loss function and has provable robustness properties? In this paper, we address this question and propose a fully automatic estimator, *ArchBoost*, with no tuning parameters, that has provable robustness guarantees. Since ArchBoost does not require the knowledge of the erroneous labels, or the knowledge of the errors themselves, one can probe the utilities of the algorithm in the extremely wide scope of heterogeneous problems.

ArchBoost keeps the initial motivation of the boost by majority method in that the algorithms gives up on repeatedly misclassified observations. However, unlike BBM or RobutBoost it does so without requiring any pre-tuning of the error or maximum margin. ArchBoost adaptively learns which data to give up on without a-priori intervention. Additionally, ArchBoost keeps the reweighing flavor of the AdaBoost or GradientBoost algorithms but it differs in the way it minimizes the empirical risk function as it allows for non-convex losses. While GradientBoost uses least-squares and Newton criterions for finding the optimal classifier, ArchBoost utilizes the hardness condition to define an estimating equations and solves the equations directly (not approximately). Because of that, ArchBoost doesn't reduce to the existing methods when the loss function of choice is a recognized convex loss; for example, ArchBoost does not reduce to the L_2 Boost when the loss is the least squares loss.

1.2 Methodology of the ArchBoost

Let \mathcal{X} denote a p -dimensional domain, \mathcal{Y} denote the class label set $\{-1, 1\}$, $\{(X_i, Y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ be i.i.d. data points ($p \leq n$), ϕ be a differentiable loss function, and \mathcal{F} be a class of functions from \mathbb{R}^p to \mathbb{R} . For any distribution \mathbb{P} on $\mathcal{X} \times \mathcal{Y}$, we wish to find $F \in \mathcal{F}$ that minimizes $\mathbb{E}_{\mathbb{P}}[\mathbb{1}\{YF(X) < 0\}]$. With the classification-calibration condition [BJM06] on ϕ , this problem is equivalent to finding $F^* \in \mathcal{F}$ that minimizes the ϕ -risk $R_{\phi}(F) = \mathbb{E}_{\mathbb{P}}[\phi(YF(X))]$. We summarize $F^*(x) = \arg \min_{F \in \mathcal{F}} \Phi(F(x))$, $x \in \mathcal{X}$ where $\Phi(F(x)) := \mathbb{E}[\phi(YF(X))|X = x]$ in Table 1.1.

Table 1.1: The list of commonly used loss functions and its corresponding F^*

| Classification Method | Population parameters | |
|------------------------|-------------------------|--|
| | Loss function $\phi(v)$ | Optimal Minimizer $F^*(x)$ |
| Logistic | $\log(1 + e^{-v})$ | $(\log \mathbb{P}(y = 1 x) - \log \mathbb{P}(y = -1 x))$ |
| Exponential | e^{-v} | $\frac{1}{2} (\log \mathbb{P}(y = 1 x) - \log \mathbb{P}(y = -1 x))$ |
| Least Squares | $(v - 1)^2$ | $\mathbb{P}(y = 1 x) - \mathbb{P}(y = -1 x)$ |
| Modified Least Squares | $[(1 - v)_+]^2$ | $\mathbb{P}(y = 1 x) - \mathbb{P}(y = -1 x)$ |

AdaBoost [FS97] minimizes the empirical ϕ -risk $\hat{R}_{\phi,n}(F) = \frac{1}{n} \sum_{i=1}^n \phi(Y_i F(X_i))$ with the exponential loss, $\phi(v) = e^{-v}$, in a stagewise manner. It approximates the unknown Bayes classifier with a combination of weak classifiers, h_t , obtained by employing a weak learner at each iteration t . It is critical to observe that minimization of the exponential loss by itself is not sufficient to guarantee low generalization error of the AdaBoost [Sch13]. Its excellent performance is based on the premise that at each iteration of the algorithm, the method is forced to infer something new about the observations. This amounts to reweighing the observations by a weight vector w , so that the misclassified points gain more weight in the next iteration. However, in the presence of outliers, such methodology will iteratively attempt to refit the outliers to one of the classes and hence effectively pull the decision boundary away from the ground truth. Unfortunately, all convex loss functions will inevitably keep upweighting the persistently misclassified points, and as pointed out by [LS10], they all lead to non-robust boosting methods. Therefore, new boosting principles need to be designed that allow the loss to be *non-convex*. *ArchBoost* method,

that we propose below, is such a framework which, equipped with non-convex losses, leads to adaptive and robust algorithms that have provable guarantees. By exploring the non-convexity, ArchBoost is gradually dropping out the persistent observations from the refitting procedure at each new iteration of the algorithm. In this way, if the observations are consistently being misfit, they are suspected of being outliers and are steadily assigned less importance in the risk minimization procedure. Thus, ArchBoost tilts (i.e. arches) the weight distribution to the non-outlying observations. As an example of a weight updating rule that is effective at arching, we consider the loss function and the weight function, respectively, as

$$\phi(v) = 4/(1 + e^v)^2, \quad w(v) = e^v/(1 + e^v)^3 \tag{1.1}$$

with $v = yF(x)$. To further illustrate this idea, we present graphically (1.1) in Figure 1.1, together with the losses and weight distributions of AdaBoost and LogitBoost.

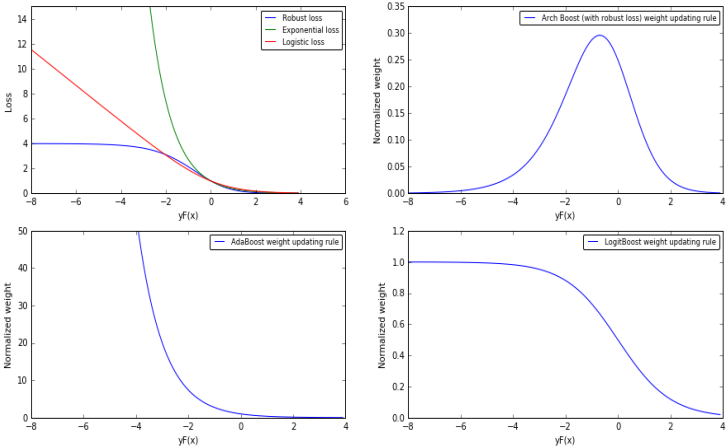


Figure 1.1: AdaBoost, LogitBoost, ArchBoost loss functions and the corresponding normalized weight updating rules.

The novel boosting framework *ArchBoost* is presented in Algorithm 1. It iteratively builds an additive model $F_T(x) = \sum_{t=1}^T \alpha_t h_t(x)$ where h_t belongs to some space of weak classifiers denoted by \mathcal{H} (e.g. decision trees). Different from Gradient boost and AdaBoost, ArchBoost finds the optimal weak learner h_t , the step size α_t , and the weight updating vector w_t by exploring

Algorithm 1 ArchBoost (ϕ)

Given training sample: $(x_1, y_1), \dots, (x_n, y_n)$ initialize the weights $w_0(x_i, y_i) = 1/n$

for $t = 1, \dots, T$ **do**

- 3: (a) Normalize the weight by assigning $w_t = w_t / \sum_i w_t(x_i, y_i)$
(b) Fit the classifier to obtain a class probability estimate $\mathbb{P}_{w_t}(Y = 1|x) \in [0, 1]$ using current weights w_t on the training data.
- 6: (c) Set $h_t(x)$ to be the solution of estimating equation (1.6).
(d) Find α_t by solving the empirical counterpart of (1.7).
(e) Set $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$.
- 9: (f) Update the weights $w_t = -\phi'(yF_t(x))$.

end for

Output the classifier: $\text{sign}(F_T(x))$.

the *Hardness Condition* defined as

$$\mathbb{E}_{w_{t+1}}[Y h_t(X) | X = x] = 0, \quad (1.2)$$

where $\mathbb{E}_w[g(X, Y) | X = x] := \mathbb{E}[w(X, Y)g(X, Y) | X = x] / \mathbb{E}[w(X, Y) | X = x]$. This condition means that, from iteration t to $t + 1$, the weights on \mathcal{X} are updated from w_t to w_{t+1} such that $h_t(X)$ is orthogonal to Y with respect to the inner product defined on the reweighed data. Thus, the weak hypothesis h_t behaves like a random guess on the reweighed data, and hence, the h_{t+1} will be a good supplement to h_t .

Provided that \mathcal{F} includes all measurable functions, we observe that $F^*(x)$ can be defined by the first order optimality condition $\mathbb{E}[Y \phi'(Y F^*(X)) | X = x] = 0$, where ϕ' is defined as the first order derivative $\frac{d}{dv} \phi(v)$. In classification problems, the parameter v of loss function ϕ is $v = Y F(X)$ – that is, the margin of a classifier F applied to a data point (X, Y) . Rewriting the expectation in terms of the class probabilities, we obtain the following representation of the first order optimality conditions

$$\frac{\phi'(-F^*(x))}{\phi'(F^*(x))} = \frac{\mathbb{P}(Y = 1 | X = x)}{\mathbb{P}(Y = -1 | X = x)}. \quad (1.3)$$

We aim to mimic equation above in each of the iteration steps of the proposed framework. In more details, at iteration t , with the current estimate $F_{t-1}(x) = h_1(x) + \dots + h_{t-1}(x)$ at hand, we wish to find a new weak hypothesis $h_t \in \mathcal{H}$, such that $F_t(x) = F_{t-1}(x) + h_t(x)$ with $h_t(x)$ solving the following equation

$$\frac{\phi'(-F_{t-1}(x) - h_t(x))}{\phi'(F_{t-1}(x) + h_t(x))} = \frac{\mathbb{P}(Y = 1|x)}{\mathbb{P}(Y = -1|x)}. \quad (1.4)$$

Next, we aim to explore (1.4) and build an estimating equation to find the optimal h_t . The method of estimating equations is a way of specifying how the optimal h_t should be estimated. This can be thought of as a generalization of many classical methods including the framework of M-estimation. Estimating equation (1.4) involves an unknown quantity $\mathbb{P}(Y = 1|x)$. One may substitute \mathbb{P} with \mathbb{P}_{w_t} , but this coarse estimation could be very biased, especially when the data has outliers. Therefore, we propose to estimate the right hand side of (1.4) by introducing a *bias correction function* $\mathbb{C}_{t-1}(x)$ that depends on both the current estimate F_{t-1} and x , and is such that

$$\frac{\mathbb{P}(Y = 1|x)}{\mathbb{P}(Y = -1|x)} = \mathbb{C}_{t-1}(x) \frac{\mathbb{P}_{w_t}(Y = 1|x)}{\mathbb{P}_{w_t}(Y = -1|x)}. \quad (1.5)$$

Here the conditional probability $\mathbb{P}_{w_t}(Y = 1|x) := \mathbb{E}_{w_t}[\mathbb{1}_{[Y=1]}|X = x]$. Now, we observe that $\mathbb{P}(Y = 1|x)$ and $\mathbb{P}_{w_t}(Y = 1|x)$ satisfies

$$\frac{\phi'(F_{t-1}(x))}{\phi'(-F_{t-1}(x))} \frac{\mathbb{P}(Y = 1|x)}{\mathbb{P}(Y = -1|x)} = \frac{\mathbb{E}[\mathbb{1}_{[Y=1]} \phi'(YF_{t-1}(X))|x]}{\mathbb{E}[\mathbb{1}_{[Y=-1]} \phi'(YF_{t-1}(X))|x]} = \frac{\mathbb{P}_{w_t}(Y = 1|x)}{\mathbb{P}_{w_t}(Y = -1|x)}.$$

Hence, with the bias correction function defined as $\mathbb{C}_{t-1}(x) = \phi'(-F_{t-1}(x))/\phi'(F_{t-1}(x))$, equations (1.5) and (1.4) lead to

$$\frac{\phi'(-F_{t-1}(x) - h_t(x))}{\phi'(F_{t-1}(x) + h_t(x))} = \frac{\phi'(-F_{t-1}(x))}{\phi'(F_{t-1}(x))} \frac{\mathbb{P}_{w_t}(Y = 1|x)}{\mathbb{P}_{w_t}(Y = -1|x)}. \quad (1.6)$$

Therefore, the estimating equation principle of ArchBoost selects the optimal h_t as a solution to the estimating equation (1.6). For the loss function (1.1), for example $\mathbb{C}_{t-1}(x) = e^{F_{t-1}(x)}$. Additionally, note that \mathbb{P}_w can always be estimated as long as we use a weak learner that is capable to give class probabilities. One example is decision tree in which case in each terminal region R_j , one can estimate $\mathbb{P}_{w_t}(Y = 1|x)$ by $\sum_{x_i \in R_j, y_i=1} w(x_i, y_i) / \sum_{x_i \in R_j} w(x_i, y_i)$.

Observe that we can explicitly solve equation (1.6) for many commonly used loss functions. For the robust loss (1.1) in Figure 1.1, (1.6) becomes

$$e^{F_{t-1}(x)+h_t(x)} = e^{F_{t-1}(x)} \frac{\mathbb{P}_{w_t}(Y = 1|x)}{\mathbb{P}_{w_t}(Y = -1|x)},$$

leading to $h_t = \log \mathbb{P}_w(Y = 1|x) - \log \mathbb{P}_w(Y = -1|x)$. The results for existing losses are summarized in Table 1.2. Observe that for different choices of the weight vector w_t , the resulting h_t changes. The Hardness Condition works as the guideline of updating the weights w_t . To ensure that $\alpha_t h_t$ indeed decreases the ϕ -risk we consider an additional line search step

$$\alpha_t = \arg \min_{\alpha \in \mathbb{R}} \mathbb{E} \left[\phi \left(Y F_{t-1}(X) + Y \alpha h_t(X) \right) \right]. \quad (1.7)$$

We observe that for

$$w_{t+1}(X, Y) := -\phi'(Y F_t(X)), \quad (1.8)$$

the α_t , (1.7), satisfies $\mathbb{E}_{w_{t+1}}[Y \alpha_t h_t(X)] \propto \mathbb{E} \left[-\phi'(Y F_{t-1}(X) + Y \alpha_t h_t(X)) \cdot Y \alpha_t h_t(X) \right] = 0$. For the robust loss (1.1), $w(v) = e^v / (1 + e^v)^3$ is proportional to $-\phi'(v) = 8e^v(1 + e^v)^{-3}$ up to a constant. Therefore, by updating weights according to (1.8), the Hardness Condition (1.2) is satisfied.

Lastly, we emphasize that throughout the above derivation, we did not put any convexity restriction on the loss function. The only assumption we made is that $\Phi(F(x))$ has only one critical point that is the global minimum, a condition satisfied by many *non-convex* functions, e.g.

Table 1.2: The list of commonly used loss functions and their weak hypotheses h

| Classification Method | Population parameters | |
|------------------------|----------------------------|--|
| | Loss function $\phi(v)$ | Optimal weak hypotheses $h(x)$ |
| Logistic | $\log(1 + e^{-v})$ | $\log \mathbb{P}_w(Y = 1 x) - \log \mathbb{P}_w(Y = -1 x)$ |
| Exponential | e^{-v} | $\frac{1}{2} (\log \mathbb{P}_w(Y = 1 x) - \log \mathbb{P}_w(Y = -1 x))$ |
| Least Squares | $(v - 1)^2$ | $C(1 - F(x))(1 + F(x))/(CF(x) + 1)$ |
| Modified Least Squares | $[(1 - v)_+]^2$ | $C(1 - F(x))(1 + F(x))/(CF(x) + 1)$ |

* $C = \mathbb{P}_w(Y = 1|x) - \mathbb{P}_w(Y = -1|x)$

invex functions of [BIM86]. In this way, the ArchBoost algorithm can be applied to a broad family of non-convex loss functions (see Section 1.3). Moreover, note that the weak hypotheses of the least squares loss and modified least squares loss (Table 1.2) depend on the current estimate $F(x)$ and the weighted conditional probability $\mathbb{P}_{w_t}(Y = 1|x)$, which is different from that of Gradient boosting [Fri01]. Observe that the Gradient boosting effectively fits a least-squares method on pseudo-responses (see Step 4 of Gradient boost that approximates equation (9) therein), and hence the optimal weak learner is not chosen robustly. ArchBoost is an improvement as it designs a fully robust algorithm. Moreover, Gradient boost doesn't define the weights w and hence has a very different viewpoint. Although it can be applied to non-convex losses using the simple steepest descent, the solution is unstable and the corresponding algorithms using our non-convex losses (Section 1.3) behave even worse than LogitBoost.

1.3 Robust non-convex loss functions

Not every non-convex function is a valid candidate for the developed *ArchBoost* method. Any binary classification problem can be written as

$$\min_{v \in \mathbb{R}} \left[\mathbb{P}(Y = 1|x)\phi(v) + \mathbb{P}(Y = -1|x)\phi(-v) \right], \quad (1.9)$$

where $v := YF(x)$ is the margin. We assume that (4.31) has a unique optimal solution in \mathbb{R} for every $x \in \mathcal{X}$. Note that this condition is not equivalent to the convexity of ϕ but rather to the local convexity around the true parameter of interest.

Definition 1. A function ϕ is an **ArchBoosting loss function** if it is differentiable and (i) $\phi(v) \geq 0$ for all $v \in \mathbb{R}$ and $\inf_{v \in \mathbb{R}} \phi(v) = 0$; (ii) for any $0 < \alpha < 1$, $\alpha\phi(v) + (1 - \alpha)\phi(-v)$ has only one critical point v^* which is the global minimum; (iii) for any $0 \leq \alpha \leq 1$ and $\alpha \neq \frac{1}{2}$, $\inf\{\alpha\phi(v) + (1 - \alpha)\phi(-v) : v(2\alpha - 1) \leq 0\} > \inf\{\alpha\phi(v) + (1 - \alpha)\phi(-v) : v \in \mathbb{R}\}$.

Conditions (i) and (iii) together imply that ϕ is an upper bound of the 0-1 loss up to a constant scaling. Condition (iii) is called "classification calibration" [BJM06] and is satisfied as long as ϕ is convex, differentiable and $\phi'(0) < 0$. It is considered the weakest possible condition for the resulting classifier to be Bayes-consistent. However, when considering non-convex losses, the set of regularity conditions doesn't exist in the current literature.

Lemma 1. *All continuously differentiable convex functions $\phi : \mathbb{R} \rightarrow \mathbb{R}_+$ such that ϕ' is not a constant satisfy Condition (ii). Moreover, all positive, continuously differentiable functions ϕ such that $\phi'(v) \neq 0$ for all $v \in \mathbb{R}$, satisfy Condition (ii) as long as the function $g : (0, \infty) \rightarrow (0, 1)$, defined as $g(v) := \phi'(-v)/\phi'(v)$ is strictly increasing and surjective.*

By Lemma 1, the logistic, exponential, least square and modified least square losses are all valid ArchBoosting losses. Differentiability of the loss is a non-crucial, technical condition and the hinge loss can be shown to satisfy Conditions (i)-(iii). However, the sigmoid loss $\phi_{sig}(v) = (1 + e^v)^{-1}$ does not satisfy Condition (ii).

Observe that the right hand side of (1.4) does not depend on the loss function ϕ and can take values in the positive real line \mathbb{R}_+ . Hence, we can parameterize it with any strictly increasing surjective function $g : \mathbb{R} \rightarrow \mathbb{R}_+$ i.e. $\phi'(-v)/\phi'(v) = g(v)$. The classical motivation for reparametrization [NB04] – often called link functions – is that one uses a parametric representation that has a natural scale matching the desired one. One such function satisfying

second part of Lemma 1 is $g(v) = e^{(\gamma-1)v}$ with constant $\gamma > 1$. This parametrization is not unique but it admits a solution to the differential equation $\phi'(-v)/\phi'(v) = e^{(\gamma-1)v}$. The solution (see Supplement) is a family of non-convex losses, which we name γ -robust losses,

$$\phi_\gamma(v) = 2^\gamma(1 + e^v)^{-\gamma}, \quad \gamma > 1. \tag{1.10}$$

We plot the γ -robust losses and the corresponding normalized weight updating functions in Figure 1.2.

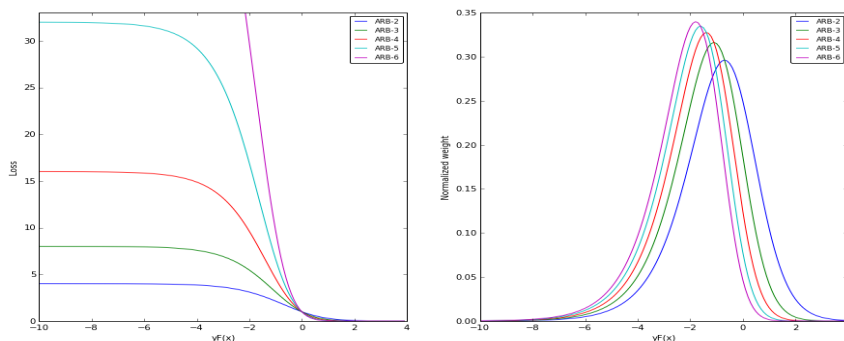


Figure 1.2: γ -robust losses, ϕ_γ and the corresponding normalized weight updating rules.

Parameter γ is not a tuning parameter, but rather an index of a family of non-convex losses much like Huber and Tukey’s biweight losses. All ϕ_γ are bounded functions ($\leq 2^\gamma$) and hence the effects of the outliers are necessarily bounded. Moreover, the weight updating rules down-weights the largely misclassified data points. When $\gamma = 1$, the weight updating curve is equivalent to the sigmoid loss $\phi(v) = 1 - \tanh(\lambda v)$ when $\lambda = 1/2$ [MBBF99a]. Moreover, for $\gamma = 2$, the loss ϕ_2 is similar to the Savage loss $\phi(v) = (1 + e^{2v})^{-2}$ of [MSV09], in which they used the probability elicitation technique. The following Lemma 2 allows us to use ϕ_γ together with the ArchBoost method. The resulting family of robust boosting algorithms, named *Adaptive Robust Boost- γ* (*ARB- γ*), are presented in Algorithm 2.

Lemma 2. For all $\gamma > 1$, ϕ_γ is an ArchBoosting loss function.

Algorithm 2 Adaptive Robust Boost (ARB)- γ

Given: $(x_1, y_1), \dots, (x_n, y_n)$, initialize the weight vector w_0 , e.g. $w_0(x_i, y_i) = 1/n$
for $t = 1, \dots, T$ **do**
3: (a) Normalize the weight vector $w_t = w_t / \sum_i w_t(x_i, y_i)$
(b) Compute the weak classifier to obtain a class probability estimate $\mathbb{P}_{w_t}(Y = 1|x) \in [0, 1]$, using weights w_t on the training data.
(c) Set $h_t(x) = \log \frac{\mathbb{P}_{w_t}(Y=1|x)}{\mathbb{P}_{w_t}(Y=-1|x)} \in \bar{\mathbb{R}}$.
6: (d) Find α_t by solving empirical counterpart of (1.7).
(e) Set $F_t(x) = F_{t-1}(x) + \alpha_t h_t(x)$
(f) Set $w_{t+1} = e^{yF_t(x)}(1 + e^{yF_t(x)})^{-\gamma-1}$
9: **end for**
Output the classifier: $\text{sign}(F_T(x))$

1.4 Theoretical Considerations

Despite the substantial body of existing work on boosting classifiers (e.g. [BJM06, Bre04, Fre95, FHT00, KP02, ZY05]), research on robust boosting has been limited to methodological proposals with little supporting theory (e.g., [Lit91, KL93, Gen03, NL02, KS03, Ros05, LKB08, BK13, MCX⁺16, MG16]).

1.4.1 Numerical convergence

In this section, we discuss the numerical convergence of the ArchBoost algorithm whenever the loss ϕ belongs to the class of ArchBoosting loss functions. The main difference from the existing work (e.g. [KS03] and [ZY05]) is that they used the gradient descent rule in the first paper or an approximate minimization in the second one, while we only use the hardness condition to select the weak hypothesis h . Here, \mathcal{F}^T is a set of T -combinations of functions in \mathcal{H} , more precisely, $\mathcal{F}^T = \{F : F = \sum_{t=1}^T \alpha_t h_t, \alpha_t \in \mathbb{R}, h_t \in \mathcal{H}\}$. Then every $f \in \cup_{T=1}^{\infty} \mathcal{F}^T$ can be represented as $\sum_{h \in \mathcal{H}_f} \alpha^h h$ for an appropriate subset $\mathcal{H}_f \subset \mathcal{H}$, and its l_1 -norm is defined as $\sum_{h \in \mathcal{H}_f} |\alpha^{(h)}|$, and its l_2 -norm as $\sum_{h \in \mathcal{H}_f} \sqrt{|\alpha^{(h)}|^2}$. Lastly, let $\{\bar{f}_t\}$ be a sequence of reference functions with empirical risk converging to $R_{\phi, n}^* = \inf_{F \in \cup_{T=1}^{\infty} \mathcal{F}^T} \hat{R}_{\phi, n}(F)$.

Condition 1. (i) ϕ is Lipschitz differentiable; (ii) $\hat{\mu}(h_t, w_t) = (1/n) \sum_{i=1}^n Y_i h_t(X_i) w_t(X_i, Y_i) \rightarrow 0$ as $t \rightarrow \infty$; (iii) the step sizes α_t satisfy

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \quad \sum_{t=1}^{\infty} \alpha_t^2 < \infty, \quad \sum_{t=1}^{\infty} \frac{\alpha_{t+1} \xi_t \log t}{t^{c_t}} < \infty,$$

for some $\xi_t = o(1)$, $\xi_t \geq 0$; (iv) \bar{f}_t satisfies $\|\bar{f}_t - F_t\|_1 = o(\log t)$, $\|\bar{f}_t - F_t\|_2^2 \leq \frac{\|\bar{f}_t - F_t\|_1^2}{t^{c_t}}$ where $c_t \rightarrow 0$ and $t^{c_t} \rightarrow 1$ as $t \rightarrow \infty$.

Theorem 1. Let ϕ be an ArchBoosting loss function and assume the weak learner is able to divide the domain X into disjoint regions and give the class probability estimations (e.g. decision tree). Let F_T be the ArchBoost classifier, then $\hat{R}_{\phi, n}(F_T)$ will converge in \mathbb{R} as $T \rightarrow \infty$. In addition, under Condition 1, $\hat{R}_{\phi, n}(F_T) \rightarrow R_{\phi, n}^*$ as $T \rightarrow \infty$.

Unlike existing results, Theorem 1 does not require any additional algorithmic tuning parameters (see Theorem 3.1 of [ZY05] and choices of ε_t, Λ_t). It is worth mentioning again that the proof techniques in the existing literature do not extend to non-convex losses. We bridge the gap by developing new analysis. Results in [BT07] (e.g., Theorem 6) hold under an assumption of a positive lower bound on the Hessian of the empirical risk, which is strictly violated by any non-convex loss. Furthermore, Theorem 1 allows the approximate minimization step (1.7) to be inexact (by contrast, see Theorem 6 of [BT07]).

Remark 1. The reference sequence $\{\bar{f}_t\}$ needs to be in a local neighborhood of F_t . For all \bar{f}_t such that $\|\bar{f}_t\|_1 = o(\log(t))$, the condition further reduces to $\|\bar{f}_t - F_t\|_1 \leq \|\bar{f}_t - 0\|_1$, that is, the distance between \bar{f}_t and F_t is smaller than the distance between \bar{f}_t and a random guess. This can be achieved by shrinking the step sizes α_t at a constant rate over every iteration. Moreover, the effects of the second constraint regarding \bar{f}_t can be explained as a non-sparsity assumption on the difference between F_t and \bar{f}_t , and is asymptotically negligible because $t^{c_t} \rightarrow 1$ when $t \rightarrow \infty$, which leads to the trivial inequality between l_1 and l_2 norms.

Remark 2. The classical conditions that are guarding against infinitely small step sizes are now supplemented with an additional constraint $\sum_{t=1}^{\infty} t^{-c_t} \alpha_{t+1} \xi_t \log t < \infty$. For example, if $\xi_t = O(t^{-1})$, then we can choose $\alpha_t = O(t^{-b-c_t})$ where b is any positive constant and c_t can converge to 0 at any speed. However, if $\xi_t = O((\log t)^{-1})$, we need $c_t \rightarrow 0$ slowly (e.g. $O((\log \log t)^{-1})$) and α_t can be chosen as $O(t^{-1})$. The additional constraint on the step size choice acts as a penalty on allowing non-convex loss functions [ZY05].

1.4.2 Robustness

In this section, we quantify and justify the robustness of ArchBoost Algorithm 1 through the point of view of the influence function, as well as that of the finite sample breakdown point.

Influence function

The richest quantitative robustness measure is provided by the influence function [Ham74] $u \rightarrow IF(u; T, G)$ of T at G . It is defined as the first Gâteaux derivative of a functional T at a distribution \mathbb{P} , i.e., $IF(z; T, \mathbb{P}) = \lim_{\varepsilon \rightarrow 0^+} [T((1 - \varepsilon)\mathbb{P} + \varepsilon\Delta_z) - T(\mathbb{P})]/\varepsilon$, where Δ_z is the Dirac distribution at the point z such that $\Delta_z(\{z\}) = 1$. It gives the effect that an outlying observation may have on an estimator. To simplify the analysis, we consider a subclass of binary classification models, in which the true boundary F^* is assumed to belong to a class of functions H . Here, H is defined as a Reproducing Kernel Hilbert Space (RKHS) with a bounded kernel k and the induced norm $\|\cdot\|_H$. Observe that ArchBoost is consistent only if it is properly regularized (stopped after a certain number of steps; see Theorem 5). Hence, to study its robustness properties we consider a regularized criterion

$$f_{\mathbb{P}, \lambda} = \arg \min_{f \in H} \{ \mathbb{E}_{\mathbb{P}} [\phi(Y, f(X))] + \lambda \|f\|_H^2 \}.$$

The loss ϕ is a function of tuple $(Y, f(X))$ only for convenience of analysis. The feature map is $\Psi : \mathcal{X} \rightarrow H$ with $\Psi(x) = k(x, \cdot)$.

Theorem 2. *The influence function of $f_{\mathbb{P}, \lambda}$ takes the form $IF(z; T, \mathbb{P}) = -S^{-1} \circ J$, where \circ is defined to mean S^{-1} acting on J and operators $S : H \rightarrow H$ and $J \in H$ are defined as $S = \mathbb{E}_{\mathbb{P}} \left[\phi''(Y, f_{\mathbb{P}, \lambda}(X)) \langle \Psi(X), \cdot \rangle \Psi(X) \right] + 2\lambda id_H$, $J = \phi'(z_y, f_{\mathbb{P}, \lambda}(z_x)) \Psi(z_x) - \mathbb{E}_{\mathbb{P}}[\phi'(Y, f_{\mathbb{P}, \lambda}(X)) \Psi(X)]$, where $id_H : H \rightarrow H$ is the identity mapping and $z = (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$ is the contamination point. In the above display, the derivative is defined as $\phi'(u, v) := \frac{\partial}{\partial v} \phi(u, v)$.*

For a non-convex loss function ϕ , ϕ'' is not guaranteed to be nonnegative. However, we show that it is sufficient to have the non-negativity of the expectation (locally around F^*) rather than of the second derivative itself.

Lemma 3. *For a binary classification problem, given any distribution \mathbb{P} , whenever ϕ is a twice continuous differentiable ArchBoosting loss function, then $\mathbb{E}_{\mathbb{P}} \left[\phi''(Y, F^*(X)) q^2(X) \right] \geq 0$ for any measurable function $q : \mathcal{X} \rightarrow \mathbb{R}$. Furthermore, if \mathbb{P} and \mathcal{X} are such that $\mathbb{P}(Y = 1 | X = x) \in [\delta, 1 - \delta]$ for some $0 < \delta < \frac{1}{2}$, and if $p\phi''(1, v_p^*) + (1 - p)\phi''(-1, v_p^*) > 0$ at the global minimum v_p^* for all $p \in [\delta, 1 - \delta]$, then there exists $r > 0$ such that $\mathbb{E}_{\mathbb{P}} \left[\phi''(Y, G(X)) q^2(X) \right] \geq 0$ for all measurable function G with $\|G - F^*\|_{\infty} < r$.*

Conditions of the above lemma are satisfied for all γ -robust loss function. With $\gamma = 2$ and any x , $\mathbb{E}_Y[\phi''(Y, F^*(X)) q^2(X) | X = x] = 2p_x^2(1 - p_x)^2 q^2(x) \geq 0$ where $p_x = \mathbb{P}(Y = 1 | X = x)$. Thus, $\mathbb{E}_{\mathbb{P}} \phi''(Y, F^*(X)) q^2(X) \geq 0$. Furthermore, if $p_x \in [\delta, 1 - \delta]$ for some $\delta \in (0, \frac{1}{2})$, then $p_x \phi''(1, F^*(x)) + (1 - p_x) \phi''(-1, F^*(x)) = 2p_x^2(1 - p_x)^2 \geq 2\delta^2(1 - \delta)^2 > 0$ for all $p_x \in [\delta, 1 - \delta]$ ¹.

Theorem 3. *For a binary classification problem, let $\phi : \mathbb{R} \rightarrow [0, \infty)$ be a twice continuously differentiable ArchBoosting loss function and let H be a RKHS with bounded kernel k . Assume \mathbb{R} is a distribution on $\mathcal{X} \times \mathcal{Y}$ such that for all $x \in \mathcal{X}$, $\mathbb{R}(Y = 1 | X = x) \in [\delta, 1 - \delta]$ for some $0 < \delta < \frac{1}{2}$,*

¹Observe that the condition of $p_x \in [\delta, 1 - \delta]$ for some $\delta \in (0, \frac{1}{2})$ restricts our setting to the “low-noise” setting where the true probability of the class membership is bounded away from 0 or 1.

and $p\phi''(1, v_p^*) + (1-p)\phi''(-1, v_p^*) > 0$ at the global minimum v_p^* for all $p \in [\delta, 1-\delta]$. Then there exists $r > 0$ such that for all $\|f_{\mathbb{R}, \lambda} - F^*\|_\infty < r$,

$$\|IF(z; f_{\mathbb{R}, \lambda}, \mathbb{R})\|_H \leq \sqrt{\frac{C_\phi}{\lambda}} + \frac{M_k |\phi'(z_y, f_{\mathbb{R}, \lambda}(z_x))|}{2\lambda}, \quad (1.11)$$

where M_k is the upper bound of the kernel k and $C_\phi = \phi(0, 0)$.

Theorem 3 shows that the robustness mainly comes from the diminishing property of $|\phi'|$. In fact, for any non-convex ArchBoosting loss function, due to Assumption 2, we have $|\phi'(z_y, f_{\mathbb{R}, \lambda}(z_x))| \rightarrow 0$ when $|z_y f(z_x)| \rightarrow \infty$. If we plot $\|IF(z; f_{\mathbb{R}, \lambda}, \mathbb{R})\|_H$ versus $z_y f_{\mathbb{R}, \lambda}(z_x)$, then it will decrease towards a constant far from the origin, much alike the redescending M-estimators. Moreover, Theorem 3 implies that $\|IF(z; f_{\mathbb{R}, \lambda}, \mathbb{R})\|_H$ is unbounded for the exponential loss (AdaBoost), bounded but not diminishing for the logistic loss (LogitBoost) and diminishing for the γ -robust losses (ArchBoost).

Breakdown point

Empirical robustness property defined as breakdown point in [DH83] has proved most successful in the context of location, scale and regression problems (e.g. [Rou84, SR92, Tyl94], etc.). This success has sparked many attempts to extend the concept to other situations (e.g. [RW01, Gen03, DG05], etc.). However, very little work has been done in the classification context. The breakdown point, as defined in [Ham68], is roughly the smallest amount of contamination that may cause an estimator to take on arbitrarily large aberrant values. The breakdown points of $1/n$ for the mean and $1/2$ for the median do reflect their finite-sample behavior. However, an alternative view is desired in the classification context as the magnitude of an estimator may not relate to necessarily bad classification – that is, the size of the weak hypothesis is not crucially related to the classification boundary. Instead, in the context of boosting, we look for the estimator that keeps the gradient of the risk minimization in the *oracle direction*. The meaning of oracle

direction will be further explained in Remark 4. To that end, let $\mathcal{S}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ be a set of observed, contaminated samples among which $O_{m:n} = \{(X_{m+1}, Y_{m+1}), \dots, (X_n, Y_n)\}$ being a set of outliers. Let h_t be the weak hypothesis and denote the vectors $\mathbf{h}_t = (h_t(X_1), \dots, h_t(X_n))$. Let $-\mathbf{g}_t = (-g_t(X_1), \dots, -g_t(X_n))$ stands for the negative gradient of the empirical risk $\hat{R}_{\phi, n}$ on \mathcal{S}_n , whereas $-\mathbf{g}_o = (-g_t(X_1), \dots, -g_t(X_m), 0, \dots, 0)$ is the embedding of the negative gradient of the empirical risk on the sample without outliers $\mathcal{S}_n \setminus O_{m:n}$ into \mathbb{R}^n .

Theorem 4. *For every region R^j , define $\eta_j := |p_j - \frac{1}{2}| / \min(p_j, 1 - p_j)$, where $p_j \in (0, 1)$ and $p_j \neq \frac{1}{2}$. Then at iteration t , if any ArchBoost algorithm, conditional on the realizations $\{(X_i, Y_i) = (x_i, y_i)\}_{i=1}^n$, satisfies that for all R_j ,*

$$\sum_{i: x_i \in O_{m:n} \cap R^j} w_t(x_i, y_i) \leq \eta_j \sum_{i: x_i \in R^j \setminus O_{m:n}} w_t(x_i, y_i), \quad (1.12)$$

then the gradient descent direction is preserved, that is, $-\langle \mathbf{g}_o, \mathbf{h}_t \rangle \geq 0$.

Conditions of the above theorem are very mild. Theorem 4 suggests that any ArchBoost algorithm that satisfies the above conditions preserves the descending direction of the non-contaminated empirical ϕ -risk, hence it minimizes the oracle risk while disregarding the outliers.

Remark 3. When $p_j = \frac{1}{2}$ – that is, the total weight on positive labels is the same as that of the negative ones in region R^j – the elements of \mathbf{h} corresponding to the points in R^j are 0 and consequently have no influence on the sign of $-\langle \mathbf{g}_o, \mathbf{h} \rangle$. Moreover, the case of $p_j = 0$ or 1 is not of the main interest as in this case, then informally we have 100% confidence about which label the data in R^j should contain. In this case, all the data in that region have the same labels and hence it is reasonable to say there are no outliers. If we allow η_j to be infinity, then $p_j = 0$ or 1 correspond to the inequality $\sum_{i \in O \cap R^j} w(x_i, y_i) \leq \infty$ and this is certainly true for any weight w .

Remark 4. Theorem 4 establishes that whenever (1.12) holds \mathbf{h} will have a direction along which the oracle empirical risk of the non-contaminated data decreases. Figure 1.2 clearly illustrates

that (1.12) is more likely to be satisfied for the ARB- γ than for the AdaBoost or the LogitBoost algorithm. For example, if $y_i = -1$ and $\mathbb{P}(Y = -1|X = x_i) = 0.001$, then for Real AdaBoost, $w(x_i, y_i)/w_b \simeq 32$, and for ARB-2, $w(x_i, y_i)/w_b \simeq 0.008$ where w_b is the weight for a data point (x_b, y_b) such that $F^*(x_b) = 0$. It can be seen that AdaBoost puts 4000 times more weight on this outlier data than ARB-2, and hence violates (1.12).

Condition 2. *Let the class of weak hypothesis \mathcal{H} satisfy $\lim_{T \rightarrow \infty} \inf_{f \in \mathcal{F}^T} R_\phi(f) = R_\phi^*$ for a VC-dimension $d_{VC}\{\mathcal{H}\} < \infty$. Moreover, the function ϕ is a decreasing ArchBoosting loss function that is also bounded and Lipschitz.*

For a rich class \mathcal{H} , the first part of Condition 2 is true [BT07]. The class \mathcal{T} of binary trees with the number of terminal nodes larger or equal to $d + 1$, where d is the dimension of \mathcal{X} [Bre04] satisfies it. If a loss function ϕ satisfies the second part of this condition, then both $\lim_{v \rightarrow \infty} \phi(v)$ and $\lim_{v \rightarrow -\infty} \phi(v)$ exist in \mathbb{R} , and the first derivative converges to zero away from the origin. This lessens the effect of gross outliers and in turn leads to good robust properties of the resulting estimator.

Theorem 5. *Let L_ϕ and M_ϕ be the Lipschitz constant and the maximum value of ϕ , respectively. Let $V = d_{VC}(\mathcal{H})$, $c = 24 \int_0^1 \sqrt{\log \frac{8g}{\mu^2}} d\mu$. Then, under Condition 2, (a) for sequences $T_n, \zeta_n \rightarrow \infty$ and $\delta_n \rightarrow 0$ as $n \rightarrow \infty$, there exists a sequence $E_n(\zeta_n) \rightarrow 0$ such that, with probability at least $1 - \delta_n$,*

$$\sup_{f \in \mathcal{F}^{T_n}} |\hat{R}_{\phi, n}(f) - R_\phi(f)| \leq c \zeta_n L_\phi \sqrt{\frac{(V+1)(T_n+1) \log_2\left(\frac{2(T_n+1)}{\log 2}\right)}{n}} + M_\phi \sqrt{\frac{\log \frac{1}{\delta_n}}{2n}} + E_n;$$

(b) $\sup_{f \in \mathcal{F}^{T_n}} |\hat{R}_{\phi, n}(f) - R_\phi(f)| \rightarrow 0$ a.s. if $T_n = n^{1-\varepsilon}$, $\varepsilon \in (0, 1)$; (c) with the same T_n , $R_\phi(f_n^*) \rightarrow R_\phi^*$ a.s. where $f_n^* = \arg \min_{f \in \mathcal{F}^{T_n}} R_{\phi, n}(f)$.

Theorem 5 illustrates the uniform deviation between the ϕ -risk and the empirical ϕ -risk. Note that we want $T_n \rightarrow \infty$ as $n \rightarrow \infty$ but not too fast (slower than $O(n)$). Moreover, from part

(b), there exists a sequence of samples $\{\mathcal{S}_n^*\}_{n=1}^\infty$ such that $R_\phi(\tilde{f}_n) \rightarrow R_\phi^*$ as $n \rightarrow \infty$. Here \tilde{f}_n is the optimal classifier obtained by minimizing the empirical risk on \mathcal{S}_n^* . Given any sample \mathcal{S}_n , the misclassification error of any classifier f on \mathcal{S}_n is $L(f) = \mathbb{P}(f(X) \neq Y | \mathcal{S}_n)$. The Bayes risk is then defined as $L^* = \inf_{f \in \mathcal{M}} L(f) = \mathbb{E}_X[\min(\eta(X), 1 - \eta(X))]$, where $\eta(X) = \mathbb{P}(Y = 1 | X)$ and \mathcal{M} stands for the family of all measurable functions. Next we state the intermediary lemma that connects the reference sequence \tilde{f}_n to the ArchBoost estimator F_{T_n} .

Lemma 4. *For the above reference sequence $\{\tilde{f}_n\}_{n=1}^\infty$ and a non-negative sequences $T_n = n^{1-\varepsilon}$, $\varepsilon \in (0, 1)$, and with the choice of α_t as in Theorem 1, we have as $n \rightarrow \infty$, (a) $(\hat{R}_{\phi,n}(\tilde{f}_n) - R_\phi(\tilde{f}_n))_+ \rightarrow 0$ a.s. and (b) $(\hat{R}_{\phi,n}(F_{T_n}) - \hat{R}_{\phi,n}(\tilde{f}_n))_+ \rightarrow 0$ a.s.*

Theorem 6. *Assuming conditions of Theorem 5 hold. Then, with the stopping time T_n as in Theorem 5 and the step size α_t as in Theorem 1, the ArchBoost classifier F_{T_n} satisfies $L(\text{sign}(F_{T_n})) \rightarrow L^*$ a.s. as $n \rightarrow \infty$.*

1.5 Numerical Experiments

In this section we provide an extensive simulation and real data analysis illustrating superior performance of the ArchBoost framework and ARB- γ algorithms in particular.

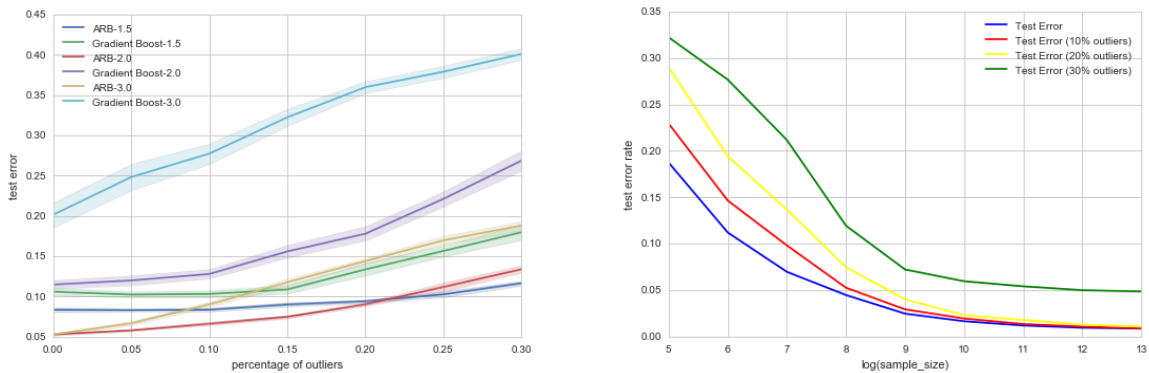
1.5.1 Gaussian - Student Mixture

In this section, design $X \sim \mathcal{N}(0, \Sigma_p)$, and we define the elliptical boundary according to the median of $\|X\|_2^2$, that is, $Y = 1$ if and only if $X^T \Sigma^{-1} X \geq \text{median}(\mathcal{X}_p^2)$. In the first example, $p = 10$ and $\Sigma_{10} = I_{10}$ with $n = 12000$ and 2000 of them are used as a training sample [FHT01]. In the second and third example, we let $[\Sigma_p]_{ij} = (0.3)^{|i-j|}$ be a Toeplitz matrix with $p = 10$ and $p = 100$ respectively. In the third example, $n = 36000$ with 6000 used for training. In all experiments, we use 5 fold cross-validation and use decision tree as the weak learner with the

hardness of the robustness belief. If one is more certain of the cleanliness of the data, larger γ may provide a compromise between robustness and non-robustness. Therefore, in practice, we recommend to choose γ to be 1.5 or 2. Choosing γ too large will depress the robustness of the algorithm, and choosing γ too close to 1 will lead to unnecessary instability.

1.5.2 Comparison with Non-convex Gradient Boost

To illustrate that non-convexity is not the only feature that enables ArchBoost to have great performance, we showcase that it behaves much better than the Gradient boost with a 1.5-robust loss function (4.34). It is worth pointing out that such Gradient boosting must be implemented using steepest descent methods and that non-convexity of the loss leads to high instability of estimates over iterations. We contrast the methods by generating samples from the model as in Figure 1.3a.



(a) ARB- γ and Gradient Boost with γ -robust loss.

(b) Consistency of ARB- γ

Figure 1.4: Comparisons with Non-convex Gradient Boost and Consistency

From Figure 1.4a we immediately observe that for every choice of γ the ARB- γ achieves lower test error than the corresponding Gradient boost with γ -robust loss with the difference being larger for larger number of outliers and larger γ . We observe that similarly as before ARB-2 achieves smallest error (5%,9%) if the percentage of outliers is smaller than 20% whereas ARB-1.5 achieves smallest error (9%,11%) if the percentage of outliers is larger than 20%. The

corresponding test errors for Gradient boost with 2 and 1.5-robust loss are much higher (ranging from (11%,19%) to (21%,35%) respectively).

1.5.3 Consistency

In order to show consistency of the proposed ArchBoost algorithms, we generate i.i.d. data from the model as in Figure 1.3a but now varying sample sizes $\exp(k) + 20000$, for $k = 5, 6, \dots, 13$. Then we use $\exp(k)$ data for training and the rest 20000 for testing. In Figure 1.4b, we can see that the test error is indeed decreasing to 0 for various percentages of outliers. The higher the number of outliers the larger the sample size n should be for the algorithm to converge. This is not unexpected as the outliers are effectively eating up (shrinking) the sample size (the algorithm is discarding them successively in each iteration).

1.5.4 The Long/Servedio problem

[LS10] constructed a challenging experiment with $X \in \mathbb{R}^{21}$ with binary features $X_i \in \{-1, +1\}$ and label $y_i \in \{-1, +1\}$. First, the label y is chosen to be -1 or $+1$ with equal probability. Then for any given y , the features X_i are generated according to the following mixture distribution:

- **Large margin:** With probability $\frac{1}{4}$, set $X_i = y$ for all $1 \leq i \leq 21$.
- **Pullers:** With probability $\frac{1}{4}$, set $X_i = y$ for $1 \leq i \leq 11$ and $X_i = -y$ for $12 \leq i \leq 21$.
- **Penalizers:** With probability $\frac{1}{2}$, randomly choose 5 coordinates from the first 11 features and 6 from the last 10 to be equal to y . The remaining features are set to $-y$.

We generate 800 samples and flip each label with probability $\epsilon \in [0, 0.5)$. The data from this distribution can be perfectly classified by $\text{sign}(\sum_i X_i)$. The classifiers are trained using the noisy data and tested on the original clean data [Fre09]. In total, 20 datasets are generated, and

on each of them, 10% of the labels were flipped. Stopping times of the algorithms are $T \leq 800$. The average test errors and sample deviations are reported in Table 1.3, from which we conclude that the ARB-2 outperforms Real AdaBoost and LogitBoost, and is even better than RobustBoost (target parameter $\theta = 0.15$).

Table 1.3: Long/Servedio problem

| data type | Real AdaBoost | LogitBoost | RobustBoost ($\theta = 0.15$) | ARB-2 |
|---------------------------|---------------|---------------|------------------------------------|---------------------|
| noise($\epsilon = 0.1$) | 28.24%(1.53%) | 26.61%(1.51%) | 11.04%(0.67%) | 9.82%(0.43%) |
| clean | 25.07%(1.92%) | 22.59%(1.74%) | 0.21%(0.35%) | 0.02%(0.04%) |

Figure 1.5 shows the average test errors and the 95% confidence intervals of different ARB- γ algorithms. The conclusion is that ARB-1.5 behaves uniformly better than all the other algorithms. The breakdown point will get higher when $\gamma \rightarrow 1^+$, implying that smaller γ leads to better robustness properties.

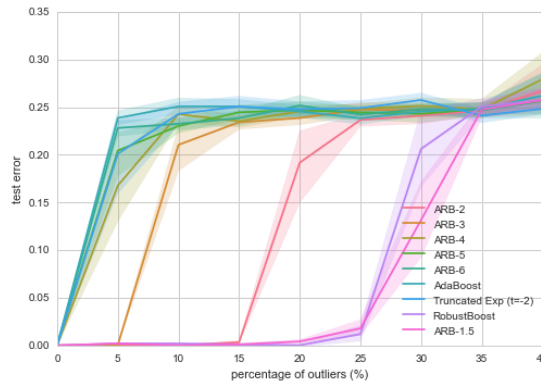


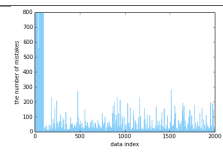
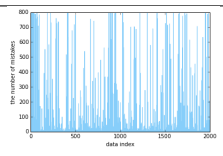
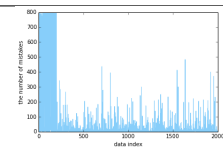
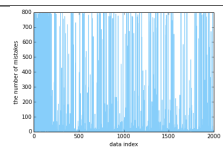
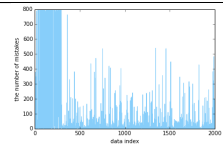
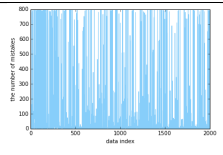
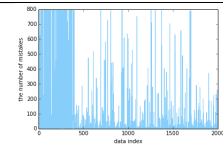
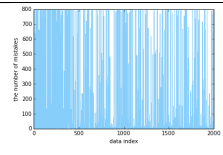
Figure 1.5: Comparison of ARB- γ on Long/Servedio problem with different ϵ

1.5.5 Outlier detection

In this experiment, we generate 2000 data points as in Section 1.5.1, and add noise to the first ϵ percentage. After 800 iterations, we record the times that each data point is misclassified, and count how many of the points that are misclassified more than 600 times (denoted as T)

actually belong to the noisy set (denoted as T_o). The ratio T_o/T and the results are shown in Table 1.4. When the percentage of outliers is less than 15%, for the ARB-2, more than 99% of the points that have been misclassified for more than 600 times are indeed the outliers, but for the Real AdaBoost, this number is only around 31%. Informally, for ARB-2, when $\epsilon \leq 15\%$, we have more than 99% "confidence" to conclude that a data point, which is misclassified for more than 600 times, is indeed an outlier.

Table 1.4: Outliers detection. The x-axis stands for the index of the training points ranging from 1 to 2000, and the y-axis stands for the times a point is misclassified, ranging from 0 to 800.

| ϵ | ARB-2 | AdaBoost | ϵ | ARB-2 | AdaBoost |
|------------|--|--|------------|---|--|
| 0.05 |  |  | 0.1 |  |  |
| T_o/T | 100% | 30.49% | T_o/T | 100% | 32.22% |
| 0.15 |  |  | 0.2 |  |  |
| T_o/T | 99.04% | 37.38% | T_o/T | 85.48% | 44.40% |

1.5.6 Real data application

We consider the Wisconsin (diagnostic) breast cancer data ² with ten real-valued features computed for each cell nucleus ³ for 569 individuals, with 357 benign and 212 malignant cells. The training set has 150 benign samples and 150 malignant samples, randomly obtained. The maximum stopping time is set to be 200. We use tree stump as the weak learner in all three problems. Results are reported in Table 1.5 and in Figure 1 in the Supplement.

Observe that ARB-2 behaves the best on the original data set, and ARB-1.5 outperforms others in the presence of noise. Compared to [SWW14] and their test error rate of 4%, our

²[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

³radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension

Table 1.5: Comparison of the average test errors and sample deviation (over 100 repetitions and using five-fold cross-validation) of four algorithms on the Wisconsin breast cancer dataset.

| Percentage of flipped labels | Methods | | |
|------------------------------|-------------------|----------------------|--------------|
| | ARB-2 | ARB-1.5 | Robust Boost |
| 0% | 3.47%(1.41%) | 3.43% (1.34%) | 4.71%(1.70%) |
| 5% | 4.80%(1.79%) | 4.47% (1.75%) | 4.82%(1.66%) |
| 10% | 5.85%(1.82%) | 5.11% (1.79%) | 5.44%(1.81%) |
| 15% | 6.67%(2.18%) | 5.92% (2.22%) | 6.53%(2.20%) |
| | GradientBoost-1.5 | LogitBoost | Ada Boost |
| 0% | 5.44%(1.76%) | 4.82%(1.85%) | 4.06%(1.58%) |
| 5% | 6.29%(1.81%) | 5.64%(1.97%) | 5.43%(2.04%) |
| 10% | 7.34%(1.99%) | 6.19%(1.81%) | 6.33%(1.85%) |
| 15% | 8.11%(2.46%) | 6.83%(2.28%) | 7.07%(2.37%) |

methods uniformly achieve smaller and comparable test error rates on the on the clean and perturbed datasets.

Next we consider a dataset that is part of the 'MicroArray quality control II' project with accession number GSE20194⁴. The dataset contains 278 newly diagnosed breast cancer patients, aged from 26 to 79 years spanning all three major races and their mixtures. Estrogen-receptor status helps guide treatment for breast cancer patients. Of 278 patients, 164 had positive estrogen-receptor status (PERS) and 114 have negative estrogen-receptor status (NERS). Each sample includes 22283 biomarker probe-sets. We choose 3000 probe-sets with the smallest p-values in the two-sample t-test (e.g. [ZWWL16]). We randomly choose 50 samples with PERS and 50 samples with NERS for a training set. Then the labels of the training samples are randomly flipped. The stopping time is set to be at most 100. Results are summarized in Table 1.6 and in Figure 2 in Supplement. The best test errors of 15% and 9% were achieved in [DM13] and [ZWWL16], respectively. However, our methods achieve errors comparable to those even when the labels were randomly perturbed.

Lastly, we compare ARB-2, ARB-1.5, RobustBoost and Real AdaBoost on the sensorless

⁴<http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE20194>

Table 1.6: Comparison of the average test errors and sample deviation (over 100 repetitions and using five-fold cross-validation) of four algorithms on the GSE20194 gene dataset.

| Percentage of flipped labels | Methods | | |
|------------------------------|-------------------|-----------------------|-----------------------|
| | ARB-2 | ARB-1.5 | RobustBoost |
| 0% | 9.40%(1.89%) | 9.31% (1.96%) | 10.19%(2.05%) |
| 5% | 10.02%(2.64%) | 9.88% (2.67%) | 11.21%(2.89%) |
| 10% | 12.04%(4.92%) | 11.97% (4.67%) | 12.39%(4.11%) |
| 15% | 15.72%(6.91%) | 15.70%(6.56%) | 14.58% (5.93%) |
| Percentage of flipped labels | Methods | | |
| | GradientBoost-1.5 | LogitBoost | AdaBoost |
| 0% | 9.87%(1.91%) | 9.54%(2.31%) | 9.63%(2.22%) |
| 5% | 10.16%(2.40%) | 10.21%(3.32%) | 10.17%(3.07%) |
| 10% | 12.31%(3.35%) | 12.14%(5.16%) | 12.34%(5.07%) |
| 15% | 16.12%(5.94%) | 16.32%(7.20%) | 16.79%(7.07%) |

drive diagnosis dataset ⁵. We have 58509 samples and each with 49 features and 11 different classes; 14000 points are chosen and then from these, 2000 are used for training and 2000 for validation. The stopping times are set ≤ 3000 . The test errors on clean data are summarized in Table 1.7 and Figure 3 in the Supplement. RobustBoost behaves worse and the best for 10% or 15% and 0% of the labels flipped, respectively. With higher levels of the noise, the test errors of ARB-1.5 and RobustBoost are very close with ARB-1.5 not needing to fine tune any target parameters.

Table 1.7: Comparison of the average test errors and sample deviation (over 100 repetitions) of four algorithms on the Sensorless drive diagnosis dataset.

| Percentage of flipped labels | Methods | | |
|------------------------------|-------------------|----------------------|-----------------------|
| | ARB-2 | ARB-1.5 | Robust Boost |
| 0% | 5.79%(0.50%) | 5.21% (0.41%) | 6.82%(0.42%) |
| 5% | 9.49%(0.69%) | 8.06% (0.83%) | 8.74%(0.67%) |
| 10% | 12.21%(0.79%) | 10.80%(0.91%) | 10.69% (0.85%) |
| 15% | 14.34%(1.01%) | 12.85%(0.89%) | 12.81% (1.10%) |
| Percentage of flipped labels | Methods | | |
| | GradientBoost-1.5 | LogitBoost | Ada Boost |
| 0% | 12.52%(1.45%) | 6.18%(0.48%) | 6.77%(0.50%) |
| 5% | 13.98%(1.30%) | 10.30%(0.67%) | 11.86%(0.79%) |
| 10% | 16.00%(1.41%) | 12.10%(0.72%) | 13.99%(0.80%) |
| 15% | 19.31%(1.61%) | 14.97%(0.88%) | 17.34%(0.89%) |

⁵<https://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis>

1.6 Proof of Theorems

1.6.1 Proof of Theorem 1

Proof of Theorem 1 (a). Firstly, we show that at each iteration t , as long as the empirical margin $\hat{\mu}(w_t, h_t)$ is positive, the empirical risk decreases by adding the weak hypotheses h_t to the current estimate. Secondly, we show that the weak hypothesis returned by our ArchBoost algorithm always has a positive empirical margin before convergence.

Step 1: On the sample \mathcal{S}_n , denote $\mathbf{F}_{t-1} = (F_{t-1}(x_1), \dots, F_{t-1}(x_n))$. Denote the partial derivative w.r.t. $F(X_i)$ as $g_t(X_i) = \left[\frac{\partial \hat{R}_{\phi, n}(\mathbf{F})}{\partial F(X_i)} \right]_{F(X_i)=F_{t-1}(X_i)} = \frac{1}{n} Y_i \phi'(Y_i F_{t-1}(X_i))$. Then the gradient of $\hat{R}_{\phi, n}$ at \mathbf{F}_{t-1} is $\nabla \hat{R}_{\phi, n}(\mathbf{F}_{t-1}) = \frac{1}{n} \mathbf{g}$ for $\mathbf{g} = (g_t(X_1), \dots, g_t(X_n))^\top$. Recall that $w_t(X_i, Y_i) = -\phi'(Y_i F_{t-1}(X_i))$ for each $i = 1, \dots, n$. Suppose that we choose a weak hypothesis h_t with positive empirical margin w.r.t. weights w_t , that is, $\hat{\mu}(h_t, w_t) > 0$, and denote $\mathbf{h}_t = (h_t(X_1), \dots, h_t(X_n))$. Then $\langle -\nabla \hat{R}_{\phi, n}(\mathbf{F}_{t-1}), \mathbf{h}_t \rangle = \frac{1}{n} \sum_{i=1}^n Y_i h_t(X_i) w_t(X_i, Y_i) = \hat{\mu}(h_t, w_t) > 0$, where $\langle \cdot, \cdot \rangle$ is the standard inner product in \mathbb{R}^n . Therefore, we have $\langle -\nabla \hat{R}_{\phi, n}(\mathbf{F}_{t-1}), \mathbf{h}_t \rangle > 0 \iff \hat{\mu}(w_t, h_t) > 0$. Next, observe that if $\langle -\nabla \hat{R}_{\phi, n}(\mathbf{F}_{t-1}), \mathbf{h}_t \rangle > 0$, then \mathbf{h}_t is a descending direction of $\hat{R}_{\phi, n}(\mathbf{F})$ at \mathbf{F}_{t-1} , therefore $\hat{R}_{\phi, n}[\mathbf{F}_t] = \hat{R}_{\phi, n}[\mathbf{F}_{t-1} + \alpha_t \mathbf{h}_t] < \hat{R}_{\phi, n}[\mathbf{F}_{t-1}]$. Note that an appropriate step size α_t can be found by the line search $\alpha_t = \arg \min_{\alpha} \hat{R}_{\phi, n}[\mathbf{F}_{t-1} + \alpha \mathbf{h}_t]$. In summary, if at step t , we choose a base learner h_t such that $\hat{\mu}(w_t, h_t) > 0$ and choose a suitable step size α_t either by line search or set to be appropriately small, then $\hat{R}_{\phi, n}(F_t) < \hat{R}_{\phi, n}(F_{t-1})$.

Step 2: In any region R_t^j , $h_t \equiv \gamma_t^j$. Then,

$$\begin{aligned} \langle -\mathbf{g}_t, \mathbf{h}_t \rangle &= \sum_{j=1}^{J_t} \sum_{i \in R_t^j} Y_i w_t(X_i, Y_i) \gamma_t^j \\ &= \sum_{j=1}^{J_t} \gamma_t^j \left(\mathbb{P}_{w_t}(Y = 1 | X \in R_t^j) - \mathbb{P}_{w_t}(Y = -1 | X \in R_t^j) \right) \sum_{i \in R_t^j} w_t(X_i, Y_i). \end{aligned}$$

From (1.6), we have $\frac{\phi'(-F_t(x))}{\phi'(F_t(x))} = \frac{\mathbb{P}_{w_t}(Y=1|x)}{\mathbb{P}_{w_t}(Y=-1|x)} \frac{\phi'(-F_{t-1}(x))}{\phi'(F_{t-1}(x))}$. Observe that if $\mathbb{P}_{w_t}(Y = 1|x) > \mathbb{P}_{w_t}(Y = -1|x)$, then $\frac{\phi'(-F_t(x))}{\phi'(F_t(x))} > \frac{\phi'(-F_{t-1}(x))}{\phi'(F_{t-1}(x))}$. By second part of Lemma 1, $F_t(x) > F_{t-1}(x)$, that is, $h_t(x) > 0$. Therefore, there exists a strictly increasing function θ with the only root at $1/2$ such that $\gamma_t^j = \theta(\mathbb{P}_{w_t}(Y = 1|X \in R_t^j))$. Hence, $\langle -\mathbf{g}_t, \mathbf{h}_t \rangle = \sum_{j=1}^{J_t} \theta(\mathbb{P}_{w_t}(Y = 1|X \in R_t^j))(2\mathbb{P}_{w_t}(Y = 1|X \in R_t^j) - 1) \sum_{i \in R_t^j} w_t(X_i, Y_i) \geq 0$. The last inequality is because $\theta(\mathbb{P}_{w_t}(Y = 1|X \in R_t^j))$ always has the same sign as $2\mathbb{P}_{w_t}(Y = 1|X \in R_t^j) - 1$, and “=” holds if and only if $\mathbb{P}_{w_t}(Y = 1|X \in R_t^j) = \frac{1}{2}$ for all $j = 1, \dots, J_t$.

Proof of Theorem 1 (b). Here, we develop ideas similar to the proof of Lemma 4.1 and Lemma 4.2 in [ZY05]. There are two differences here in comparison to [ZY05]. First, the loss is non-convex function and second, the optimal hypothesis is chosen differently. For $f \in \cup_{T=1}^{\infty} \mathcal{F}^T$, let $H_f \subset \mathcal{H}$ be the set that contains all weak hypotheses in f . For example, $f_1 = \sum_{h \in H_f} \alpha_1^h h$ and $f_2 = \sum_{h \in H_f} \alpha_2^h h$. Moreover, denote $\bar{f}_t = \sum_{h \in H_t} \omega_t^h h$, $F_t = \sum_{h \in H_t} \alpha_t^h h$. For notation simplicity, we denote $R = \hat{R}_{\phi, n}$ since we have fixed a loss function ϕ and sample size n . Let $s^h = \text{sign}(\omega_t^h - \alpha_t^h)$. By Taylor expansion, we have $R(F_t + \alpha_{t+1} s^h h) \leq R(F_t) + \alpha_{t+1} s^h \langle \nabla R(F_t), h \rangle + \frac{\alpha_{t+1}^2}{2} \sup_{\xi \in [0, 1]} R''_{F_t, h}(\xi \alpha_{t+1} s^h h)$, where $R_{F_t, h}(\alpha) := R(F_t + \alpha h)$. Since the Hessian of R is bounded, there exists $M > 0$ s.t. $\sup_{\xi \in [0, 1]} R''_{F_t, h}(\xi \alpha_{t+1} s^h h) < M$. Therefore, $R(F_t + \alpha_{t+1} s^h h) \leq R(F_t) + \alpha_{t+1} s^h \langle \nabla R(F_t), h \rangle + \frac{\alpha_{t+1}^2}{2} M$. By Algorithm 1 that $R(F_{t+1}) = R(F_t + \alpha_{t+1} h_{t+1})$. Moreover, by (1.6), h_{t+1} is chosen as the $\arg \min_{h \in \mathcal{H}_t} \mathbb{E}_w [R(F_t + \alpha_{t+1} h)]$. Hence, for any $h \in \mathcal{H}_t$, $\mathbb{E}_w [R(F_t + \alpha_{t+1} h_{t+1})] \leq \mathbb{E}_w [R(F_t + \alpha_{t+1} h)]$. Moreover, for any bounded random variable Z , $|\mathbb{E}_w [Z] - \mathbb{E}[Z]| \leq K$ for a positive constant K . Combining the above, we have $R(F_{t+1}) \leq R(F_t + \alpha_{t+1} s^h h) + 2\varepsilon_t + 2K$, for $\varepsilon_t = \sup_{h \in \mathcal{H}_t} \left| R(F_t + \alpha_{t+1} s^h h) - \mathbb{E} [R(F_t + \alpha_{t+1} s^h h)] \right|$. By the arguments very much similar to Lemmas S1 and S2 of the Supplement, it easy to obtain $\varepsilon_t = o_P(1)$. Since $\|\bar{f}_t - F_t\|_1 = o(\log t)$, and $\|\bar{f}_t - F_T\|_2^2 \leq \frac{\|\bar{f}_t - F_t\|_1^2}{t^{c_t}}$ where $c_t \in (0, 1)$ and $c_t \rightarrow 0$ as $t \rightarrow \infty$, we

have $\frac{\|\bar{f}_t - F_t\|_1^2}{t^{c_t}} = o\left(\frac{\log t}{t^{c_t}} \|\bar{f}_t - F_t\|_1\right)$. Hence,

$$\begin{aligned}
\|\bar{f}_t - F_t\|_2^2 (R(F_{t+1}) - 2\varepsilon_t - 2K) &= o\left[\frac{\log t}{t^{c_t}} \sum_{h \in H_t} |\alpha_t^h - \omega_t^h| R(F_t + \alpha_{t+1} s^h h)\right] \\
&= o\left[\frac{\log t}{t^{c_t}} \sum_{h \in H_t} |\alpha_t^h - \omega_t^h| \left(R(F_t) + \alpha_{t+1} s^h \langle \nabla R(F_t), h \rangle + \frac{\alpha_{t+1}^2}{2} M\right)\right] \\
&= o\left[\frac{\log t}{t^{c_t}} \|\bar{f}_t - F_t\|_1 R(F_t) + \frac{\alpha_{t+1} \log t}{t^{c_t}} \langle \nabla R(F_t), \bar{f}_t - F_t \rangle + \frac{M \alpha_{t+1}^2 \log t}{2 t^{c_t}} \|\bar{f}_t - F_t\|_1\right] \quad (1.13)
\end{aligned}$$

Now we look at the situation when $\hat{\mu}(h_k, w_k) = 0$. From part (a), this happens if and only if $\mathbb{P}_{w_k}(Y = 1 | X \in R_k^j) = \frac{1}{2}$, $\forall j$, i.e., $\nabla R(F_k) \perp \mathcal{H}$. Since $\hat{\mu}(h_t, w_t) \rightarrow 0$, $\nabla R(F_t)$ is perpendicular to $\cup_{T=1}^{\infty} \mathcal{F}^T$, and $\langle \nabla R(F_t) - \nabla R(\bar{f}_t), \bar{f}_t - F_t \rangle \rightarrow 0$ since $\bar{f}_t - F_t \in \cup_{T=1}^{\infty} \mathcal{F}^T$. Since ϕ is Lipschitz differentiable, there exists $L > 0$ s.t. $R(F_t) - R(\bar{f}_t) \leq \langle \nabla R(\bar{f}_t), F_t - \bar{f}_t \rangle + \frac{L}{2} \|\bar{f}_t - F_t\|_2^2$. Then $\langle \nabla R(\bar{f}_t), \bar{f}_t - F_t \rangle \leq R(\bar{f}_t) - R(F_t) + \frac{L}{2} \|\bar{f}_t - F_t\|_2^2$. When t is large enough, there exists sequence $\tilde{\varepsilon}_t \rightarrow 0$ s.t. $\langle \nabla R(F_t), \bar{f}_t - F_t \rangle \leq R(\bar{f}_t) - R(F_t) + \frac{L}{2} \|\bar{f}_t - F_t\|_2^2 + \tilde{\varepsilon}_t$. Then, by (1.13),

$$\begin{aligned}
&\|\bar{f}_t - F_t\|_2^2 (R(F_{t+1}) - 2\varepsilon_t - 2K) \\
&= o\left[\frac{\log t}{t^{c_t}} \|\bar{f}_t - F_t\|_1 R(F_t) + \frac{\alpha_{t+1} \log t}{t^{c_t}} \langle \nabla R(F_t), \bar{f}_t - F_t \rangle + \frac{\alpha_{t+1}^2 \log t}{2 t^{c_t}} \|\bar{f}_t - F_t\|_1 M\right] \\
&= o\left[\frac{\log t}{t^{c_t}} \|\bar{f}_t - F_t\|_1 R(F_t) + \frac{\alpha_{t+1} \log t}{t^{c_t}} (R(\bar{f}_t) - R(F_t)) + \eta_t\right], \quad (1.14)
\end{aligned}$$

where $\eta_t := \frac{\alpha_{t+1} \log t}{t^{c_t}} \left(\frac{L}{2} \|\bar{f}_t - F_t\|_2^2 + \tilde{\varepsilon}_t\right) + \frac{\alpha_{t+1}^2 \log t}{2 t^{c_t}} \|\bar{f}_t - F_t\|_1 M$. Then by dividing $\|\bar{f}_t - F_t\|_2^2$ on both sides of (1.14), we get

$$\begin{aligned}
R(F_{t+1}) &= o\left[\frac{\log t}{t^{c_t}} \frac{\|\bar{f}_t - F_t\|_1}{\|\bar{f}_t - F_t\|_2^2} R(F_t) + \frac{\alpha_{t+1} \log t}{t^{c_t} \|\bar{f}_t - F_t\|_2^2} (R(\bar{f}_t) - R(F_t)) + \bar{\eta}_t + 2\varepsilon_t + 2K\right] \\
&= o\left[\frac{\log t}{t^{c_t/2} \|\bar{f}_t - F_t\|_2} R(F_t) + \frac{\alpha_{t+1} \log t}{t^{c_t} \|\bar{f}_t - F_t\|_2^2} (R(\bar{f}_t) - R(F_t)) + \bar{\eta}_t + 2\varepsilon_t + 2K\right],
\end{aligned}$$

where $\bar{\eta}_t := \frac{\alpha_{t+1} \log t}{t^{c_t}} \left(\frac{L}{2} + \frac{\bar{\varepsilon}_t}{\|\bar{f}_t - F_t\|_2^2} \right) + \frac{\alpha_{t+1}^2 \log t}{2t^{c_t/2} \|\bar{f}_t - F_t\|_2} M$. Therefore,

$$\begin{aligned} R(F_{t+1}) - R(\bar{f}_t) &= o \left[\frac{\log t}{t^{c_t} \|\bar{f}_t - F_t\|_2} R(F_t) + \frac{\alpha_{t+1} \log t}{t^{c_t} \|\bar{f}_t - F_t\|_2^2} (R(\bar{f}_t) - R(F_t)) + \bar{\eta}_t + 2\varepsilon_t + 2K \right] \\ &\leq \frac{\xi_t \log t}{t^{c_t} \|\bar{f}_t - F_t\|_2} R(F_t) + \frac{\alpha_{t+1} \xi_t \log t}{t^{c_t} \|\bar{f}_t - F_t\|_2^2} (R(\bar{f}_t) - R(F_t)) + \xi_t \bar{\eta}_t + 2\xi_t \varepsilon_t + 2K \xi_t, \end{aligned}$$

for some sequence $\xi_t \rightarrow 0$ as $t \rightarrow \infty$. Now, for $c_t \rightarrow 0$, and with α_t satisfying conditions in (b), and by Lemma 4.2 in [ZY05], we have $R(F_{t+1}) - R(\bar{f}_t) \rightarrow 0$ as $t \rightarrow \infty$.

1.6.2 Proof of Theorem 3

With $IF(z; T, \mathbb{P}) = g_z \in H$, we can write the quantity $2\lambda g_z + \mathbb{E}_{\mathbb{P}} \phi''(Y, f_{\mathbb{P}, \lambda}(X)) g_z(X) \Psi(X)$ as $\mathbb{E}_{\mathbb{P}} \phi'(Y, f_{\mathbb{P}, \lambda}(X)) \Psi(X) - \phi'(z_y, f_{\mathbb{P}, \lambda}(z_x)) \Psi(z_x)$. By taking inner product $\langle \cdot, \cdot \rangle_H$ with g_z itself, we have

$$2\lambda \|g_z\|_H^2 + \mathbb{E}_{\mathbb{P}} \phi''(Y, f_{\mathbb{P}, \lambda}(X)) g_z^2(X) = \mathbb{E}_{\mathbb{P}} \phi'(Y, f_{\mathbb{P}, \lambda}(X)) g_z(X) - \phi'(z_y, f_{\mathbb{P}, \lambda}(z_x)) g_z(z_x). \quad (1.15)$$

Moreover, the Frechet derivative at $f_{\mathbb{P}, \lambda}$ is a zero mapping, hence,

$$2\lambda \langle f_{\mathbb{P}, \lambda}, g_z \rangle_H + \mathbb{E}_{\mathbb{P}} \phi'(Y, f_{\mathbb{P}, \lambda}(X)) g_z(X) = 0. \quad (1.16)$$

We also note that since $f_{\mathbb{P}, \lambda}$ is the global minimum, then $\lambda \|f_{\mathbb{P}, \lambda}\|_H^2 + R_{\phi}(f_{\mathbb{P}, \lambda}) \leq \lambda \|0_H\|_H^2 + R_{\phi}(0_H) = C_{\phi}$ where $C_{\phi} = R_{\phi}(0_H) = \phi(0, 0)$ is a constant, that is,

$$\lambda \|f_{\mathbb{P}, \lambda}\|_H^2 \leq \lambda \|f_{\mathbb{P}, \lambda}\|_H^2 + \mathbb{E}_{\mathbb{P}} \phi(Y, f_{\mathbb{P}, \lambda}(X)) \leq C_{\phi}. \quad (1.17)$$

Finally, we have

$$\begin{aligned}
2\lambda \|g_z\|_H^2 &\leq 2\lambda \|g_z\|_H^2 + \mathbb{E}_{\mathbb{P}} \phi''(Y, f_{\mathbb{P},\lambda}(X)) g_z^2(X) \\
&\stackrel{(i)}{=} \mathbb{E}_{\mathbb{P}} \phi'(Y, f_{\mathbb{P},\lambda}(X)) g_z(X) - \phi'(z_y, f_{\mathbb{P},\lambda}(z_x)) g_z(z_x) \\
&\stackrel{(ii)}{=} -2\lambda \langle f_{\mathbb{P},\lambda}, g_z \rangle_H - \phi'(z_y, f_{\mathbb{P},\lambda}(z_x)) g_z(z_x) \\
&\stackrel{(iii)}{\leq} 2\lambda \|f_{\mathbb{P},\lambda}\|_H \|g_z\|_H - \phi'(z_y, f_{\mathbb{P},\lambda}(z_x)) g_z(z_x) \\
&\stackrel{(iv)}{\leq} 2\sqrt{\lambda C_\phi} \|g_z\|_H + |\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))| |g_z(z_x)| \\
&= 2\sqrt{\lambda C_\phi} \|g_z\|_H + |\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))| \langle g_z, k(z_x, \cdot) \rangle_H \\
&\stackrel{(v)}{\leq} 2\sqrt{\lambda C_\phi} \|g_z\|_H + |\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))| \sqrt{\langle g_z, g_z \rangle_H} \sqrt{\langle k(z_x, \cdot), k(z_x, \cdot) \rangle_H} \\
&= 2\sqrt{\lambda C_\phi} \|g_z\|_H + |\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))| \|g_z\|_H |k(z_x, z_x)|.
\end{aligned}$$

where (i) is due to (1.15); (ii) due to (1.16); (iii) is due to the Cauchy-Schwartz inequality; (iv) is due to (1.17); (v) is again due to the Cauchy-Schwartz inequality. Since k is a bounded kernel, $\exists M_k > 0$ such that $|k(x_1, x_2)| \leq M_k$ for all $x_1, x_2 \in \mathcal{X}$. Hence, $|\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))| \|g_z\|_H |k(z_x, z_x)| \leq M_k |\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))| \|g_z\|_H$, which in turn leads to $2\lambda \|g_z\|_H^2 \leq 2\sqrt{\lambda C_\phi} \|g_z\|_H + M_k |\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))| \|g_z\|_H$ and hence $\|g_z\|_H \leq \sqrt{\frac{C_\phi}{\lambda}} + \frac{M_k |\phi'(z_y, f_{\mathbb{P},\lambda}(z_x))|}{2\lambda}$.

1.7 Acknowledgement

Chapter 1, in full, is a reprint of the material as it appears in Journal of the American Statistical Association. Li, Hanbo; Bradic, Jelena. Boosting in the presence of outliers: adaptive classification with nonconvex loss functions, JASA, 1-15, 2018. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 2

Forest-type Regression with General Losses and Robust Forest

2.1 Introduction

Since its development by Breiman [Bre01], random forest has proven to be both accurate and efficient for classification and regression problems. In regression setting, random forest will predict the conditional mean of a response variable by averaging predictions of a large number of regression trees. Later then, many other machine learning algorithms were developed upon random forest. Among them, robust versions of random forest have also been proposed using various methodologies. Besides the sampling idea [Bre01] which adds extra randomness, the other variations are mainly based on two ideas: (1) use more robust criterion to construct regression trees [GPS07, BB06, RL12]; (2) choose more robust aggregation method [Mei06, RL12, TPC06].

Meinshausen [Mei06] generalized random forest to predict quantiles by discovering that besides calculating the weighted mean of the observed response variables, one could also get information for the weighted distribution of observed response variables using the sets of local weights generated by random forest. This method is strongly connected to the adaptive nearest

neighbors procedure [LJ06] which we will briefly review in section 2.1.2. Different from classical k -NN methods that rely on pre-defined distance metrics, the dissimilarities generated by random forest are data dependent and scale-invariant.

Another state-of-the-art algorithm AdaBoost [FS95, FS96] has been generalized to be applicable to a large family of loss functions [Fri01, MBBF99b, LB16]. Recent development of more flexible boosting algorithms such as xgboost [CG16] have become the go-to forest estimators with tabular or matrix data. One way in which recent boosting algorithms have an advantage over the random forest is the ability to customize the loss function used to reduce the influence of outliers or optimize a metric more suited to the specific problem other than the mean squared error.

In this chapter, we will propose a general framework for forest-type regression which can also be applied to a broad family of loss functions. It is claimed in [Mei06] that quantile random forest is another nonparametric approach which does not minimize an empirical loss. However, we will show in fact both random forest and quantile random forest estimators can be re-derived as regression methods using the squared error or quantile loss respectively in our framework. Inspired by the adaptive nearest neighbor viewpoint, we explore how random forest makes predictions using the local weights generated by ensemble of trees, and connect that with locally weighted regression [FG96, TH87, Sta89, New94, Loa06, HL93]. The intuition is that when predicting the target value (e.g. $\mathbb{E}[Y|X = x]$) at point x , the observations closer to x should receive larger weights. Different from predefining a kernel, random forest assigns the weights data dependently and adaptively. After we illustrate the relation between random forest and local regression, we will use random forest weights to design other regression algorithms. By plugging robust loss functions like Huber loss and Tukey's redescending loss, we get forest-type regression methods that are more robust to outliers. Finally, motivated from the truncated squared error loss example, we will show that decreasing the number of nearest neighbors in random forest will also immediately improve its generalization performance.

The layout of this chapter is as follows. In Section 2.1.1 and 2.1.2 we review random forest and adaptive nearest neighbors. Section 2.2 introduces the general framework of forest-type regression. In Section 2.3 we plug in robust regression loss functions to get robust forest algorithms. In Section 2.4 we motivate from the truncated squared error loss and investigate the importance of choosing right number of nearest neighbors. Finally, we test our robust forests in Section 2.5 and show that they are always superior to the traditional formulation in the presence of outliers in both synthetic and real data set.

2.1.1 Random forest

Following the notation of Breiman [Bre01], let θ be the random parameter determining how a tree is grown, and data $(X, Y) \in \mathcal{X} \times \mathcal{Y}$. For each tree $T(\theta)$, let L be the total number of leaves, and R_l denotes the rectangular subspace in \mathcal{X} corresponding to the l -th leaf. Then for every $x \in \mathcal{X}$, there is exactly one leaf l such that $x \in R_l$. Denote this leaf by $l(x, \theta)$.

For each tree $T(\theta)$, the prediction of a new data point $X = x$ is the average of data values in leaf $l(x, \theta)$, that is, $\hat{Y}(x, \theta) = \sum_{j=1}^n w(X_j, x, \theta) Y_j$, where

$$w(X_j, x, \theta) = \frac{\mathbb{1}_{\{X_j \in R_{l(x, \theta)}\}}}{\#\{j : X_j \in R_{l(x, \theta)}\}}. \quad (2.1)$$

Finally, the conditional mean $\mathbb{E}[Y|X = x]$ is approximated by the averaged prediction of m trees, $\hat{Y}(x) = m^{-1} \sum_{t=1}^m \hat{Y}(x, \theta_t)$. After rearranging the terms, we can write the prediction of random forest as

$$\hat{Y}(x) = \sum_{i=1}^n w(X_i, x) Y_i, \quad (2.2)$$

where the averaged weight $w(X_i, x)$ is defined as

$$w(X_i, x) = \frac{1}{m} \sum_{t=1}^m w(X_i, x, \theta_t). \quad (2.3)$$

From equation (2.2), the prediction of the conditional expectation $\mathbb{E}[Y|X = x]$ is the weighted average of the response values of all observations. Furthermore, it is easy to show that $\sum_{i=1}^n w(X_i, x) = 1$.

2.1.2 Adaptive nearest neighbors

Lin and Jeon [LJ06] studies the connection between random forest and adaptive nearest neighbor. They introduced the so-called potential nearest neighbors (PNN): A sample point x_i is called a k -PNN to a target point x if there exists a monotone distance metric under which x_i is among the k closest to x among all the sample points.

Therefore, any k -NN method can be viewed as choosing k points from the k -PNNs according to some monotone metric. For example, under Euclidean metric, the classical k -NN algorithm sorts the observations by their Euclidean distances to the target point and outputs the k closest ones. This is equivalent to weighting the k -PNNs using inverse L_2 distance.

More interestingly, they prove that those observations with positive weights (2.3) all belong to the k -PNNs [LJ06]. Therefore, random forests is another weighted k -PNN method, but it assigns weights to the observations different from any k -NN method under a pre-defined monotonic distance metric. In fact, the random forest weights are adaptive to the data if the splitting scheme is adaptive.

2.2 General framework for forest-type regression

In this section, we generalize the classical random forest to a general forest-type regression (FTR) framework which is applicable to a broad family of loss functions. In Section 2.2.1, we motivate the framework by connecting random forest predictor with locally weighted regression. Then in Section 2.2.2, we formally propose the new forest-type regression framework. In Section 2.2.3, we rediscover the quantile random forest estimator by plugging the quantile loss function into our framework.

2.2.1 Squared error and random forest

Classical random forest can be understood as an estimator of conditional mean $\mathbb{E}[Y|X]$. As shown in (2.2), the estimator $\hat{Y}(x)$ is weighted average of all response Y_i 's. This special form reminds us of the classical least squares regression, where the estimator is the sample mean. To be more precise, we rewrite (2.2) as

$$\sum_{i=1}^n w(X_i, x)(Y_i - \hat{Y}(x)) = 0. \quad (2.4)$$

Equation (2.4) is the estimating equation (first order condition) of the locally weighted least squares regression [RW94]:

$$\hat{Y}(x) = \arg \min_{\lambda \in \mathbb{R}} \sum_{i=1}^n w(X_i, x)(Y_i - \lambda)^2 \quad (2.5)$$

In classical local regression, the weight $w(X_i, x)$ serves as a local metric between the target point x and observation X_i . Intuitively, observations closer to target x should be given more weights when predicting the response at x . One common choice of such local metric is kernel $K_h(X_i, x) = K((X_i - x)/h)$. For example, the tricube kernel $K(u) = (1 - |u|^3)^3 \mathbb{1}(|u| \leq 1)$ will ignore the impact of observations outside a window centered at x and increase the weight of an

observation when it is getting closer to x . The form of kernel-type local regression is as follows:

$$\arg \min_{\lambda \in \mathbb{R}} \sum_{i=1}^n K_h(X_i - x)(Y_i - \lambda)^2,$$

The random forest weight $w(X_i, x)$ (2.3) defines a similar data dependent metric, which is constructed using the ensemble of regression trees. Using an adaptive splitting scheme, each tree chooses the most informative predictors from those at its disposal. The averaging process then assigns positive weights to these training responses, which are called voting points in [LJ06]. Hence via the random forest voting mechanism, those observations close to the target point get assigned positive weights equivalent to a kernel functionality [FHT01].

2.2.2 Extension to general loss

Note that the formation (2.5) is just a special case when using squared error loss $\phi(a, b) = (a - b)^2$. In more general form, we have the following local regression problem:

$$\hat{Y}(x) = \arg \min_{s \in \mathcal{F}} \sum_{i=1}^n w(X_i, x) \phi(s(X_i), Y_i) \quad (2.6)$$

where $w(X_i, x)$ is a local weight, \mathcal{F} is a family of functions, and $\phi(\cdot)$ is a general loss. For example, when local weight is a kernel and \mathcal{F} stands for polynomials of a certain degree, it reduces to local polynomial regression [FG96]. Random forest falls into this framework with squared error loss, a family of constant functions and local weights (2.3) constructed from ensemble of trees.

Algorithm 3 Forest-type regression

Step 1: Calculate local weights $w(X_i, x)$ using ensemble or trees.

Step 2: Choose a loss $\phi(\cdot, \cdot)$ and a family \mathcal{F} of function. Then do the locally weighted regression

$$\hat{Y}(x) = \arg \min_{s \in \mathcal{F}} \sum_{i=1}^n w(X_i, x) \phi(Y_i, s(X_i)).$$

In Algorithm 3, we summarize the forest-type regression as a general two-step method. Note that here we only focus on local weights generated by random forest, which uses ensemble of trees to recursively partition the covariate space \mathcal{X} . However, there are many other data dependent dissimilarity measures that can potentially be used, such as k -NN, m_p -dissimilarity [ATHW14], shared nearest neighbors [JP73], information-based similarity [Lin98], mass-based dissimilarity [TZC⁺16], etc. And there are many other domain specific dissimilarity measures. To avoid distraction, we will only use random forest weights throughout the rest of this paper.

2.2.3 Quantile loss and quantile random forest

Meinshausen [Mei06] proposed the quantile random forest which can extract the information of different quantiles rather than just predicting the average. It has been shown that quantile random forest is more robust than the classical random forest [Mei06, RL12]. In this section, we show quantile random forest estimator is also a special case of Algorithm 3. It is well known that the τ -th quantile of an (empirical) distribution is the constant that minimizes the (empirical) risk using τ -th quantile loss function $\rho_\tau(z) = z(\tau - \mathbb{1}_{\{z < 0\}})$ [Koe05]. Now let the loss function in Algorithm 3 be the quantile loss $\rho_\tau(\cdot)$, \mathcal{F} be the family of constant functions, and $w(X_i, x)$ be random forest weights (2.3). Solving the optimization problem

$$\hat{Y}_\tau(x) = \arg \min_{\lambda \in \mathbb{R}} \sum_{i=1}^n w(X_i, x) \rho_\tau(Y_i - \lambda),$$

we get the corresponding first order condition

$$\sum_{i=1}^n w(X_i, x) (\tau - \mathbb{1}_{\{Y_i - \hat{Y}_\tau(x) < 0\}}) = 0.$$

Recall that $\sum_{i=1}^n w(X_i, x) = 1$, hence, we have

$$\sum_{i=1}^n w(X_i, x) \mathbb{1} \{Y_i < \hat{Y}_\tau(x)\} = \tau. \quad (2.7)$$

The estimator $\hat{Y}_\tau(x)$ in (2.7) is exactly the same estimator proposed in [Mei06]. In particular, when $\tau = 0.5$, the equation $\sum_{i=1}^n w(X_i, x) \mathbb{1} \{Y_i < \hat{Y}_{0.5}(x)\} = 0.5$ will give us the median estimator $\hat{Y}_{0.5}(x)$. Therefore, we have rediscovered quantile random forest from a totally different point of view as a local regression estimator with quantile loss function and random forest weights.

2.3 Robust forest

From the framework 3, quantile random forest is insensitive to outliers because of the more robust loss function. In this section, we test our framework on other robust losses and proposed fixed-point method to solve the estimating equation. In Section 2.3.1 we choose the famous robust loss – (pseudo) Huber loss, and in Section 2.3.2, we further investigate a non-convex loss – Tukey’s biweight.

2.3.1 Huber loss

The Huber loss [Hub64]

$$H_\delta(y) = \begin{cases} \frac{1}{2}y^2 & \text{for } |y| \leq \delta, \\ \delta(|y| - \frac{1}{2}\delta) & \text{elsewhere} \end{cases}$$

is a well-known loss function used in robust regression. The penalty acts like squared error loss when the error is within $[-\delta, \delta]$ but becomes linear outside this range. In this way, it will penalize the outliers more lightly but still preserves more efficiency than absolute deviation when data is concentrated in the center and has light tails (e.g. Normal). By plugging Huber loss into the FTR

framework 3, we get a robust counterpart of random forest. The estimating equation is

$$\sum_{i=1}^n w_i(x) \text{sign}(\hat{Y}(x) - Y_i) \min(\hat{Y}(x) - Y_i, \delta) = 0. \quad (2.8)$$

Direct optimization of (2.8) with local weights is hard, hence instead we will investigate the pseudo-Huber loss (see Figure 2.1),

$$L_\delta(y) = \delta^2 \left(\sqrt{1 + \left(\frac{y}{\delta}\right)^2} - 1 \right)$$

which is a smooth approximation of Huber loss [CBFAB97].

The estimating equation

$$\sum_{i=1}^n w_i^{pH}(x) (\hat{Y}_{pH}(x) - Y_i) = 0. \quad (2.9)$$

is very similar to that of square error loss if we define a new weight

$$w_i^{pH}(x) = \frac{w_i(x)}{\sqrt{1 + \left(\frac{\hat{Y}_{pH}(x) - Y_i}{\delta}\right)^2}}. \quad (2.10)$$

Then the (pseudo) Huber estimator can be expressed as

$$\hat{Y}_{pH}(x) = \frac{\sum_{i=1}^n w_i^{pH}(x) Y_i}{\sum_{i=1}^n w_i^{pH}(x)}. \quad (2.11)$$

Informally, the estimator (2.11) can be viewed as a weighted average of all the responses Y_i 's.

From (2.10), we know the new weight for pseudo-Huber loss has an extra scaling factor

$$\left(\sqrt{1 + (\delta^{-1}u)^2} \right)^{-1} \quad (2.12)$$

and hence will shrink more to zero whenever $\delta^{-1}|\hat{Y}_{pH}(x) - Y_i|$ is large. The tuning parameter

δ acts like a control of the level of robustness. A smaller δ will lead to more shrinkage on the weights of data that have responses far away from the estimator.

The estimating equation (2.9) can be solved by fix-point method which we propose in Algorithm 4. For notation simplicity, we will use $w_{i,j}$ to denote $w(X_i, x_j)$, where X_i is the i -th training point and x_j is the j -th testing point. The convergence to the unique solution (if exists) is guaranteed by Lemma 5.

Algorithm 4 pseudo-Huber loss (δ)

Input: Test points $\{x_j\}_{j=1}^m$, initial guess $\{\hat{Y}^{(0)}(x_j)\}$, local weights $w_{i,j}$, training responses $\{Y_i\}_{i=1}^n$, and error tolerance ϵ_0 .

while $\epsilon > \epsilon_0$ **do**

(a) Update the weights

$$w_{i,j}^{(k)} = \frac{w_{i,j}}{\sqrt{1 + \left(\frac{\hat{Y}^{(k-1)}(x_j) - Y_i}{\delta}\right)^2}}$$

(b) Update the estimator

$$\hat{Y}^{(k)}(x_j) = \frac{\sum_{i=1}^n w_{i,j}^{(k)} Y_i}{\sum_{i=1}^n w_{i,j}^{(k)}}$$

(c) Calculate error

$$\epsilon = \frac{1}{m} \sum_{j=1}^m \left(\hat{Y}^k(x_j) - \hat{Y}^{(k-1)}(x_j) \right)^2$$

(d) $k \leftarrow k + 1$

end while

Output the pseudo-Huber estimator:

$$\hat{Y}_{pH}(x_j) = \hat{Y}^{(k)}(x_j)$$

Lemma 5. *Define*

$$K_{\delta}(y) = \frac{\sum_{i=1}^n \frac{w_i Y_i}{\sqrt{1 + \left(\frac{y - Y_i}{\delta}\right)^2}}}{\sum_{i=1}^n \frac{w_i}{\sqrt{1 + \left(\frac{y - Y_i}{\delta}\right)^2}}},$$

where $\sum_{i=1}^n w_i = 1$. Let $K = \max_{i=1, \dots, n} |Y_i|$. Then Algorithm 4 can be written as $\hat{Y}^{(k)}(x) =$

$K_\delta(\hat{Y}^{(k-1)})$, and converges exponentially to a unique solution as long as $\delta > 2K$.

From Lemma 5, we know it is important to standardize the responses Y_i so that δ will be of the same scale for different problems. In practice, we observe that one will not need to choose δ that satisfies the worst-case condition $\delta > K$ in order for convergence, but making δ too small does lead to slow convergence rate. For assigning the initial guess $\hat{Y}^{(0)}$, two simplest ways are to either take the random forest estimator we got or a constant vector equaling to the sample mean. Throughout the rest of this paper, we will choose the weights to be random forest weights (2.3).

2.3.2 Tukey's biweight

Non-convex function has played an important role in the context of robust regression [Hub11, HRRS11]. Unlike convex losses, the penalization on the errors can be bounded and hence the contribution of outliers in the estimating equation will eventually vanish. Our forest regression framework 3 also incorporates the non-convex losses which will show through the Tukey's biweight function $T_\delta(\cdot)$ [Hub11], which is an example of redescending loss whose derivative will vanish to zero as the input goes outside the interval $[-\delta, \delta]$. It is defined in the following way:

$$\frac{d}{dy}T_\delta(y) = \begin{cases} y \left(1 - \frac{y^2}{\delta^2}\right)^2 & \text{for } |y| \leq \delta, \\ 0 & \text{elsewhere.} \end{cases}$$

Similarly, by rearranging the estimating equation, we have

$$\hat{Y}_{tukey}(x) = \frac{\sum_{i=1}^n w^{tukey}(X_i, x) Y_i}{\sum_{i=1}^n w^{tukey}(X_i, x)}$$

where

$$w^{tukey}(X_i, x) = w(X_i, x) \max \left\{ 1 - \left(\frac{\hat{Y}_{tukey} - Y_i}{\delta} \right)^2, 0 \right\}$$

with an extra scaling factor (see Figure 2.2)

$$\max \left\{ 1 - \left(\frac{u}{\delta} \right)^2, 0 \right\}. \quad (2.13)$$

In another word, the final estimator actually only depends on data with responses inside $[-\delta, \delta]$, and the importance of any data (X_i, Y_i) will be shrinking to zero when $|\hat{Y}_{tukey}(x) - Y_i|$ gets closer to the boundary value δ .

2.4 Truncated squared loss and nearest neighbors

In this section, we will further use the framework 3 to investigate truncated squared error loss, and use this example to motivate the relation between random forest generalization performance and the number of adaptive nearest neighbors.

2.4.1 Truncated squared error

For the truncated squared error loss

$$S_{\delta}(y) = \begin{cases} \frac{1}{2}y^2 & \text{for } |y| \leq \delta, \\ \frac{1}{2}\delta^2 & \text{elsewhere} \end{cases}$$

the corresponding estimating equation is

$$\sum_{|\hat{Y}_{trunc}(x) - Y_i| \leq \delta} w(X_i, x) (\hat{Y}_{trunc}(x) - Y_i) = 0.$$

If we define a new weight

$$w^{trunc}(X_i, x) = w(X_i, x) \mathbb{1}\{|\hat{Y}_{trunc}(x) - Y_i| \leq \delta\}, \quad (2.14)$$

then the estimator for truncated squared loss is

$$\hat{Y}_{trunc}(x) = \frac{\sum_{i=1}^n w^{trunc}(X_i, x) Y_i}{\sum_{i=1}^n w^{trunc}(X_i, x)}. \quad (2.15)$$

The estimator (2.15) is like a trimmed version of the random forest estimator (2.2). We first sort $\{Y_i\}_{i=1}^n$ and trim off the responses where $|\hat{Y}_{trunc}(x) - Y_i| > \delta$. Therefore, for any truncation level δ , the estimator $\hat{Y}_{trunc}(x)$ only depends on data satisfying $|\hat{Y}_{trunc}(x) - Y_i| \leq \delta$ with the same local random forest weights (2.1).

2.4.2 Random Forest Nearest Neighbors

In classical random forest, all the data with positive weights (2.3) are included when calculating the final estimator $\hat{Y}(x)$. However, from section 2.4.1, we know in order to achieve robustness, some of the data should be dropped out of consideration. For example, using the truncated squared error loss, we will only consider the data satisfying $|Y_i - \hat{Y}_{trunc}(x)| \leq \delta$. In classical random forest, the criterion of tree split is to reduce the mean squared error, then in most cases, data points inside one terminal node will tend to have more similar responses. So informally larger $|\hat{Y}_{trim}(x) - Y_i|$ will indicate smaller local weight $w(X_i, x)$. Therefore, instead of solving for (2.15), we investigate a related estimator

$$\hat{Y}_{wt}(x) = \frac{\sum_{w(X_i, x) \geq \varepsilon} w(X_i, x) Y_i}{\sum_{w(X_i, x) \geq \varepsilon} w(X_i, x)} \quad (2.16)$$

where $\varepsilon > 0$ is a constant in $(0, 1)$. Recall that in [LJ06], they show all the observations with positive weights are considered voting points for random forest estimator. However, (2.16) implies that we should drop observations with weights smaller than a threshold in order for the robustness. More formally, let σ be a permutation such that $w(X_{\sigma(1)}, x) \geq \dots \geq w(X_{\sigma(n_0)}, x) > 0$,

then (2.2) is equivalent to

$$\hat{Y}(x) = \sum_{i=1}^{n_0} w(X_{\sigma(i)}, x) Y_{\sigma(i)}.$$

Then we can define the k random forest nearest neighbors (k -RFNN) of x to be $\{X_{\sigma(1)}, \dots, X_{\sigma(k)}\}$, $k \leq n_0$, and get predictor

$$\hat{Y}_k(x) = \sum_{i=1}^k \tilde{w}(X_{\sigma(i)}, x) Y_{\sigma(i)}, \quad (2.17)$$

where $\tilde{w}(X_{\sigma(i)}, x) = w(X_{\sigma(i)}, x) / \sum_{j=1}^k w(X_{\sigma(j)}, x)$. In the numerical experiments (Section 2.5.3), we will test the performance of the estimator (2.17) with different k , and show that by merely choosing the right number of nearest neighbors, one can largely improve the performance of classical random forest.

Shi and Horvath [SH06] proposed a similar ensemble tree based nearest neighbor method. In their approach, if the observations X_i and X_j lie in the same leaf, then the similarity between them is increased by one. At the end, the similarities are normalized by dividing the total number of trees in the forest. Therefore, their weights (similarities) $w(X_i, x)$ will be $m^{-1} \sum_{t=1}^m \mathbb{1}_{\{X_i \in R_{l(x, \theta)}\}}$ contrast to (2.3). So different from their approach, for random forest, the similarity between X_i and X_j will be increased by $1/\#\{p : X_p \in R_{l(X_i, \theta)}\}$ if they both lie in the same leaf $l(X_i, \theta)$. This means the increment in the similarity also depends on the number of data points in the leaf.

2.5 Experiments

In this section, we plug in the quantile loss, Huber loss and Tukey's biweight loss into the general forest framework and compare these algorithms with random forest. Unless otherwise stated, for both Huber and Tukey forest, the error tolerance is set to be 10^{-6} , and every forest is an ensemble of 1000 trees with maximum terminal node size 10. The robust parameter δ are set

to be 0.005 and 0.8 for Huber and Tukey forest, respectively.

2.5.1 One dimensional toy example

We generate 1000 training data points from a Uniform distribution on $[-5, 5]$ and another 1000 testing points from the same distribution. The true underlying model is $Y = X^2 + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, 1)$. But on the training samples, we choose 20% of the data and add noise $2\mathcal{T}_2$ to the responses, where \mathcal{T}_2 follows t-distribution with degree of freedom 2.

Table 2.1: Comparison of random forest (RF), quantile random forest (QRF), Huber forest (Huber) and Tukey forest (Tukey) on one dimensional example.

| MEASURE | RF | QRF | HUBER | TUKEY |
|---------|------|------|-------------|-------------|
| MSE | 2.56 | 1.88 | 1.85 | 1.82 |
| MAD | 1.20 | 1.07 | 1.06 | 1.07 |
| MAPE | 0.16 | 0.13 | 0.12 | 0.12 |

In Figure 2.3, we plot the true squared curve and different forest predictions. It is clear that Huber and Tukey forest achieve competitive robustness as quantile random forest, and can almost recover the true underlying distribution, but random forest is largely impacted by the outliers. We also repeat the experiments for 20 times, and report the average mean squared error (MSE), mean absolute deviation (MAD) and median absolute percentage error (MAPE) in Table 2.1.

2.5.2 Multivariate example

We generate data from 10 dimensional Normal distribution, i.e. $X \sim \mathcal{N}_{10}(\vec{0}, \Sigma)$. Then we test out algorithms on following models.

1. $Y = \sum_{i=1}^{10} X_i^2 + \varepsilon$ and $\varepsilon \sim \mathcal{N}(0, 1)$, $\Sigma = \mathbb{I}$.
2. $Y = \sum_{i=1}^{10} X_i^2 + \varepsilon$ and $\varepsilon \sim \mathcal{N}(0, 1)$, $\Sigma = \text{Toeplitz}(\rho = 0.7)$.

Then for each model, we randomly choose η proportion of the training samples and add noise $15\mathcal{T}_2$ where \mathcal{T}_2 follows t-distribution with degree of freedom 2. The noise level $\eta \in \{0, 0.05, 0.1, 0.15, 0.2\}$. The results are summarized in Table 2.2 and 2.3. On the clean data, random forest still play the best, however, Huber forest's performance is also competitive and lose less efficiency than QRF and Tukey forest. On the noisy data, all three robust methods outperform random forest. Among them, Huber forest is most robust and stable.

Table 2.2: Comparison of the four methods in the setting (1). The average MSE is reported in first row, and average MAD in second row.

| MSE | 0% | 5% | 10% | 15% | 20% |
|-------|-------------|-------------|--------------|--------------|--------------|
| RF | 8.19 | 12.14 | 20.32 | 22.61 | 25.23 |
| QRF | 9.80 | 11.63 | 13.30 | 13.83 | 14.71 |
| HUBER | 9.02 | 9.86 | 10.40 | 10.49 | 10.88 |
| TUKEY | 10.56 | 12.41 | 18.16 | 12.34 | 16.62 |
| MAD | 0% | 5% | 10% | 15% | 20% |
| RF | 2.10 | 2.49 | 2.73 | 2.89 | 3.02 |
| QRF | 2.23 | 2.37 | 2.66 | 2.75 | 2.84 |
| HUBER | 2.20 | 2.28 | 2.36 | 2.38 | 2.43 |
| TUKEY | 2.37 | 2.45 | 2.54 | 2.52 | 2.66 |

Table 2.3: Comparison of the four methods in the setting (2).

| MSE | 0% | 5% | 10% | 15% | 20% |
|-------|-------------|--------------|--------------|--------------|--------------|
| RF | 9.21 | 13.00 | 13.69 | 14.92 | 17.78 |
| QRF | 11.47 | 12.07 | 12.21 | 12.29 | 13.16 |
| HUBER | 11.19 | 12.08 | 12.15 | 12.20 | 12.74 |
| TUKEY | 12.84 | 13.09 | 13.31 | 14.52 | 14.60 |
| MAD | 0% | 5% | 10% | 15% | 20% |
| RF | 1.88 | 2.19 | 2.74 | 2.80 | 2.83 |
| QRF | 2.06 | 2.13 | 2.28 | 2.32 | 2.41 |
| HUBER | 2.04 | 2.15 | 2.17 | 2.17 | 2.22 |
| TUKEY | 2.26 | 2.34 | 2.39 | 2.35 | 2.41 |

2.5.3 Nearest neighbors

In this section, we check how the number of adaptive nearest neighbors k in (2.17) will have impact on the performance of k -RFNN. We consider the same two models (1) and (2), and keep both training sample size and testing sample size to be 1000. The relations between MSE, MAD and the number of adaptive nearest neighbors are illustrated in Figure 2.4. Recall that k -RFNN with all 1000 neighbors is equivalent to random forest. From the figures, we clearly observe a kink at $k = 15$, which is much less than 1000.

2.5.4 Real data

We take two regression datasets from UCI machine learning repository [Lic13], and one real estate dataset from OpenIntro. For each dataset, we randomly choose $2/3$ observations for training and the rest for testing. MSE and MAD are reported by averaging over 20 trials. The results are presented in Table 2.4. To further test the robustness, we then repeat the experiment but add extra \mathcal{T}_2 noise to 20% of the standardized training data response variables everytime. The results are in Table 2.5. Robust forests outperform random forest in most of the cases except for Ames data sets, on which quantile random forest behaves poorly.

Table 2.4: Comparison of the four methods on two UCI repository datasets: (1) concrete compressive strength (CCS) [Yeh98]; (2) airfoil self-noise (Airfoil); and one OpenIntro dataset: Ames residential home sales (Ames).

| MSE | RF | QRF | HUBER | TUKEY |
|-----------------------|-------------|--------------|--------------|-------|
| CCS | 37.22 | 34.79 | 32.98 | 34.42 |
| AIRFOIL | 18.22 | 10.04 | 14.28 | 16.55 |
| AMES($\times 10^8$) | 4.51 | 12.21 | 5.22 | 5.91 |
| MAD | RF | QRF | HUBER | TUKEY |
| CCS | 4.62 | 4.25 | 4.17 | 4.30 |
| AIRFOIL | 3.45 | 2.30 | 3.08 | 3.17 |
| AMES($\times 10^4$) | 1.34 | 2.44 | 1.31 | 1.36 |

Table 2.5: Test on real data sets with extra noise.

| MSE | RF | QRF | HUBER | TUKEY |
|-----------------------|-------|--------------|--------------|-------|
| CCS | 68.51 | 39.21 | 39.05 | 40.27 |
| AIRFOIL | 18.22 | 10.04 | 14.28 | 16.55 |
| AMES($\times 10^8$) | 5.77 | 18.20 | 5.28 | 5.39 |
| MAD | RF | QRF | HUBER | TUKEY |
| CCS | 5.46 | 4.53 | 4.57 | 4.80 |
| AIRFOIL | 3.45 | 2.30 | 3.08 | 3.17 |
| AMES($\times 10^4$) | 1.64 | 3.23 | 1.47 | 1.55 |

2.5.5 Conclusion and discussion

The experimental results show that Huber forest, Tukey forest and quantile random forest are all much more robust than random forest in the presence of outliers. However, without outliers, Huber forest preserves more efficiency than the other two robust methods. We did not cross validate the parameter δ for different noise levels, so one would expect even better performance after carefully tuning the parameter.

Besides random forest weights, other data dependent similarities could also be used in Algorithm 3. We could also design loss functions which optimizes a metric for specific problems. The fixed-point method could be replaced by other more efficient algorithms. The framework could be easily extended to classification problems. All these will be potential future work.

2.6 Proof of Lemma 5

Proof. Because $\hat{Y}^{(k)}(x) = K_\delta(\hat{Y}^{(k-1)})$ which is a fixed-point method, we only need to show $|K'_\delta(y)| < 1$ in order for the existence and uniqueness of the solution. Define the normalized weight

$$\tilde{w}_i = \frac{w_i}{\sqrt{1 + \left(\frac{y - Y_i}{\delta}\right)^2}} / \sum_{i=1}^n \frac{w_i}{\sqrt{1 + \left(\frac{y - Y_i}{\delta}\right)^2}},$$

we have $\sum_{i=1}^n \tilde{w}_i = 1$, and $\left| K'_\delta(y) \right|$

$$\begin{aligned}
&\leq \left| \sum_{i=1}^n \tilde{w}_i Y_i \left(\sum_{j=1}^n (\mathbb{1}(i=j) - \tilde{w}_j) \frac{y - Y_j}{\delta^2 + (y - Y_j)^2} \right) \right| \\
&\leq 2 \sum_{i=1}^n \tilde{w}_i |Y_i| \max_{i=1, \dots, n} \left(\frac{|y - Y_i|}{\delta^2 + (y - Y_i)^2} \right) \\
&= 2 \sum_{i=1}^n \tilde{w}_i |Y_i| \frac{1}{\min_{i=1, \dots, n} \left(\frac{\delta^2}{|y - Y_i|} + |y - Y_i| \right)} \\
&\leq \max_{i=1, \dots, n} |Y_i| \frac{1}{\delta}.
\end{aligned}$$

Therefore, $\left| K'_\delta(y) \right| < \frac{1}{2}$ if $\delta > 2 \max_{i=1, \dots, n} |Y_i| = 2K$. □

2.7 Acknowledgement

Chapter 2, in full, is a version of the material as it appears in International Conference on Machine Learning. Li, Hanbo; Martin, Andrew. Forest-type regression with general losses and robust forest, ICML, 2017. The dissertation/thesis author was the primary investigator and author of this paper.

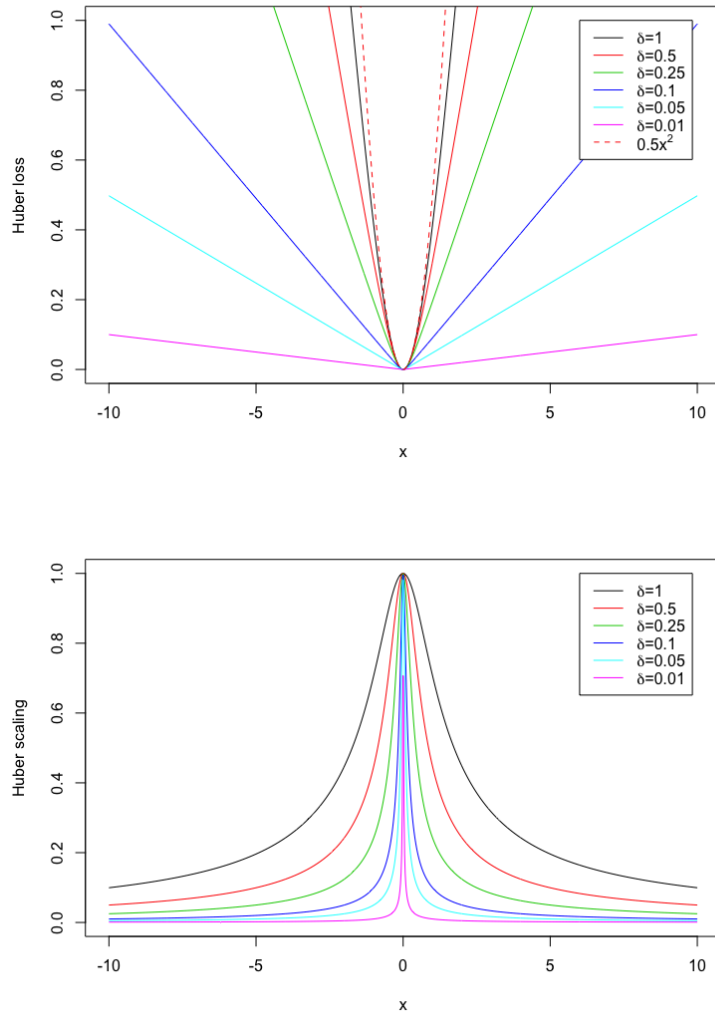


Figure 2.1: In the first row, we compare squared error loss $\frac{1}{2}x^2$ and pseudo-Huber loss with different δ . In the second row, we plot the scaling factor (2.12) of Huber loss.

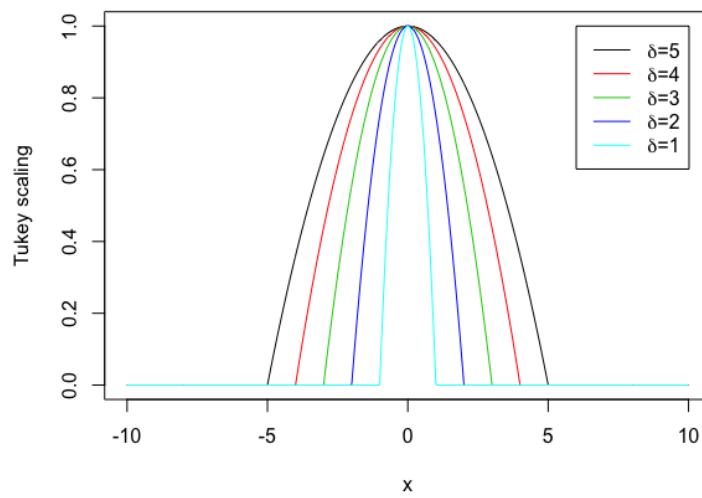


Figure 2.2: We plot the scaling factor (2.13) of Tukey's biweight. Compared to Huber scaling factor (see (2.12)), it has a hard threshold at δ .

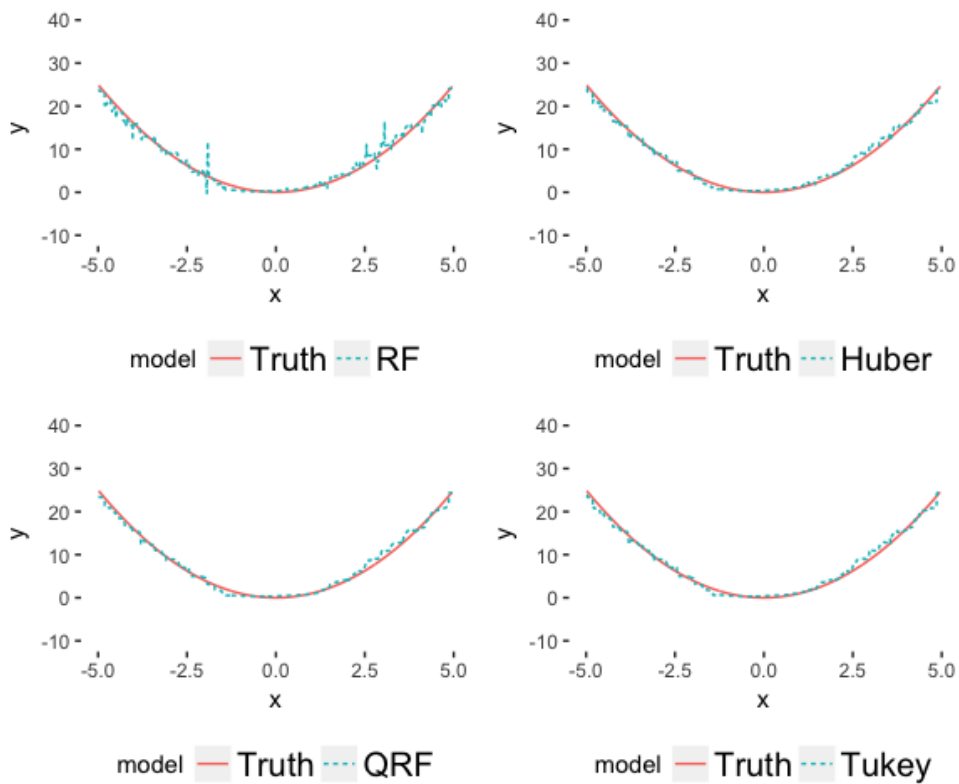


Figure 2.3: One dimensional comparison of random forest, quantile random forest, Huber forest and Tukey forest. All forests are ensemble of 500 regression trees and the maximum number of points in terminal nodes is 20.

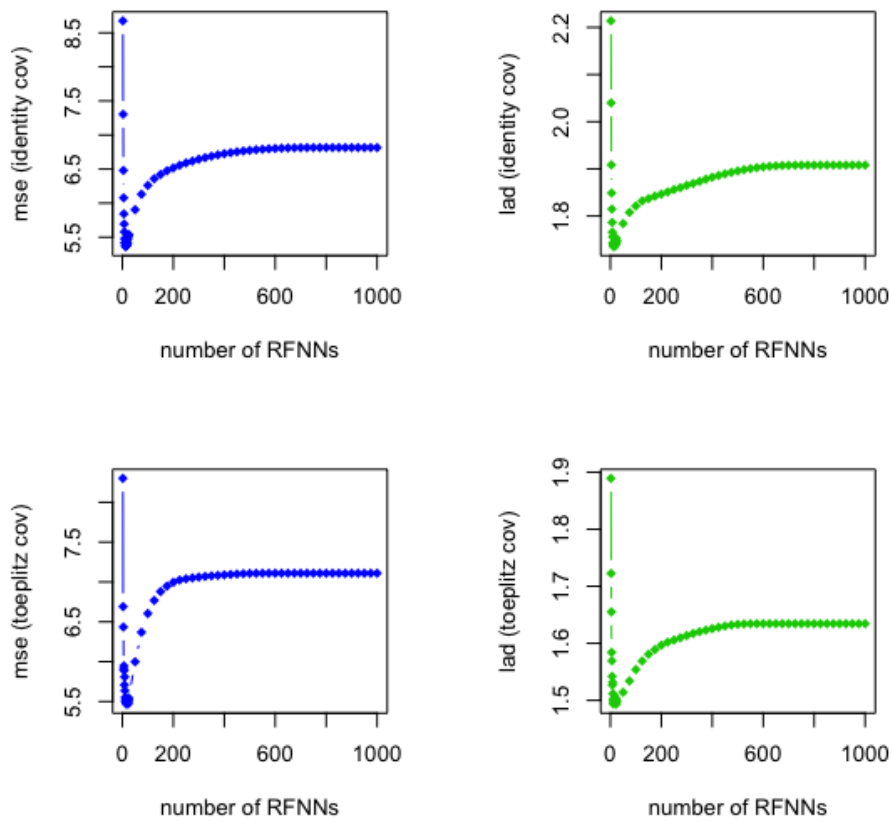


Figure 2.4: The performance of k -RFNN against the number of nearest neighbors.

Chapter 3

Forest-type Quantile Regression for Random Censored Data

3.1 Introduction

Censored data exists in many different areas. In economics, policies such as minimum wage and minimum transaction fee result in left-censored data, as quantities below the thresholds will never be observed. In bio-medical research, people are interested at the time until the occurrence of an event of interest, for example, death or occurrence of a disease. But because the time span of the study is limited or the patient quits before experiencing an event of interest, many observations will be right-censored. More formally, let T be a real-valued latent variable and X be a (possibly high-dimensional) predictor variable. In censoring problem, we have another censoring variable C which prevents us from observing all information about T . In left-censored data, we only observe $Y_i = \max(T_i, C_i)$, and in right-censored data, we only observe $Y_i = \min(T_i, C_i)$.

There have been many methods and theories dealing with censored data if we assume a

linear latent variable model

$$T_i = X_i^\top \beta + \varepsilon_i \tag{3.1}$$

with ε_i assumed to be i.i.d. with distribution function F . Under this model, the τ -th conditional quantile of T at x is $Q_{T|x}(\tau|x) = F^{-1}(\tau) + x^\top \beta$. By adding an extra dimension of one to x and absorbing the intercept $F^{-1}(\tau)$ to β , we can write the conditional quantile of T as

$$Q_{T|x}(\tau|x) = x^\top \beta \tag{3.2}$$

with the same notation x and β for simplicity. If all the censoring values C_i are also observed, [Pow84, Pow86] showed that the τ -th conditional quantile $Q(\tau|x)$ can be consistently estimated by

$$\hat{\beta} = \arg \min_{b \in \mathbb{R}^p} \sum_{i=1}^n \rho_\tau(Y_i - \min\{C_i, X_i^\top b\}) \tag{3.3}$$

for right censoring problem. For left censoring, one can simply replace min with max in (3.4). Here, $\rho_\tau(u) = u(\tau - \mathbb{1}(u < 0))$ is the τ -th quantile loss, and X_i again contains an augmented dimension of one to absorb the term $F^{-1}(\tau)$. There have been many research on how to approximately optimize (3.3) and get coefficients β .

The previous situation, in which all censoring values C_i are observed, is referred to as fixed censoring. Random censoring, in contrast, refers to situations in which we could only observe censoring values for the censored observations. For instance, for right censoring, we can observe the response $Y_i = \min(T_i, C_i)$ and the censoring indicator $\delta_i = \mathbb{1}(T_i \leq C_i)$. The usual assumption is C_i independent of T_i conditional on covariates. The Cox proportional hazard model is the dominant strategy to solve the right-censoring regression problem. However, it is only applicable to right-censored data and relies on the proportionality assumption which can sometimes be inappropriate, necessitating stratification of the baseline hazard or some other weakening of the proportional hazards condition [Koe08].

A more flexible approach for random censoring problem is to directly model conditional quantiles of the response variable. This approach offers much more flexibility as it does not assume a hazard function [Koe08]. To estimate the conditional quantiles, [Por03] proposed a recursive method which estimates a sequence of linear conditional quantile functions recursively. It can be treated as a generalization to regression of the Kaplan Meier estimator. Another closely related quantile regression model proposed by [PH08] instead makes linkage to the Nelson-Aalen estimator of the cumulative hazard function, upon which they developed a more complete asymptotic theory.

However, the parametric methods always rely on the linearity assumption on the conditional quantiles, that is,

$$Q_{\log(T)|x}(\tau|x) = x^\top \beta. \quad (3.4)$$

Here, the log transformation is arbitrary but popular in survival analysis, and can be replaced by any monotone function. This linearity assumption is too restrictive in many cases, especially when data lie on a complex manifold. Therefore, non-parametric methods are necessary.

Without censoring, for the general form of model

$$T_i = f(X_i) + u_i \quad (3.5)$$

where f is a potentially complex non-parametric function and u_i can be heteroscedastic, non-parametric methods like random forest [Bre01] and quantile random forest [Mei06] have been developed to efficiently estimate the conditional expectation and conditional quantiles of the response variable. [LM17] shows that both random forest and quantile forest belong to a general forest-type regression framework. However, all these forest algorithms cannot deal with data with censored responses. [IKBL08] proposed random survival forest that is an ensemble of survival trees to tackle the censoring problem. However, it is not directly estimating the conditional quantiles but instead estimating the cumulative hazard. Moreover, growing a survival tree

requires estimation of survival function at every split. [HBD⁺05] calculates inverse probability of censoring (IPC) weights on all observations, and then build a random forest in which each regression tree is trained on resampled observations according to IPC weights. However, the censored samples always get weights zero and will only contribute to the Kaplan-Meier estimation of the conditional censoring survival function.

This work aims to build a non-parametric conditional quantile estimator for random censored data that inherits the desirable empirical properties of regression forests, such as stability, ease of use, and flexible adaptation to different functional forms, but can be used in the wide range of censored quantile setting where the latent variable satisfies (3.5) but we can only observe $Y_i = \min(T_i, C_i)$ ($Y_i = \max(T_i, C_i)$) and the censoring indicator $\delta_i = \mathbb{1}(T_i \leq C_i)$ ($\delta_i = \mathbb{1}(T_i \geq C_i)$). We aim to showcase a solution that overcomes conceptual and methodological challenges.

We design our method as a type of adaptive and locally weighted estimators that first use a simple random forest to calculate locally weights on observations for each test point, and then solve a plug-in version of the specifically designed estimating equations using these weights. In this way we view the random forest as generating weights for local solutions of estimating equations. This locally weighting view of random forests was previously advocated by [HLBRT04] in the context of survival analysis and by [Mei06] for quantile regression, as well as in [ATW16] and [LM17] for general loss functions.

Additional challenges arise due to the random censoring nature of the observations. For fixed censoring, one observes all the censoring values and hence can straightforwardly modify the objective (3.3) to be used in the general framework [ATW16] and [LM17]. However, it is unclear how to develop a non-parametric estimator that adapts to unknown censoring in the observations. We approach this problem by carefully designing a new estimating equation which is easy to solve and can consistently predict the conditional quantiles of the latent variable.

In this paper, we propose a novel non-parametric regression method for censored data. Our method contains two steps. In the first step, we only need to build an usual random forest

on all the observations, including the censored observations, and get the random forest weight function, which is discussed in Section 3.1.1. In the second step, we can predict any conditional quantile of the latent variable T by solving an ingenious estimating equation, which is motivated by [LM17] and is designed to correct the censoring effect. The forest-type regression framework is reviewed in Section 3.1.2. Then the algorithm is described in details in Section 3.2. Then in Section 3.3, we analyze the time complexity of our algorithm and prove the statistical consistency. In Section 3.4 and 3.5, we compare our algorithm with other forest algorithms on simulated and real censored data sets.

3.1.1 Random forest weights

At a high level, trees and forests can be thought of as local smoothing methods with an adaptive neighborhood metric. Given a test point x , classical methods such as k-nearest neighbors or kernel smoother seek an appropriate number of closest points to x according to some pre-specified distance measure. In contrast, tree-based methods also seek to find training examples that are close to x , but now closeness is defined with respect to a decision tree, and the closest points to x are those that fall in the same leaf as it does. The advantage of forest based approach is its more adaptive nature of capturing model heterogeneities and changes.

Let θ denote the random parameter determining how a tree is grown, and $\{(X_i, Y_i) : i = 1, \dots, n\} \in \mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^p \times \mathbb{R}$ denote the training data. For each tree $T(\theta)$, let R_l denotes its l -th terminal leaf. Since the space \mathcal{X} is split into disjoint leaves by $T(\theta)$, we know for any $x \in \mathcal{X}$, there is exactly one leaf containing x . We let the index of the leaf be $l(x; \theta)$ and we say $x \in R_{l(x; \theta)}$.

Then for any single tree $T(\theta)$, the prediction on any data point $x \in \mathcal{X}$ is $\hat{Y}(x; \theta) = \sum_{i=1}^n w(X_i, x; \theta) Y_i$ where

$$w(X_i, x; \theta) = \frac{\mathbb{1}_{\{X_i \in R_{l(x; \theta)}\}}}{\#\{j : X_j \in R_{l(x; \theta)}\}}. \quad (3.6)$$

Then by [Bre01] and [Mei06], a random forest containing m trees has prediction of $\mathbb{E}[Y|X = x]$ as

$$\hat{Y}(x) = \sum_{i=1}^n w(X_i, x) Y_i \quad (3.7)$$

where

$$w(X_i, x) = \frac{1}{m} \sum_{t=1}^m w(X_i, x; \theta_t). \quad (3.8)$$

From now on, we call the weight $w(X_i, x)$ in (3.8) *random forest weight*. One can easily show that $\sum_{i=1}^n w(X_i, x) = 1$.

3.1.2 Forest-type regression framework

[LM17] extended random forest to a weighted regression algorithm that can use any loss function. It is shown that the way random forest combine the trees is equivalent to a weighted least square regression problem

$$\hat{Y}(x) = \arg \min_{\lambda \in \mathbb{R}} \sum_{i=1}^n w(X_i, x) (Y_i - \lambda)^2 \quad (3.9)$$

where $w(X_i, x)$ is the random forest weight. Similarly, quantile random forest [Mei06] estimator is equivalent to

$$\hat{Q}(x; \tau) = \arg \min_{\lambda \in \mathbb{R}} \sum_{i=1}^n w(X_i, x) \rho_{\tau}(Y_i - \lambda), \quad (3.10)$$

where $\rho_{\tau}(u) = u(\tau - \mathbb{1}_{u < 0})$ is the quantile loss function. And in fact, [LM17] proposed the general forest-type regression framework

$$\hat{Y}(x) = \arg \min_{s \in \mathcal{F}} \sum_{i=1}^n w(X_i, x) \phi(s(X_i), Y_i). \quad (3.11)$$

Here, $w(X_i, x)$ is the random forest weight, $\phi(\cdot)$ is a general loss function, and \mathcal{F} is a family of functions. For example, in original random forest, \mathcal{F} is the family of constant functions, and

$$\phi(u) = u^2.$$

3.2 Methodology

The quantile random forest [Mei06] cannot be directly applied to censored data $\{(X_i, Y_i)\}$ because the conditional quantile of Y is different than that of the latent variable T due to the censoring. There is no explicitly defined quantile loss function for censored data, and hence the forest framework in [LM17] cannot be directly used. However, in this section, we will propose a new estimating equation which is closely related to the estimating equation of criterion (3.10), but is designed to correct the censoring effect.

3.2.1 Motivation

In the case without censoring, plugging in the quantile loss function, we could estimate the τ -th quantile of T_i at x , denoted as $q_{\tau,x}$, to be

$$\arg \min_{q \in \mathbb{R}} \sum_{i=1}^n w(X_i, x) \rho_{\tau}(T_i - q). \quad (3.12)$$

The estimating equation of (3.12) is

$$\begin{aligned} U_n(q) &= \sum_{i=1}^n w(X_i, x) \left\{ (1 - \tau) - \mathbb{1}(T_i > q) \right\} \\ &= (1 - \tau) - \sum_{i=1}^n w(X_i, x) \mathbb{1}(T_i > q) \approx 0, \end{aligned} \quad (3.13)$$

where the second equality is true because $\sum_{i=1}^n w(X_i, x) = 1$. For simplicity and better illustration of the idea, we first assume the latent variable T_i has the same conditional probability in a neighborhood R_x of x . Out of the n data points, assume $\{X_1, \dots, X_k\} \subset R_x$ and $w(X_i, x) = 1/k$

when $X_i \in R_x$ and 0 otherwise. Now the estimating equation becomes

$$\begin{aligned} U_k(q) &= \frac{1}{k} \sum_{i=1}^k \left\{ (1 - \tau) - \mathbb{1}(T_i > q) \right\} \\ &= (1 - \tau) - \frac{1}{k} \sum_{i=1}^k \mathbb{1}(T_i > q). \end{aligned} \quad (3.14)$$

Now conditional on $\{x\} \cup \{X_i\}_{i=1}^k$,

$$\mathbb{E}[U_k(q)] = (1 - \tau) - \mathbb{P}(T > q|x)$$

which will be zero at q^* where $\mathbb{P}(T > q^*|x) = 1 - \tau$, that is, when $q^* = q_{\tau,x}$

Now suppose we further have censoring variable C_i , which is independent of T_i conditional on X_i , and we could only observe $Y_i = \min\{T_i, C_i\}$ and $\delta_i = \mathbb{1}(T_i \leq C_i)$. In order to estimate $q_{\tau,x}$, we cannot simply replace T_i with Y_i in (3.14) as the τ -th quantile of T_i is no longer the τ -th quantile of Y_i because of the censoring. In fact,

$$\mathbb{P}(Y_i > q_{\tau,x}|x) = \mathbb{P}(T_i > q_{\tau,x}|x)\mathbb{P}(C_i > q_{\tau,x}|x) = (1 - \tau)G(q_{\tau,x}|x),$$

where $G(u|x)$ is the survival function of C_i at x . In another word, the τ -th quantile of T_i is actually the $1 - (1 - \tau)G(q_{\tau,x}|x)$ -th quantile of Y_i at x . Now define a new estimating equation that is resemble of (3.14) as

$$S_k^o(q) = \frac{1}{k} \sum_{i=1}^k \left\{ (1 - \tau)G(q|x) - \mathbb{1}(Y_i > q) \right\} \approx 0. \quad (3.15)$$

If we substitute $G(q|x)$ with $G(q_{\tau,x}|x)$, an intuitive explanation for (3.15) is that because the τ -th quantile of T_i happens to be the $1 - (1 - \tau)G(q_{\tau,x}|x)$ -th quantile of Y_i at x , instead of estimating the former which is not available because of the censoring, we turn to estimate the later one. In another word, the conditional expectation, $\mathbb{E}[S_k^o(q)]$, will still be zero at the same root q^* for

(3.14). The survival function $G(\cdot|x)$ can be estimated by the Kaplan-Meier estimator $\hat{G}(\cdot|x)$ using $\{Y_i\}_{i=1}^k$ and $\{\delta_i\}_{i=1}^k$, and we then have

$$S_k(q) = \frac{1}{k} \sum_{i=1}^k \left\{ (1 - \tau) \hat{G}(q|x) - \mathbb{1}(Y_i > q) \right\} \approx 0. \quad (3.16)$$

3.2.2 Full model

In the previous section, we made an assumption that $\mathbb{P}(T|X) = \mathbb{P}(T|x)$ for all $X \in R_x$, where R_x is a neighborhood of x . But in reality, this assumption is not always true, and that is why $w(X_i, x)$ plays an important role in our final estimator, as it "corrects" the empirical probability of each T_i at x .

For example, say we have n data points $\{(X_i, T_i)\}_{i=1}^n$ and have two cases: (1) at all X_i 's we have the same conditional probability of T , i.e. $\mathbb{P}(T|X_i) = \mathbb{P}(T|X_j)$ for all i, j ; (2) T has different conditional probabilities at different locations. In the setting (1), X_i 's become irrelevant and the point mass on each T_i is $1/n$. We share the mass uniformly to the n points T_i 's as they are equally important. When $n \rightarrow \infty$, it is known that for any q ,

$$\frac{1}{n} \sum_{i=1}^n \mathbb{1}(T_i \leq q) \rightarrow \mathbb{P}(T \leq q|x). \quad (3.17)$$

However, in the case (2), the convergence (3.17) is no longer valid. We cannot simply put a mass $1/n$ on each T_i because the probability of T_i showing up at X_i could be severely different than the probability it shows up at x . An extreme example is when $\mathbb{P}(T|x) = \text{Unif}(x-1, x+1)$. Then if $|X_i - x| > 1$, any T_i showing up at X_i should not even be counted when estimating $\mathbb{P}(T|x)$ because $\mathbb{P}(T_i|x) = 0$. In another word, we should give T_i mass 0 instead of $1/n$.

Therefore, a measure of "similarity" between points X_i and x needs to come into play, because we can no longer uniformly distribute the mass since some T_i 's are more important than others for estimating $\mathbb{P}(T|x)$. For instance, if $X_i = x + 0.01$ and $X_j = x + 2$ in the previous

example, then T_i should be assigned much more weight than T_j .

Now let $w(X_i, x)$ denote the weight (mass) we assign to T_i when we are estimating $\mathbb{P}(T|x)$. In the setting (1), we just have $w(X_i, x) = 1/n$ uniformly. But in the setting (2), we should have $w(X_i, x) > w(X_j, x)$ when X_i is more similar to x than X_j in some sense. Therefore, the estimator for $\mathbb{P}(T \leq q|x)$ is then

$$\sum_{i=1}^n w(X_i, x) \mathbb{1}(T_i \leq q)$$

and it becomes clear that a proper weight $w(X_i, x)$ needs to satisfy:

$$(1) \sum_{i=1}^n w(X_i, x) = 1; \quad (2) \sum_{i=1}^n w(X_i, x) \mathbb{1}(T_i \leq q) \xrightarrow{P} \mathbb{P}(T \leq q|x) \quad \forall q. \quad (3.18)$$

However, any fixed kernel $K(X_i, x)$, even seems like a good choice at first glance, is impossible to satisfy the second condition in (3.18) for any distribution $\mathbb{P}(T|x)$. Fortunately, as shown in [Mei06], the data-adaptive random forest weight $w(X_i, x)$ introduced in Section 3.1 perfectly satisfies both conditions in (3.18). And therefore going back to (3.13), we have,

$$U_n(q_{\tau, x}) = (1 - \tau) - \sum_{i=1}^n w(X_i, x) \mathbb{1}(T_i > q_{\tau, x}) \xrightarrow{P} 0 \quad (3.19)$$

when $n \rightarrow \infty$. Then following the same logic of how we get (3.16), a heuristic extension of (3.13) to censoring case will be

$$\begin{aligned} S_n(q; \tau) &= \sum_{i=1}^n w(X_i, x) \left\{ (1 - \tau) \hat{G}(q|x) - \mathbb{1}(Y_i > q) \right\} \\ &= (1 - \tau) \hat{G}(q|x) - \sum_{i=1}^n w(X_i, x) \mathbb{1}(Y_i > q). \end{aligned} \quad (3.20)$$

3.2.3 Estimation of survival function $G(q|x)$

In the simplified example in Section 3.2.1, we assume that Y has the same conditional probability $\mathbb{P}(Y|X)$ in a neighborhood R_x of x , and hence, we can estimate $G(q|x)$ by Kaplan-

Meier estimator [KM58] (assuming no tied events)

$$\begin{aligned}\hat{G}(q|x) &= \prod_{i: X_{(i)} \in R_x, Y_{(i)} \leq q} \left(1 - \frac{1}{k-i+1}\right)^{1-\delta_{(i)}} \\ &= \prod_{i: X_i \in R_x, Y_i \leq q} \left(1 - \frac{1}{\sum_{j=1}^n \mathbb{1}(Y_j \geq Y_i) \mathbb{1}(X_j \in R_x)}\right)^{1-\delta_i}\end{aligned}\quad (3.21)$$

where $k = |R_x|$. In the more complex case like in Section 3.2.2, many consistent estimators for the conditional survival functions exists. For example, the nonparametric estimator proposed by [Ber81]

$$\tilde{G}(q|x) = \prod_{Y_i \leq q} \left\{1 - \frac{W_i(x, a_n)}{\sum_{j=1}^n \mathbb{1}(Y_j \geq Y_i) W_j(x, a_n)}\right\}^{1-\delta_i}\quad (3.22)$$

is shown to be consistent [Ber81, Dab87, Dab89, GMCS94, Akr94, LD95, VKV96]. Here, $W_i(x, a_n)$ are the Nadaraya-Watson weights

$$W_i(x, a_n) = \frac{K((x - X_i)/a_n)}{\sum_{j=1}^n K((x - X_j)/a_n)},$$

$K(\cdot)$ is a known kernel and $\{a_n\}$ is a bandwidth sequence tending to zero as n tends to infinity.

We can then simply use $\tilde{G}(q|x)$ as $\hat{G}(q|x)$ in (3.20).

However, since we already have an adaptive version of kernel – the random forest weights $w(X_i, x)$, we will propose the following two estimators for $G(q|x)$.

KM using nearest neighbors The first estimator is a resemble of (3.21). We first find the k nearest neighbors of x according to the weights $w(X_i, x)$. Denoting these points as a set N_x , then we can simply use the Kaplan-Meier estimator on N_x

$$\hat{G}(q|x) = \prod_{i: X_i \in N_x, Y_i \leq q} \left(1 - \frac{1}{\sum_{j=1}^n \mathbb{1}(Y_j \geq Y_i) \mathbb{1}(X_j \in N_x)}\right)^{1-\delta_i}.\quad (3.23)$$

Here, the number of neighbors k will be a tuning parameter.

Beran estimator with random forest weights In the second proposal, we will replace the Nadaraya-Watson weights in (3.22) with random forest weights and get

$$\hat{G}(q|x) = \prod_{Y_i \leq q} \left\{ 1 - \frac{w(X_i, x)}{\sum_{j=1}^n \mathbb{1}(Y_j \geq Y_i) w(X_j, x)} \right\}^{1-\delta_i}. \quad (3.24)$$

One could observe that (3.23) is a special case of (3.24) when the weight $w(X_i, x) = 1/k$ for $X_i \in R_x$ and 0 otherwise.

Finally, we summarize our main algorithm in Algorithm 5. The details for choosing the candidate set \mathcal{C} is in Section 3.3.1. The choice to minimize the absolute value of $S_n(q; \tau)$ is arbitrary. The goal is to find the approximate root of $S_n(q; \tau) = 0$.

Algorithm 5 Forest-type CQR

- 1: test point x , training set $\{(X_i, Y_i, \delta_i)\}_{i=1}^n$, quantile τ , node size k
 - 2: Grow a normal regression forest on the training set with each leaf containing at least k observations.
 - 3: Compute random forest weights $w(x, X_i)$ for all i as in (3.8).
 - 4: Solve for \hat{q} such that $\hat{q} \leftarrow \arg \min_{q \in \mathcal{C}} |S_n(q; \tau)|$. \mathcal{C} is a candidate set, S_n is (3.20), $\hat{G}(q|x)$ can be estimated by (3.23) or (3.24).
 - 5: **return** q^* . The τ -th quantile of latent variable T at x .
-

3.3 Theory

In this section, we will assume the random forest has terminal node size m , feature vector $X_i \in \mathbb{R}^p$, sample size is n , and k nearest neighbors are chosen in (3.23).

3.3.1 Time complexity

The step 4 in Algorithm 5 involves of finding the q^* in a candidate set \mathcal{C} that sets the estimating equation $S_n(q; \tau)$ closest to zero. For any fixed τ , $S_n(q; \tau)$ is a step function in q with jumps at Y_i 's because the discontinuities only happen at Y_i 's for $\hat{G}(q|x)$ (both (3.23) and (3.24))

and $\sum_{i=1}^n w(X_i, x) \mathbb{1}(Y_i > q)$. Therefore, the candidate set $C \subset \{Y_i\}_{i=1}^n$. And in fact, for any fixed x , only Y_i 's with the corresponding feature vector $X_i \in R_x$ (3.23) or with $w(X_i, x) > 0$ (3.24) will be jump points, and hence, $C = \{Y_i : X_i \in R_x\}$ for (3.23) or $C = \{Y_i : w(X_i, x) > 0\}$ for (3.24). We have the following theorem. The proof is given in Appendix 3.6.

Theorem 7. *For a fixed x , depending on whether $G(q|X)$ is estimated by (3.23) or (3.24), the time complexity for Algorithm 5 is $O(n \log(n))$ or $O(m \log(n)^{p-1})$, respectively.*

3.3.2 Consistency

In this section, we will show that for any fixed $\tau \in (0, 1)$, $S_n(q; \tau)$ in (3.20) will converge in probability to $(1 - \tau)G(q|x) - \mathbb{P}(Y_i > q)$ uniformly for q .

Condition 3. *The density of X is positive and bounded from above and below by positive constants on the support X .*

We note that Condition 3 is a very primitive condition on the distribution of the covariates. It is satisfied for example for Gaussian distribution and more broadly for most symmetric, continuous distributions with unbounded support. Case of bounded or discrete covariates is beyond the scope of the current work.

Condition 4. *The terminal node size $m \rightarrow \infty$ and $m/n \rightarrow 0$ as $n \rightarrow \infty$. Furthermore, for each tree splitting, every child node contains at least γ proportion of the data in the parent node, for some $\gamma \in (0, 0.5]$.*

The two conditions in Condition 4 are also required in [Mei06] (see their Assumption 2 and 3). This condition states that the leaf node size of each tree should increase with the sample size n , but at a slower rate. Intuitively, first, the trees that we are using need to be shallow (i.e., with large leaves) in order to estimate a more complex model reliably. Secondly, the sample size in each leaf must be large enough to capture the local estimating equations more adequately.

Our experiments also justify the necessity of Condition 4, as the performance of our model will deteriorate if we keep a small leaf node size but increase the sample size. We will talk about this in detail in Section 3.4.7.

Condition 5. Denote $F(y|x) = \mathbb{P}(Y \leq y|x)$. There exists a constant L such that $F(y|x)$ is Lipschitz continuous with parameter L , that is, for all $x, x' \in \mathcal{X}$,

$$\sup_y |F(y|x) - F(y|x')| \leq L \|x - x'\|_1.$$

We note that Condition 5 appears in all existing work related to quantile regression and inference thereafter.

Condition 6. The latent variable T and the censoring variable C are conditionally independent, and the conditional distribution $\mathbb{P}(T \leq q|x)$ and $\mathbb{P}(C \leq q|x)$ are both positive and strictly increasing in q for all $x \in \mathcal{X}$.

Condition 7. For any $x \in \mathcal{X}$, the estimator $\hat{G}(q|x)$ converges pointwisely to the true conditional survival function $G(q|x)$.

Condition 7 is satisfied, for example, by the Kaplan-Meier estimator (3.22) [Dab89]. Please take a look at Figure 3.4 and Figure 3.5 where we compare finite sample properties of the newly introduced estimators (3.23) and (3.24). We observe that the new distributional estimators are more adaptive and yet seemingly inherit consistency to that of the traditional KM estimator.

We proceed to showcase asymptotic properties of the proposed estimating equations. We begin by illustrating a concentration of measure phenomenon for the introduced score equations.

Theorem 8. Define

$$S(q; \tau) = (1 - \tau)G(q|x) - \mathbb{P}(Y > q). \tag{3.25}$$

Under Conditions 3 to 7, for any $x \in \mathcal{X}$, $r > 0$, $\tau \in (0, 1)$,

$$\sup_{q \in [-r, r]} |S_n(q; \tau) - S(q; \tau)| = o_p(1).$$

Proof. Conditional on X_1, \dots, X_n , the random variable $U_i = F(Y_i|X_i)$, $i = 1, \dots, n$ are i.i.d. uniform on $[0, 1]$. By Condition 6, for a given X_i ,

$$\mathbb{1}(Y_i \leq q) = \mathbb{1}(U_i \leq F(q|X_i)).$$

Then we can decompose

$$\begin{aligned} & \sum_{i=1}^n w(X_i, x) \mathbb{1}(Y_i \leq q) \\ &= \sum_{i=1}^n w(X_i, x) \mathbb{1}(U_i \leq F(q|X_i)) \\ &= \sum_{i=1}^n w(X_i, x) \mathbb{1}(U_i \leq F(q|x)) + \sum_{i=1}^n w(X_i, x) \left\{ \mathbb{1}(U_i \leq F(q|X_i)) - \mathbb{1}(U_i \leq F(q|x)) \right\}. \end{aligned}$$

The difference between the empirical distribution function and the truth can then be bounded by

$$\begin{aligned} & \left| \sum_{i=1}^n w(X_i, x) \mathbb{1}(Y_i \leq q) - F(q|x) \right| \\ & \leq \underbrace{\left| \sum_{i=1}^n w(X_i, x) \mathbb{1}(U_i \leq F(q|x)) - F(q|x) \right|}_{\text{(I)}} + \underbrace{\left| \sum_{i=1}^n w(X_i, x) \left\{ \mathbb{1}(U_i \leq F(q|X_i)) - \mathbb{1}(U_i \leq F(q|x)) \right\} \right|}_{\text{(II)}}. \end{aligned}$$

For part (I), since U_i is uniform, we have

$$\sup_{q \in \mathbb{R}} \left| \sum_{i=1}^n w(X_i, x) \mathbb{1}(U_i \leq F(q|x)) - F(q|x) \right| = \sup_{z \in [0, 1]} \left| \sum_{i=1}^n w(X_i, x) \mathbb{1}(U_i \leq z) - z \right|$$

Now since $0 \leq w(X_i, x) \leq 1/m$ and $\sum_{i=1}^n w(X_i, x) = 1$, we have

$$\sum_{i=1}^n w(X_i, x)^2 \leq \max_{i=1, \dots, n} w(X_i, x) \leq \frac{1}{m} \rightarrow 0$$

as $n \rightarrow \infty$, by Condition 4. Hence, by Chebyshev inequality, for every $z \in [0, 1]$ and $x \in \mathcal{X}$,

$$\left| \sum_{i=1}^n w(X_i, x) \mathbb{1}(U_i \leq z) - z \right| = o_p(1).$$

Then by Bonferroni's inequality,

$$\sup_{z \in [0, 1]} \left| \sum_{i=1}^n w(X_i, x) \mathbb{1}(U_i \leq z) - z \right| = o_p(1).$$

The proof of part (II)

$$\left| \sum_{i=1}^n w(X_i, x) \left\{ \mathbb{1}(U_i \leq F(q|X_i)) - \mathbb{1}(U_i \leq F(q|x)) \right\} \right| = o_p(1)$$

follows the same argument of Theorem 1 and Lemma 2 in [Mei06] by invoking Condition 4.

Finally, we notice that by Condition 7, $\sup_{q \in [-r, r]} |\hat{G}(q|x) - G(q|x)| = o(1)$ because $[-r, r]$ is compact. \square

Next, we present our main result that illustrates an asymptotic consistency of the proposed conditional quantile estimator.

Theorem 9. *Under Conditions 3 to 7, for fixed $\tau \in (0, 1)$ and $x \in \mathcal{X}$, define q^* to be the root of $S(q; \tau) = 0$, and $r > 0$ to be some constant so that $q^* \in [-r, r]$. Also define q_n to be $\arg \min_{q \in [-r, r]} |S_n(q; \tau)|$. Then $\mathbb{P}(T \leq q^* | x) = \tau$, and $q_n \xrightarrow{P} q^*$ as $n \rightarrow \infty$.*

Proof. By [VDVW96], we only need to show for any $\tau \in (0, 1)$, $x \in \mathcal{X}$,

1. $\sup_{q \in [-r, r]} |S_n(q; \tau) - S(q; \tau)| = o_p(1)$.

2. For any $\varepsilon > 0$, $\inf\{|S(q; \tau)| : |q - q^*| \geq \varepsilon, q \in [-r, r]\} > 0$.

3. $S_n(q_n; \tau) = o_p(1)$.

Part 1 has been proved by Theorem 8. For part 2, note that

$$\begin{aligned}
S(q; \tau) &= (1 - \tau)G(q|x) - \mathbb{P}(Y > q|x) \\
&= (1 - \tau)G(q|x) - \mathbb{P}(T > q|x)\mathbb{P}(C > q|x) \\
&= ((1 - \tau) - \mathbb{P}(T > q|x))G(q|x) \\
&= (\mathbb{P}(T \leq q|x) - \tau)G(q|x).
\end{aligned}$$

The second equality is because of the conditionally independency between T and C . Fix an $\varepsilon > 0$, and denote $E = \{|S(q; \tau)| : |q - q^*| \geq \varepsilon, q \in [-r, r]\}$. Since $0 < \tau < 1$, by Condition 6, there exists some $l > 0$ such that $G(q|x) \geq l$ and $|\mathbb{P}(T \leq q|x) - \tau| \geq l$ for $q \in E$. Now for part 3, by the definition of q_n , we know $|S_n(q_n; \tau)| = \min_{q \in [-r, r]} |S_n(q; \tau)|$. Also by definition of q^* , $0 = |S(q^*; \tau)| = \min_{q \in [-r, r]} |S(q; \tau)|$. Then we get

$$\begin{aligned}
|S_n(q_n; \tau)| &= |S_n(q_n; \tau)| - |S_n(q^*; \tau)| + |S_n(q^*; \tau)| - |S(q^*; \tau)| \\
&\leq |S_n(q^*; \tau) - S(q^*; \tau)| \\
&\leq \sup_{q \in [-r, r]} |S_n(q; \tau) - S(q; \tau)| \\
&= o_p(1)
\end{aligned}$$

where the first inequality is because of the definition of q_n and the triangular inequality. \square

3.4 Simulation Study

In this section, we will compare our model, censored forest regression (*crf*) with generalized random forest (*grf*) [ATW16], quantile random forest (*qrf*) [Mei06] and random survival

forest (*rsf*) [HBD⁺05] on simulated data sets.

On the simulated data sets, we will apply *qrf* and *grf* to the censored data directly, and get biased models which we denote by *qrf* and *grf*, respectively. We also apply *qrf* and *grf* to the data with uncensored responses, and call the resulted models *qrf-oracle* and *grf-oracle*.

Throughout this section, we fix the number of trees for each forest to be 1000. The only tuning parameter we have is the node size of each tree. All other parameters are kept as default.

3.4.1 Toy example

In this section, we generate latent variables $T_i \sim \text{Unif}(0, 1)$, and censoring variables $C_i \sim \mathcal{N}(0.8, 0.2^2)$. Then we have censored responses $Y_i = \min(T_i, C_i)$. We compare the estimating equation on the latent variables T_i

$$U_1(q) = (1 - \tau) - \frac{1}{n} \sum_{i=1}^n \mathbb{1}(T_i > q)$$

to the estimating equation of our proposed algorithm

$$U_2(q) = (1 - \tau)\hat{G}(q) - \frac{1}{n} \sum_{i=1}^n \mathbb{1}(Y_i > q),$$

where $\hat{G}(q)$ is the one-dimensional Kaplan-Meier estimator for the survival function of censoring variable C . The results are shown in Figure 3.1.

There we present the two estimating equations as functions of q and illustrate that the solutions to $U_1(q) = 0$ and $U_2(q) = 0$ are closer and closer together when the sample size grows. The solution for $U_1(q) = 0$ can be treated as an oracle solution where the oracle observes "uncensored" (true) response variable. In Figure 3.1 we observe that the root of our method's estimating equation is very close to the oracle root and that we are therefore finding a good approximation to the unknown parameter of interest.

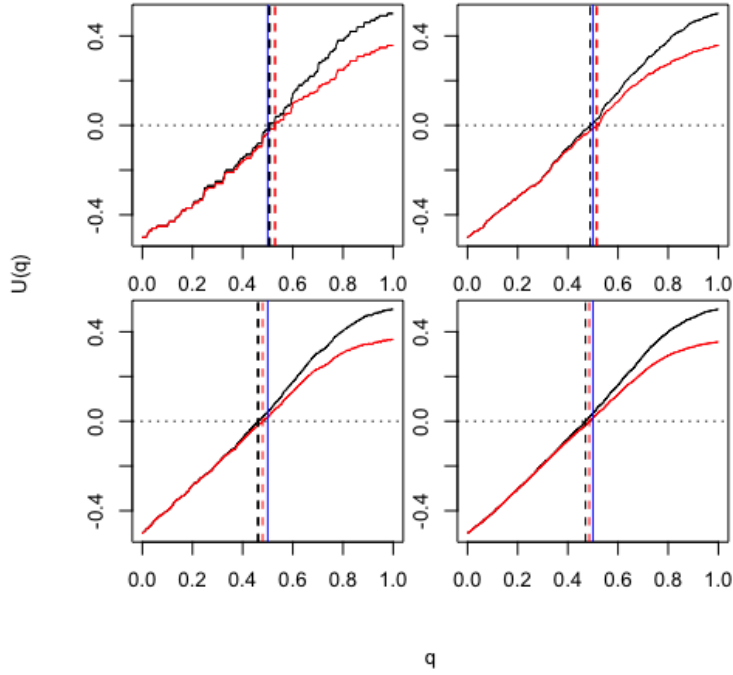


Figure 3.1: Sample loss plots when $\tau = 0.5$. In the first row, we have sample sizes 100 and 500. In the second row, we have sample sizes 1000 and 5000. The black curve is $U_1(q)$, the red curve is $U_2(q)$, the black (red) dotted vertical line is the root of $U_1(q)$ ($U_2(q)$), and the blue vertical line is $q = \tau$.

3.4.2 One-dimensional AFT model

We simulate data from an one-dimensional AFT model

$$\log(T) = X + \varepsilon$$

where $X \sim \text{Unif}(0, 2)$ and $\varepsilon \sim \mathcal{N}(0, 0.3^2)$. Then the censoring variable $C \sim \text{Exp}(\lambda = 0.08)$, and the observed response $Y = \min(T, C)$ and the censoring indicator $\delta = \mathbb{1}(T \leq C)$. The average censoring rate is about 20%. The number of training data, validation data and test data are all 300. All the forests consist of 1000 trees. The node size of each forest is determined by validation. We plot out one set of training data and the corresponding quantile predictions for $\tau = 0.3, 0.5, 0.7$ on a set of test data in Figure 3.2. We only show the results of *crf*, *grf*, and *grf-oracle* because

in one dimension, *qrf*'s performance is visually indistinguishable from *grf*. There we observe a consistency of our method as well as superior behavior to the competing method. Namely, the generalized random forest that ignores the censoring component of the data, incurs large bias; due to random censoring bias is larger for lower values of the quantiles. We observe that the proposed *crf* follows closely the oracle estimator and is extremely close to the true quantile regardless of the τ in the study.

Moreover, we proceed further and for a set of values $\tau \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, we repeat the process 40 times, and for each time, we calculate the MSE and MAD between the estimated quantiles and the true quantiles, and the τ -th quantile loss. To be more specific, let T_i be the response in test set (all uncensored), Q_i^τ be the true τ -th quantile, and \hat{Q}_i^τ be the estimated quantile, then

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{Q}_i^\tau - Q_i^\tau)^2,$$

$$L_{MAD} = \frac{1}{n} \sum_{i=1}^n |\hat{Q}_i^\tau - Q_i^\tau|,$$

$$L_{quantile} = \frac{1}{n} \sum_{i=1}^n \rho_\tau(T_i - \hat{Q}_i^\tau).$$

The reason we use $L_{quantile}$ to measure the quality of quantile predictions is that, by [Mei06], the τ -th quantile of T at x equals to $\arg \min_{q \in \mathbb{R}} \mathbb{E}[\rho_\tau(T - q) | X = x]$. The results are illustrated in Figure 3.3 where besides the above three measures we compare the concordance index (C-index) [HJCP⁺82], which is related to the area under the ROC curve [HZ05]. It estimates the probability that, in a randomly selected pair of cases, the case that fails first had a worse predicted outcome. In [IKBL08], they use the ensemble mortality as the predictive outcome for their random survival forest, and the predicted survival time for random forest regression. For our method *crf* and the other two methods, *qrf* and *grf*, we will use the τ -th conditional quantile as the predicted outcome. Since the outcomes will be different for different τ , we report the results for all $\tau \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

In Figure 3.3 we observe an oracle like behavior of the proposed *crf* method in terms all four measures of the quality of estimation and/or prediction. Namely, we observe that MAD and MSE and Quantile losses are extremely small whereas C-index is high and all are close to the corresponding oracle estimators (colored purple and blue). Moreover, we observe that the proposed *crf* method, although not primarily build for the hazard rates, is even better than survival random forest: see for example discrepancies between red and brown boxplots in the last row of Figure 3.3 where the larger the C-index is the better the method is.

3.4.3 Comparison of different conditional survival estimators

In this section, we will compare the two different conditional survival function estimators (3.23) and (3.24). We generate training data and test data from the one-dimensional AFT model defined in the previous section, but with two different censoring rate:

- $C \sim \text{Exp}(\lambda = 0.08)$, in this case, the censoring rate is about 20%.
- $C \sim \text{Exp}(\lambda = 0.20)$, in this case, the censoring rate is about 50%.

We then choose four test points $\{x_1 = 0.4, x_2 = 0.8, x_3 = 1.2, x_4 = 1.6\}$, and then plot out the conditional survival function estimators $\hat{G}(q|x_i)$ by the two different methods (3.23) and (3.24) on these four points. The results are shown in Figure 3.4 and 3.5 for three different training sample sizes $n \in \{300, 2000, 5000\}$. For the nearest neighbor estimator (3.23), we set the number of neighbors to be $n/10$, which is also the node size we choose.

We can observe that when n increases, two curves become closer and are both good approximations of the true survival curve. But the first method (3.23) does have an extra tuning parameter k – the number of nearest neighbors, so in the experiments, we always choose to use the second estimator (3.24), which is more adaptive and parameter free.

Note that the estimated survival function will degenerate at the tail of the distribution when the test point x is small (take a look at the first two columns in Figure 3.4 and 3.5). This

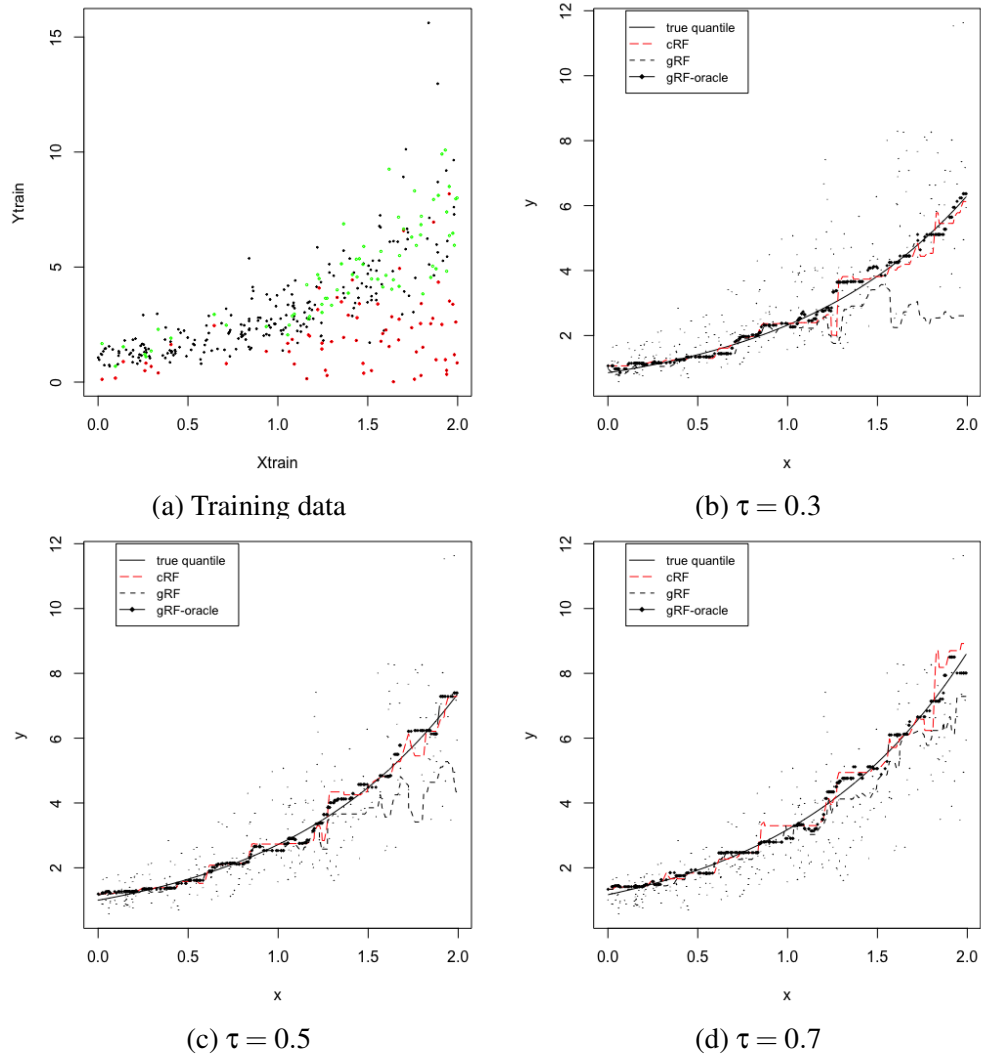


Figure 3.2: One-dimensional AFT model results. In (a), black points stand for observation that are not censored; red points are observations that are censored, and the green points are the counterpart of the red points, that is, they are the latent values of those red points if they were not censored.

is a common phenomenon even for the regular KM estimator because there is no censored observations beyond some time point. In the AFT model, the conditional distribution of the latent variable depends on the location x . When x is small, the conditional mean of T is also small, and we could not observe most of the censoring values where $C_i > T_i$, leading to degenerated survival curves. However, if we continue increasing the sample size n , we should be able to recover the entire curve even for smaller x . In fact, when we increase the censoring level from 20% (Figure

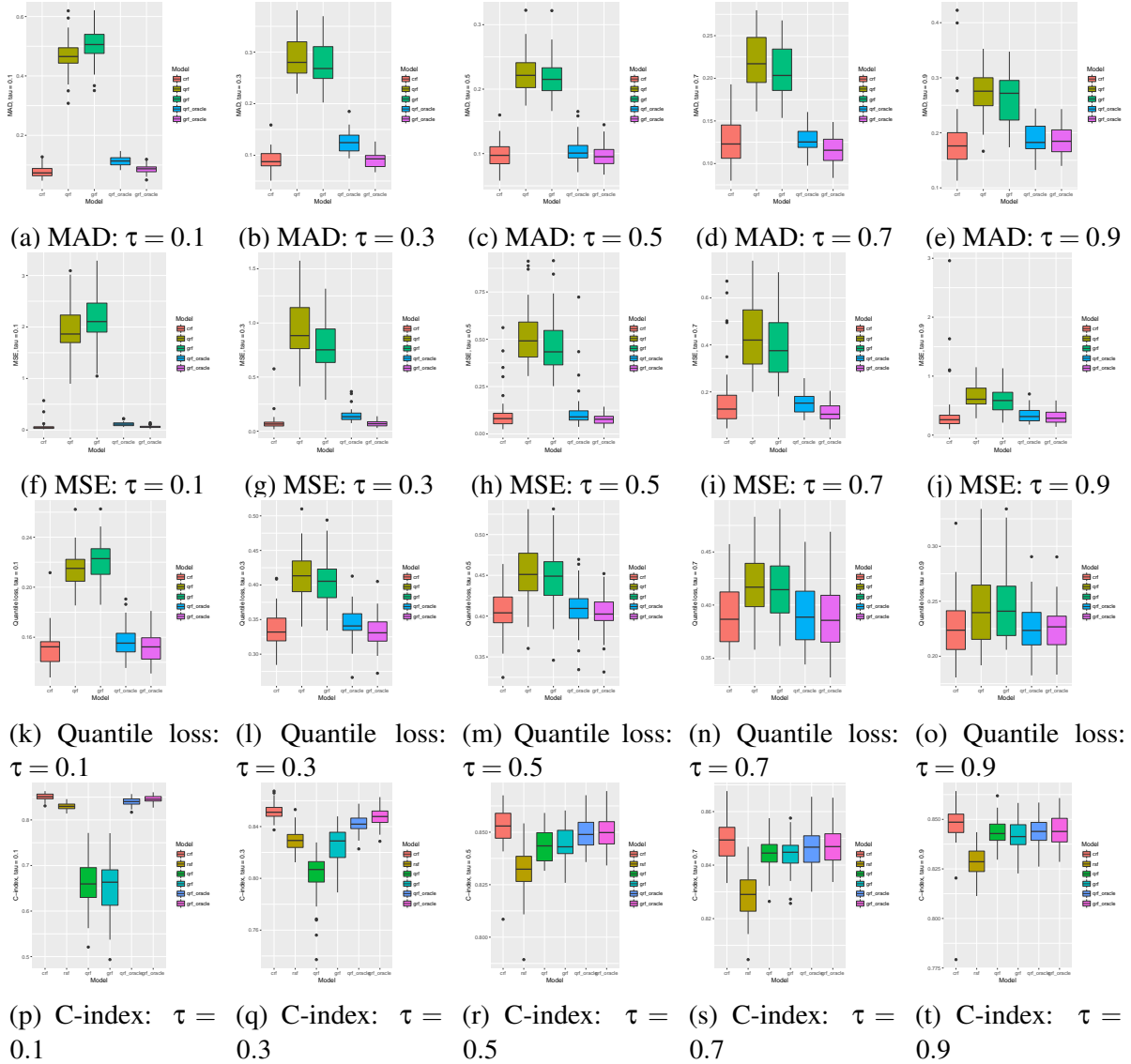


Figure 3.3: AFT 1D box plots.

3.4) to 50% (Figure 3.5), we find that both estimators give better performance because we can observe more censored values.

3.4.4 One-dimensional sine function model

Since our forest regression method *crf* is nonparametric and does not rely on any parametric assumption between response and explanatory variables, it can be used to estimate quantiles

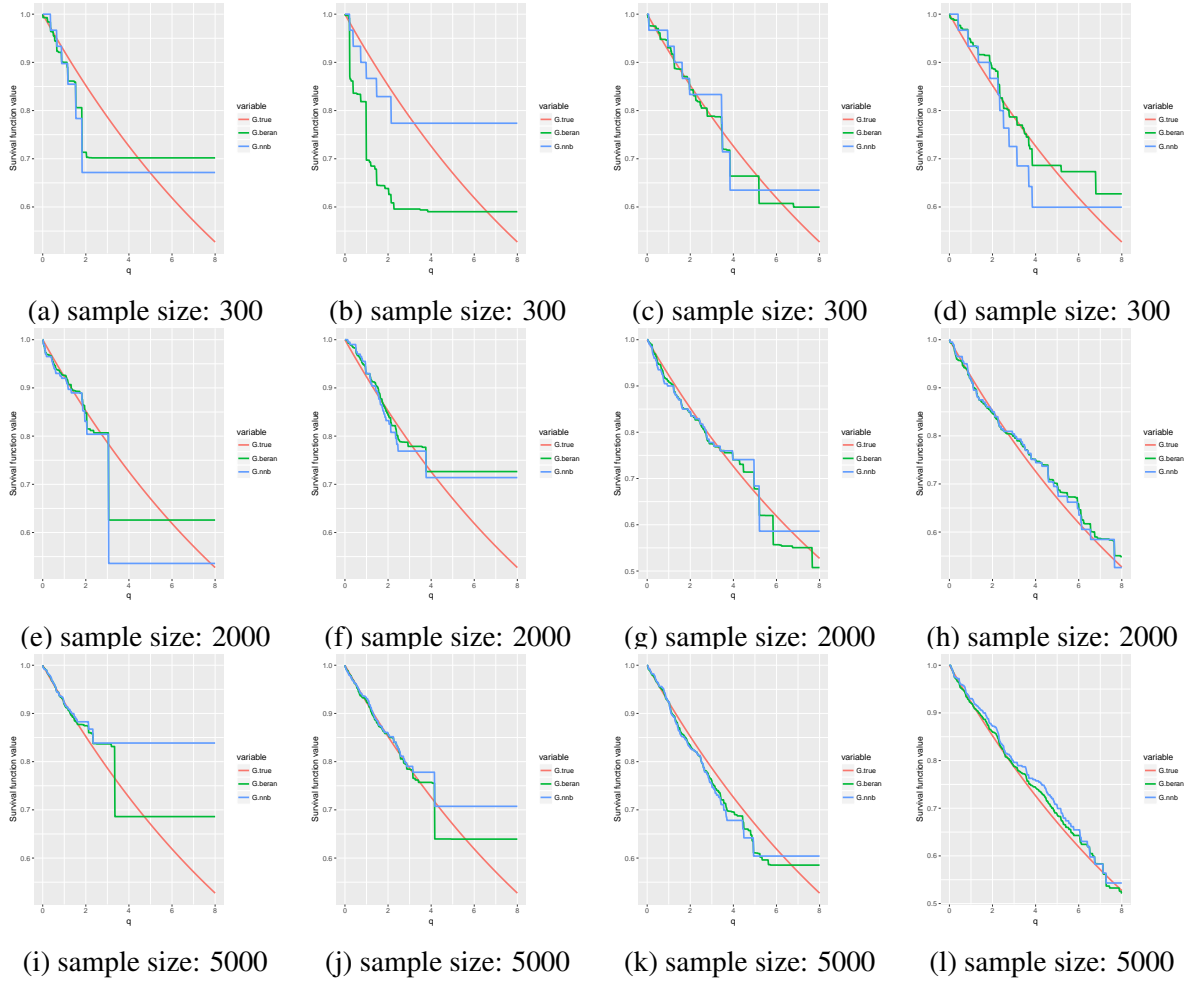


Figure 3.4: Comparison of different conditional survival estimators for the one-dimensional AFT model. The censoring variable $C \sim \text{Exp}(\lambda = 0.08)$, and the average censoring rate is around 20%. From left-most column to right-most column, we plot the conditional survival estimators for four test points, $x = 0.4, 0.8, 1.2, 1.6$.

for any general model $T = f(X) + \varepsilon$. In this section, we let $f(x) = \sin(x)$ and have the model

$$T = 2.5 + \sin(X) + \varepsilon$$

where $X \sim \text{Unif}(0, 2\pi)$ and $\varepsilon \sim \mathcal{N}(0, 0.3^2)$. Then the censoring variable $C \sim 1 + \sin(X) + \text{Exp}(\lambda = 0.2)$, and the responses $Y = \min(T, C)$. All the settings are the same as in Section 3.4.2. We plot out the training data and the quantile predictions for $\tau = 0.3, 0.5, 0.7$ in Figure 3.6.

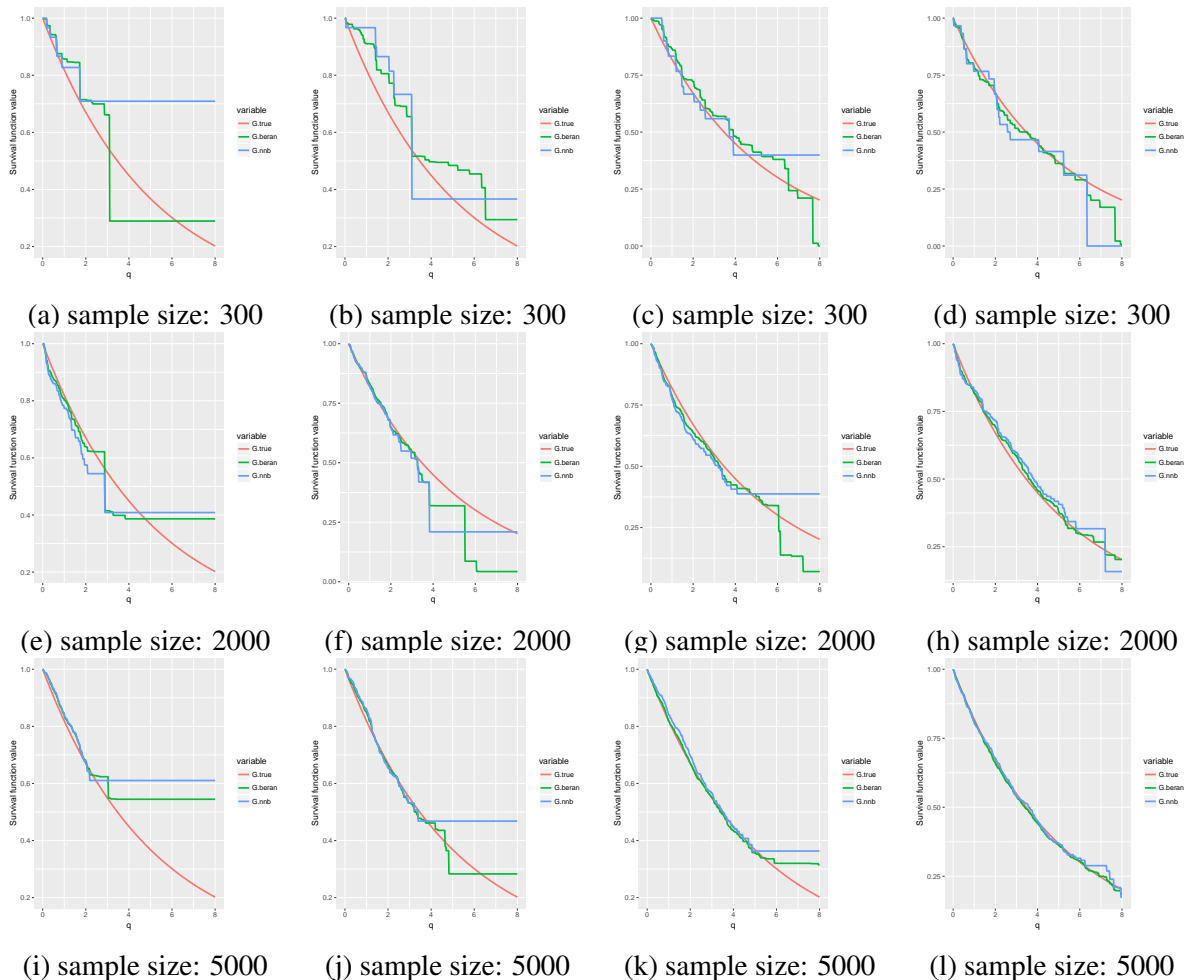


Figure 3.5: Comparison of different conditional survival estimators for the one-dimensional AFT model. The censoring variable $C \sim \text{Exp}(\lambda = 0.20)$, and the average censoring rate is around 50%. From left-most column to right-most column, we plot the conditional survival estimators for four test points, $x = 0.4, 0.8, 1.2, 1.6$.

The censoring level is about 25%. We observe that for all $\tau \in \{0.3, 0.5, 0.7\}$, crf can produce comparable quantile predictions to $grf\text{-oracle}$. Especially when $\tau = 0.3$, the quantile prediction by grf (blue dotted curve) severely deviates from the true quantile, while our method crf can still predict the correct quantile and performs as good as $grf\text{-oracle}$. We want to emphasize that $grf\text{-oracle}$ uses the latent responses T_i while our method only uses the observed responses Y_i and censoring indicators δ_i . We then repeat the experiments for 40 times and report the box plots in Figure 3.7. Again we can see that for all quantiles, our method crf behaves almost as good

as *qrf-oracle* and *grf-oracle*, and consistently better than *qrf* and *grf*. For example the order of magnitude of our error is twice and sometimes more than two times smaller than that of quantile or generalized random forest.

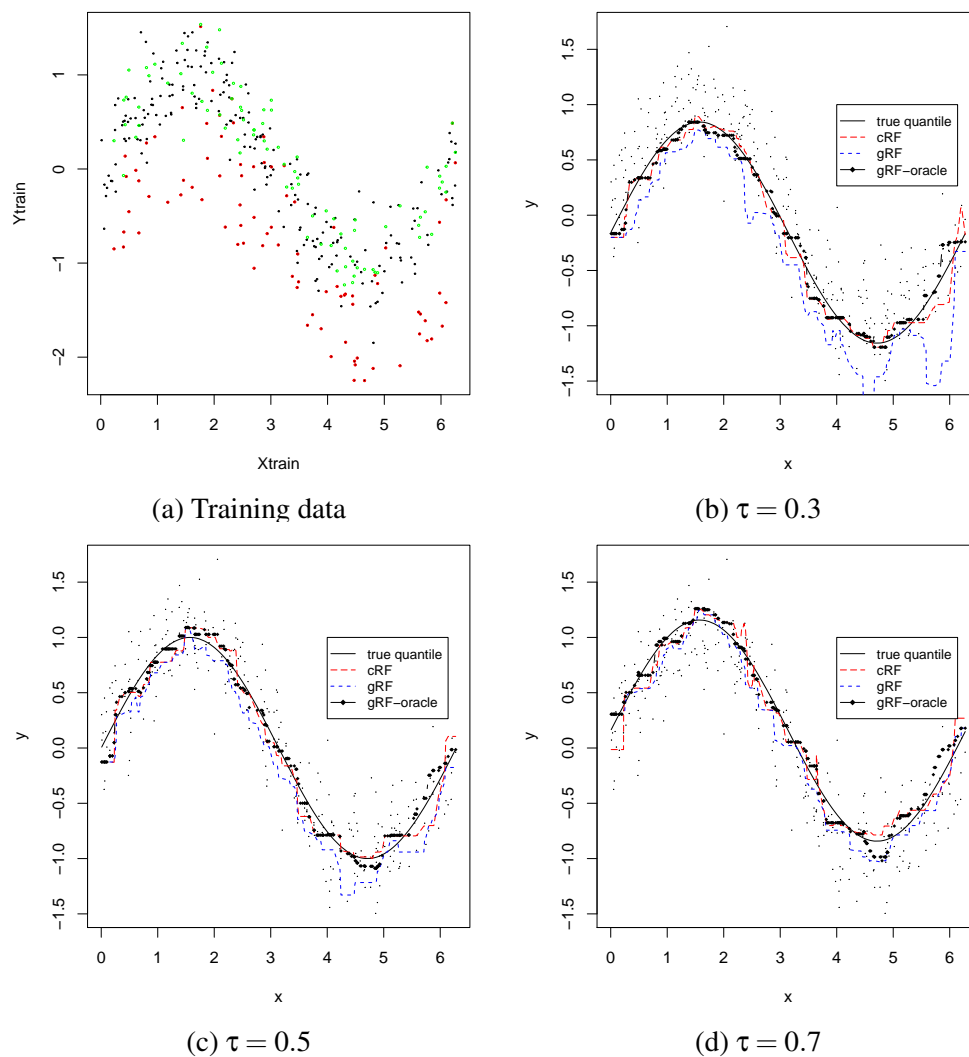


Figure 3.6: One-dimensional Sine model. In (a), black points stand for observation that are not censored; red points are observations that are censored, and the green points are the counterpart of the red points, that is, they are the latent values of those red points if they were not censored.

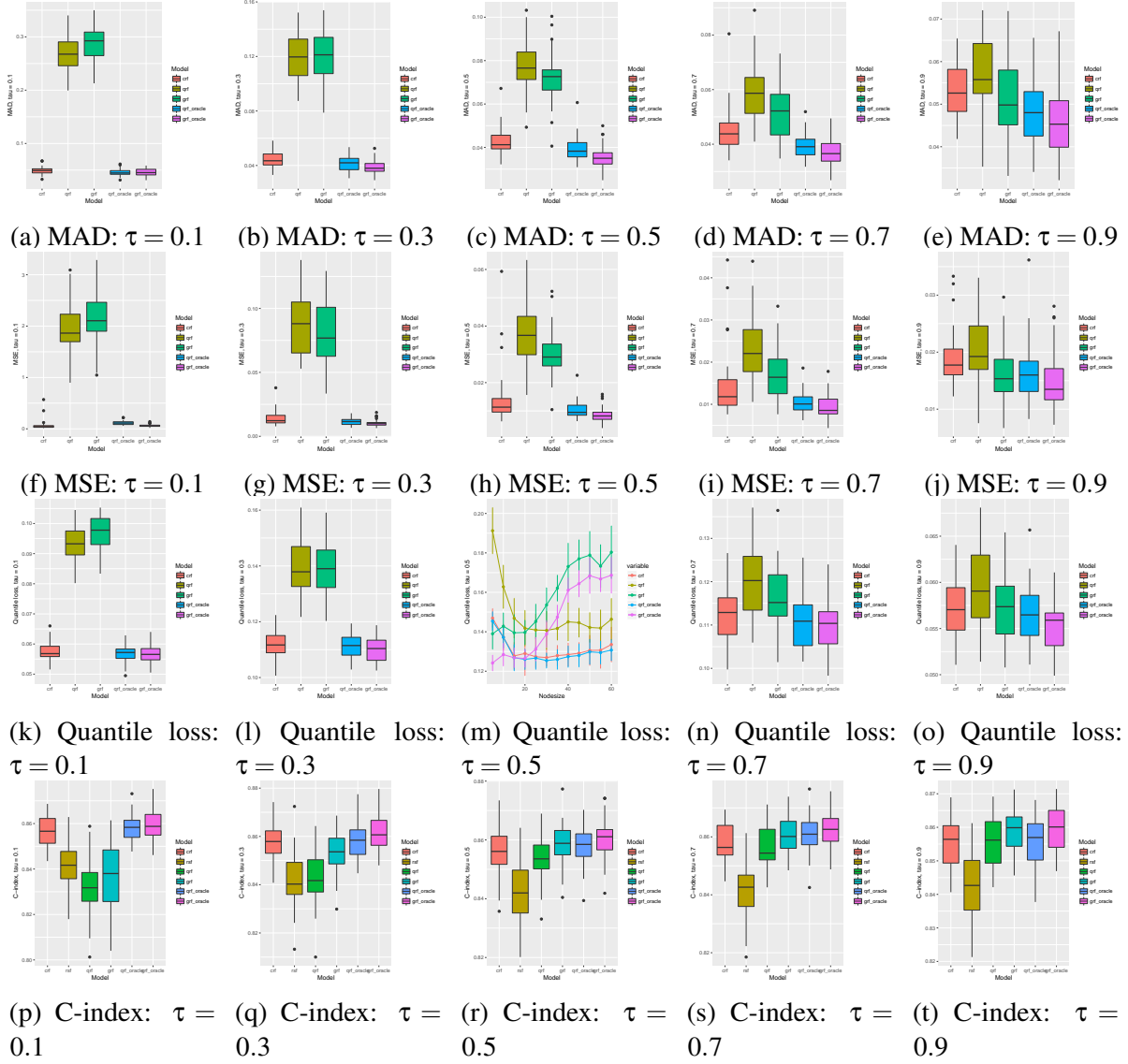


Figure 3.7: Sine 1D box plots.

3.4.5 Multi-dimensional AFT model results

In this section, we test our algorithm on a multi-dimensional AFT model

$$\log(T) = X^T \beta + \varepsilon,$$

where $\beta = (0.1, 0.2, 0.3, 0.4, 0.5)$, $X_{,j} \sim \text{Unif}(0, 2)$, and $\varepsilon \sim \mathcal{N}(0, 0.3^2)$. The censoring variable $C \sim \text{Exp}(\lambda = 0.05)$, and $Y = \min(T, C)$. The censoring level is about 22%. The number of training data is 500 and the number of test points is 300. All the forests consist of 1000 trees. The result is in Figure 3.8. Our model *crf* still outperforms *qrf* and *grf* significantly, and is comparable to *qrf-oracle* and *grf-oracle*.

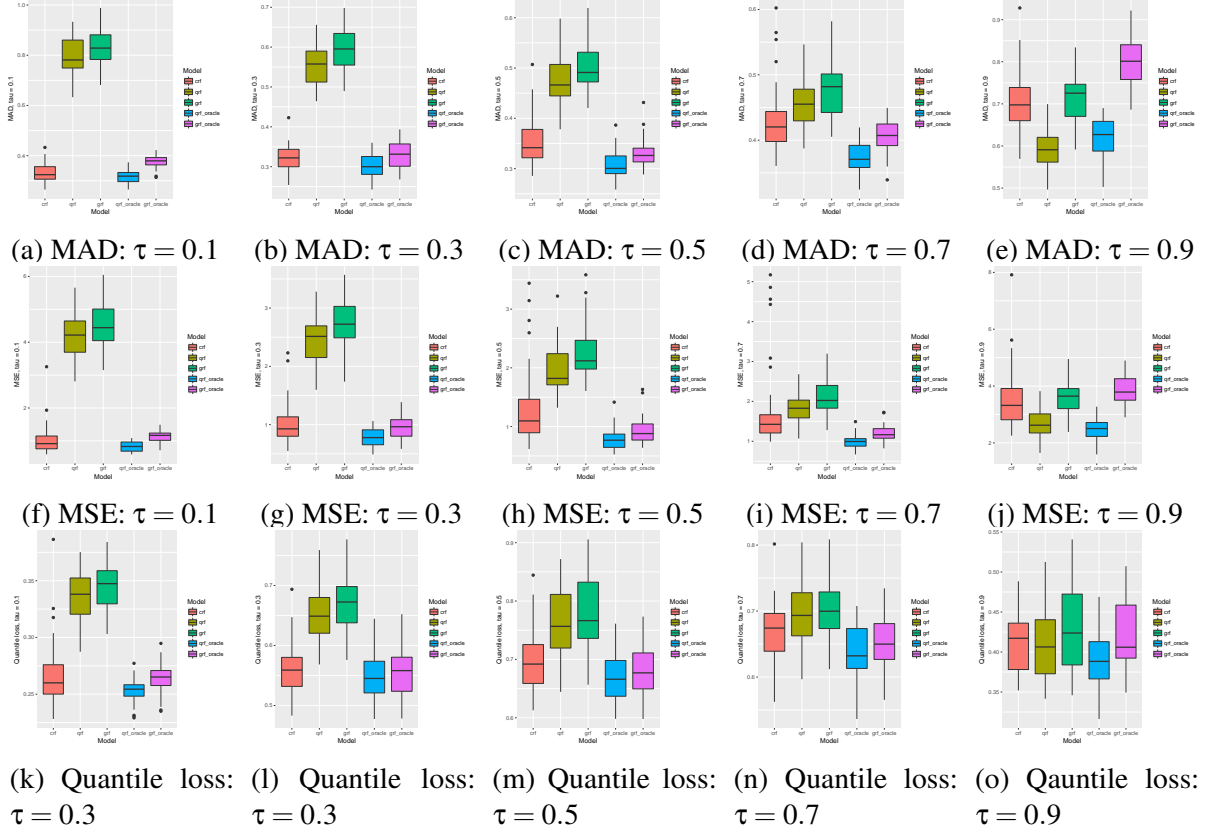


Figure 3.8: AFT multi-dimensional results.

3.4.6 Multi-dimensional complex manifold

In this section, we construct a complex model

$$T = 5 + \frac{1}{5} (\sin(X_{,1}) + \cos(X_{,2}) + X_{,3}^2 + \exp(X_{,4}) + X_{,5}) + \varepsilon,$$

where $X_{.,j}$ stands for j -th dimension of $\mathbf{X} \in \mathbb{R}^5$, and $\varepsilon \sim \mathcal{N}(0, 0.3^2)$. Then we consider two kinds of censoring variables:

- Censoring variable independent of X and T : $C \sim \text{Exp}(\lambda = 0.015)$.
- Censoring variable conditionally independent of T given \mathbf{X} : $C \sim 2 + \sin(X_1) + \cos(X_2) + 0.5X_3^2 + 0.5\exp(X_4) + X_5 + \text{Exp}(\lambda = 0.1)$.

3.4.7 Node size

In this section, we investigate the impact of node size on different methods. The data we use will be generated from the one-dimensional and multi-dimensional AFT and Sine models as defined in the previous sections. We increase the node size from 5 to 60 with step size of 5.

One-dimensional AFT and Sine models The result of sine model is summarized in Figure 3.9. One can see that for both *qrf* and our model, *crf*, the quantile loss will first decrease when node size increases. It attains minimum around node size of 30. However, for *grf*, its quantile loss is almost monotonically increasing, and attains minimum at node size of 5. But both *qrf-oracle* and *grf-oracle* can attain the best quantile loss of about 0.125. And one impressive observation is that our model, *crf*, almost performs the same as *qrf-oracle* for all node sizes. Similar conclusion can be made from the AFT result which is in Figure 3.10.

Multi-dimensional AFT model From Figure 3.11, we observe that *qrf*, *qrf-oracle* and *grf-oracle* all give similar results. The performance of our model *crf* is only slightly worse than *qrf-oracle*, but is even better than *grf-oracle*.

Multi-dimensional complex model The result is summarized in Figure 3.12. The censoring level is about 25%. From the figure, we observe that the behavior of *crf* is still only slightly worse than *qrf-oracle*. In this experiment, *grf-oracle* behaves the best. All of *crf*,

qrf-oracle and *grf-oracle* are significantly better than the biased models *qrf* and *grf*. When $\tau = 0.7$, *grf* behaves slightly better than *qrf-oracle* when node size is small. The reason is that the conditional quantiles of Y and T are closer when τ is larger, and *grf* is more stable and smooth on the data in this experiment. But we still observe that the performance of *crf* and *qrf-oracle* are very close.

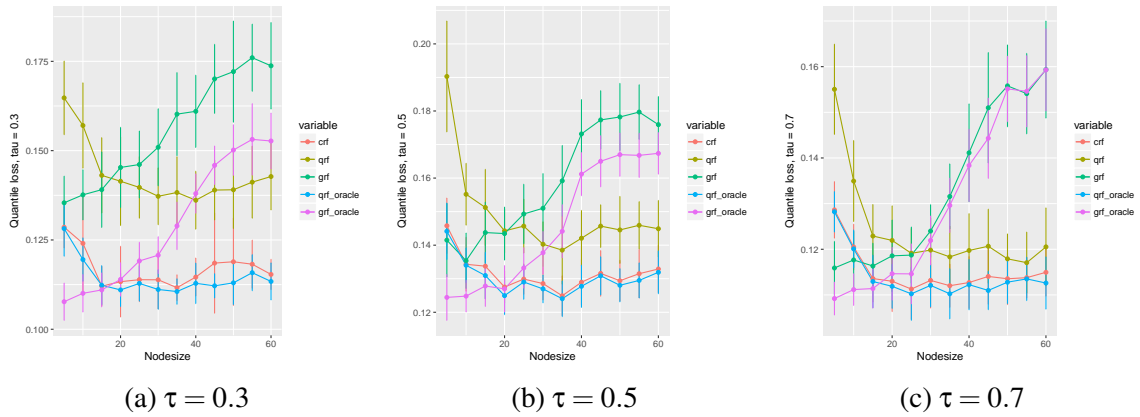


Figure 3.9: Quantile losses of Sine model with different node sizes.

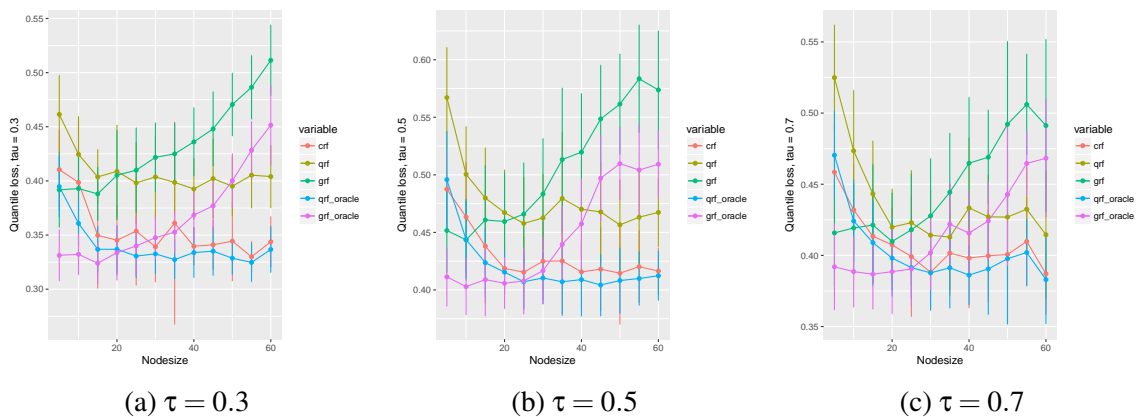


Figure 3.10: Quantile losses of 1D AFT model with different node sizes.

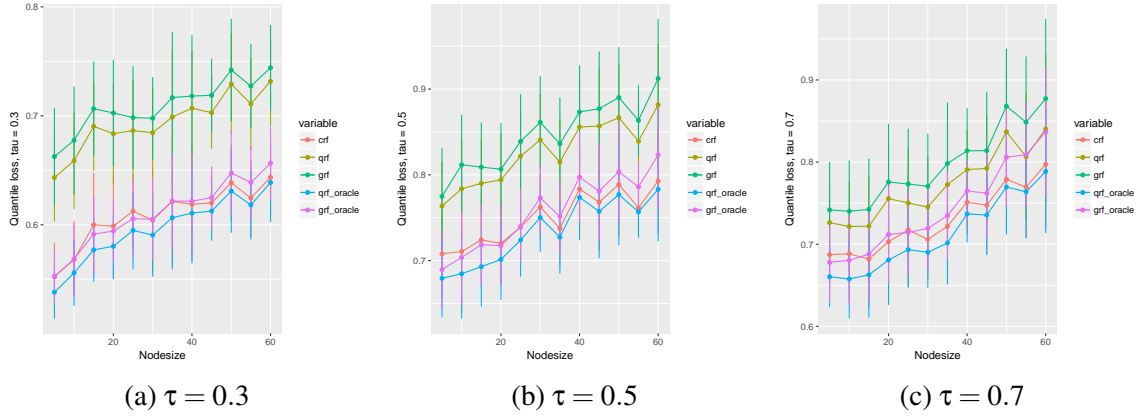


Figure 3.11: Quantile losses of multi-dimensional AFT model with different node sizes.

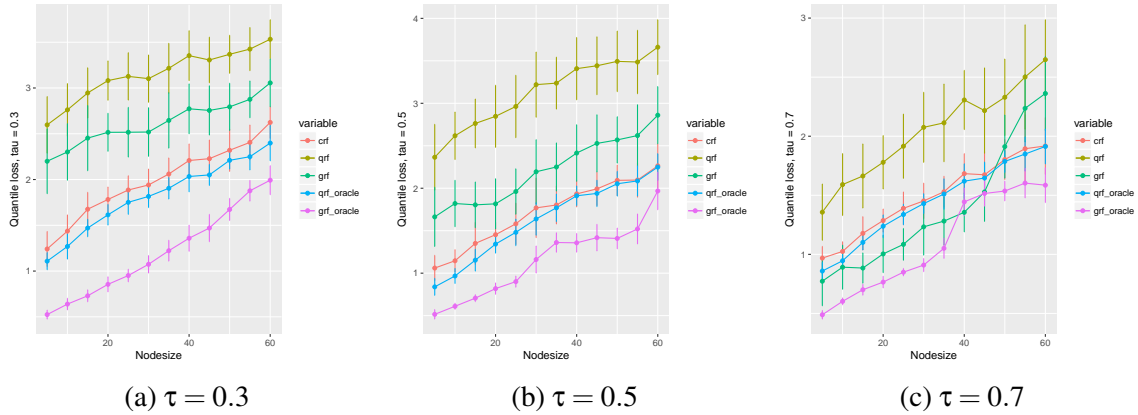


Figure 3.12: Quantile losses of multi-dimensional complex model with different node sizes.

3.5 Real Data

In this section, we compare our censored forest (crf) with quantile random forest (qrf) [Mei06] and generalized forest (grf) [ATW16] on real datasets. In order to evaluate the performances unbiasedly, we manually add censoring to the data. In addition, we apply qrf and grf to the data without censoring and we call the resulted models qrf-oracle and grf-oracle, respectively.

For all these methods, bagged versions of the training data are used for each of the 1000 trees. We use 5-fold cross validation to select the best node sizes for different methods. For all the other parameters, we keep the default settings.

Datasets We use datasets *BostonHousing*, *Ozone* from the R packages *mlbench* and *alr3*. For all the datasets, we sample censoring variables from Exponential distributions with λ set so that the censoring level is roughly 20%. For *BostonHousing* dataset, we set $\lambda = 0.01$. For *Ozone*, $\lambda = 0.025$. For *Abalone* dataset, we random sample 1000 observations and take the log-transformation of the response variable *rings*. We then set $\lambda = 0.10$.

Evaluation For each dataset, we train our model on bootstrapped version of the data, and test the performance on out-of-bag observations. This process is repeated for 40 times, and we calculate the mean and standard deviation of the prediction errors. In our context, the prediction error is measured by the τ -th quantile loss for τ -th quantile estimation. The results are illustrated in Figure 3.13.

On all data sets, our proposed method behaves better than quantile forest and generalized forest in terms of quantile losses. Especially when $\tau = 0.1, 0.3$ or 0.5 , the performance of our method is significantly better than *qrf* and *grf*, and is even comparable to that of oracle *qrf* and *grf*. It agrees with our observation in the one-dimensional example (Figure 3.2 and 3.3). While estimating larger quantiles, the true τ -th quantile of T_i and Y_i are close, and hence the performance of all five models are similar. But when τ is small, the τ -th quantile of T_i and Y_i are different because of the censoring, and in this case, our model has superior advantage and find the true quantiles of T_i almost as good as the oracle methods.

Nodesize For each dataset, we train different models using different nodesizes and compare the performance. For each node size, we bootstrap the data and repeat the experiments for 20 times, and we calculate the mean and standard deviation of the quantile predictions for quantiles $\tau = 0.3, 0.5$, and 0.7 . The result is in Figure 3.14. We observe that our method, *crf*, is uniformly better than *qrf* and *grf*, proving that *crf* is able to correct the bias introduced by censoring. Moreover, the quantile loss of *crf* is always competitive to that of *qrf-oracle* and *grf-oracle*, and is actually always better than *grf-oracle*, only slightly worse than *qrf-oracle*.

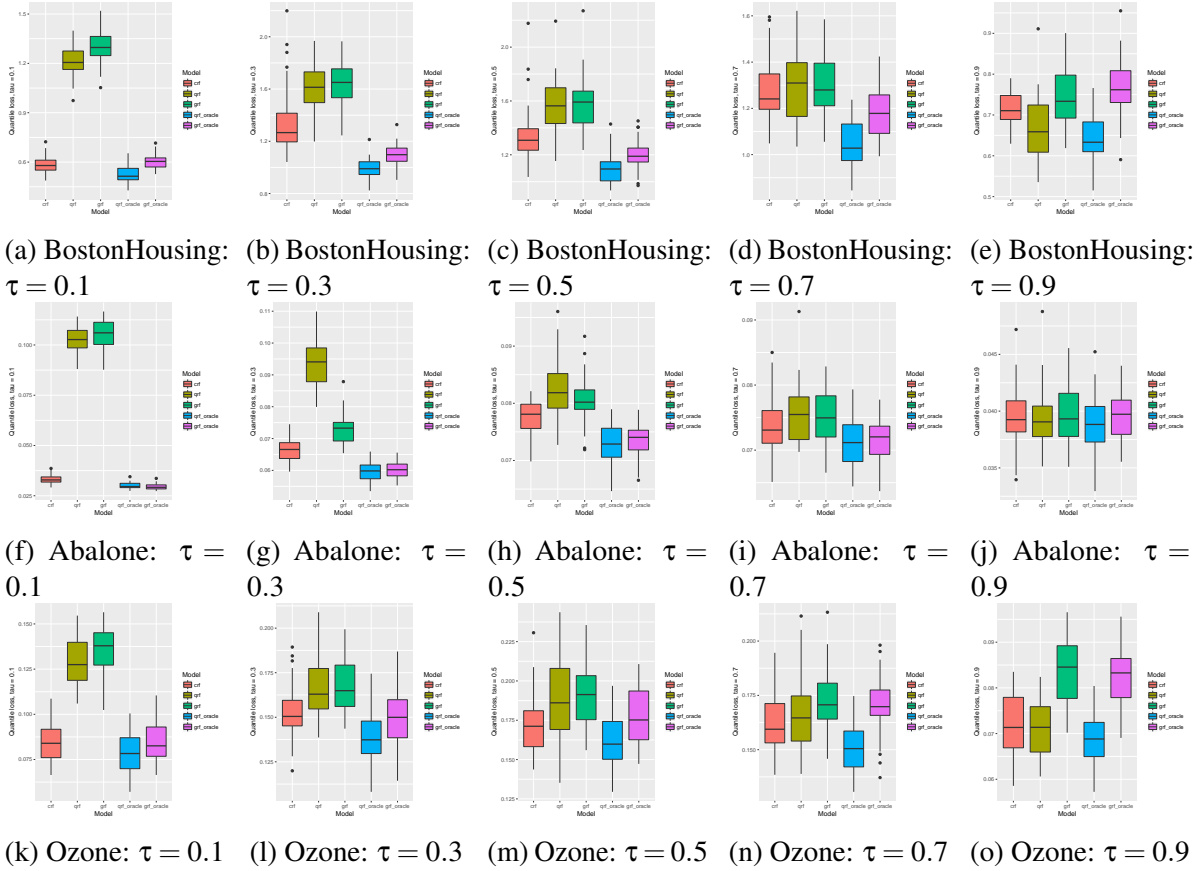


Figure 3.13: Quantile losses on real data.

3.5.1 Prediction Intervals

All the forest methods can be used to get 95% prediction intervals by predicting the 0.025 and 0.975 quantiles. Then for any location $x \in \mathcal{X}$, a straightforward confidence interval will be $[Q(x; 0.025), Q(x; 0.975)]$. The result is illustrated in Figure 3.16. For each data set, we bootstrap the data and calculate the 0.025 and 0.975 quantile for the out of bag points. Then for each node size, we repeat this process for 20 times and calculate the average coverage rate of the confidence intervals.

We observe that in all of the cases, our method *crf* and *qrf-oracle* give the coverage closest to 95%. As can be seen from Figure 3.14, both *qrf* and *grf* perform much worse on predicting lower quantiles. They tend to under-estimate the lower quantiles and hence make the confidence

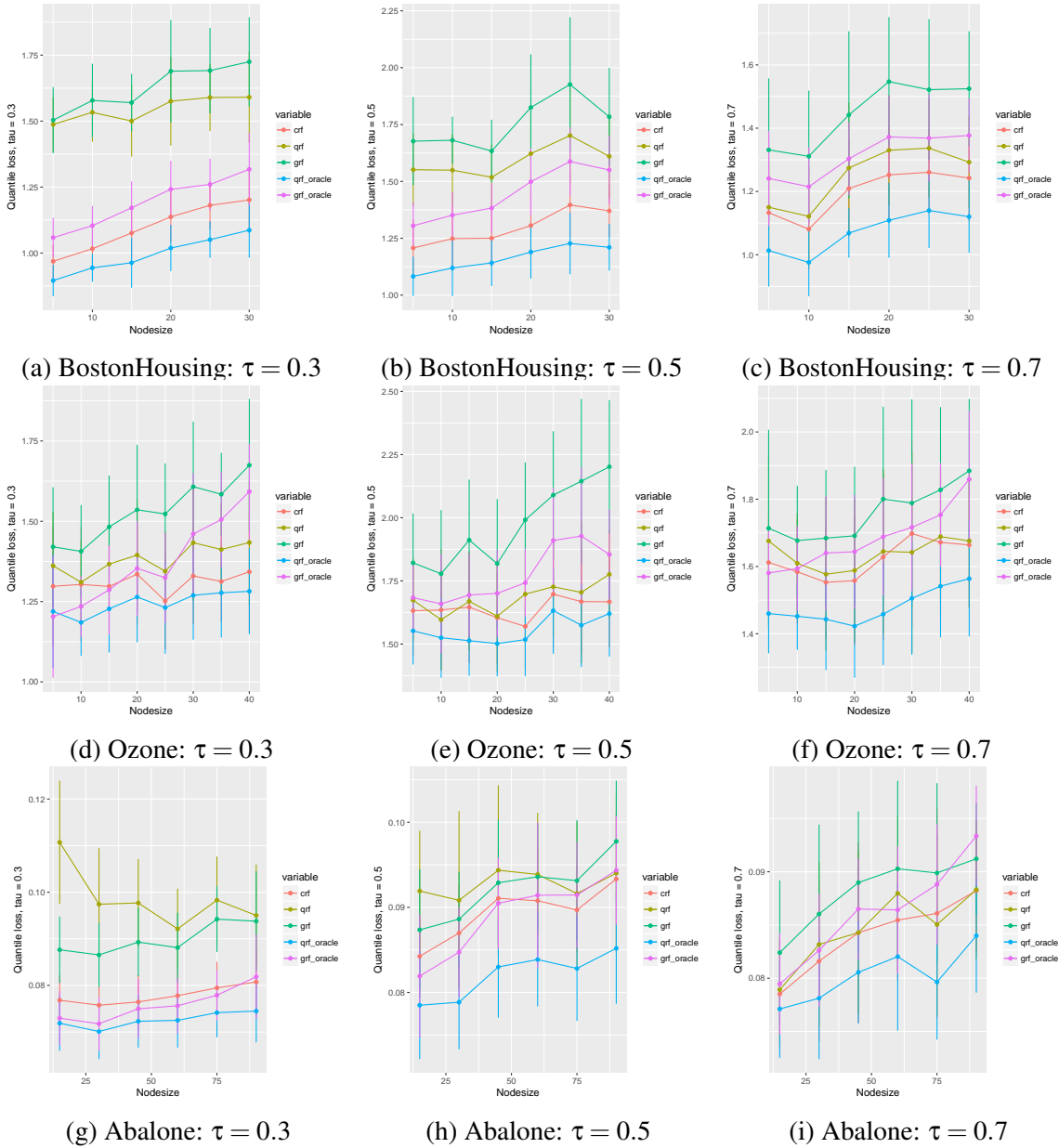


Figure 3.14: Quantile losses of different node sizes on real data.

intervals much wider than the true ones.

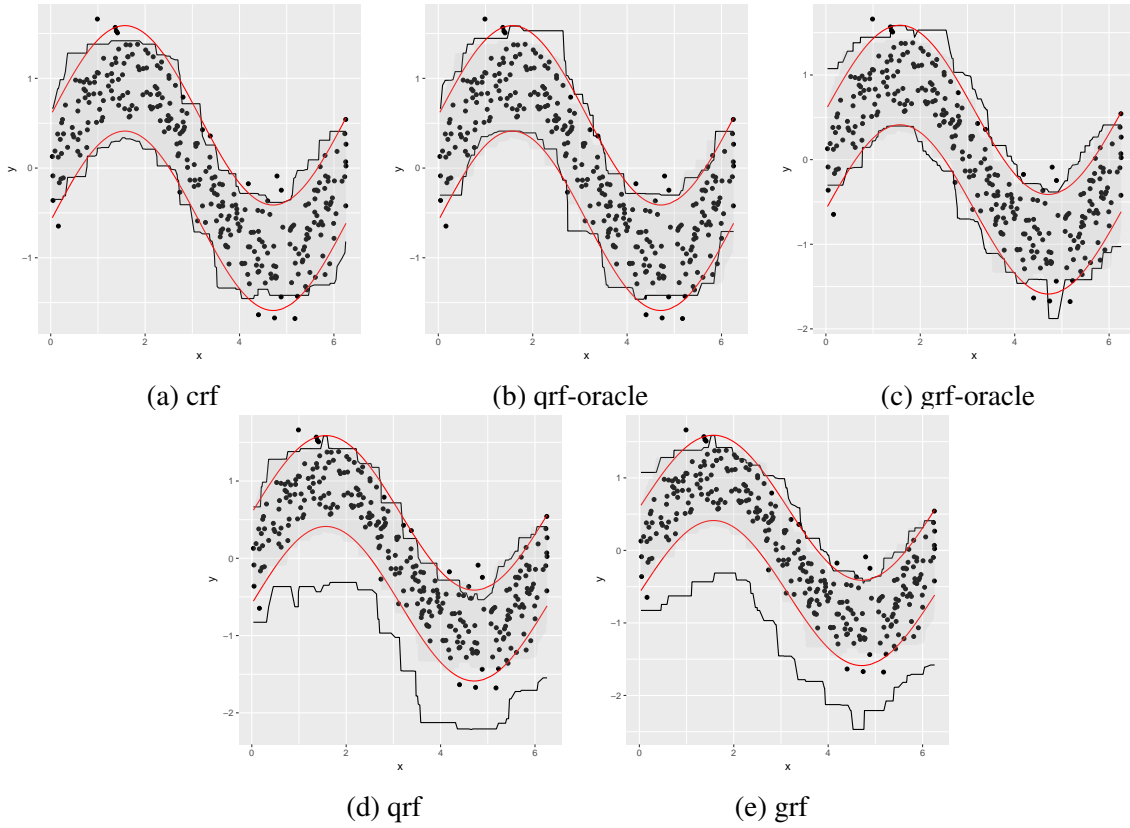


Figure 3.15: Prediction intervals of Sine.

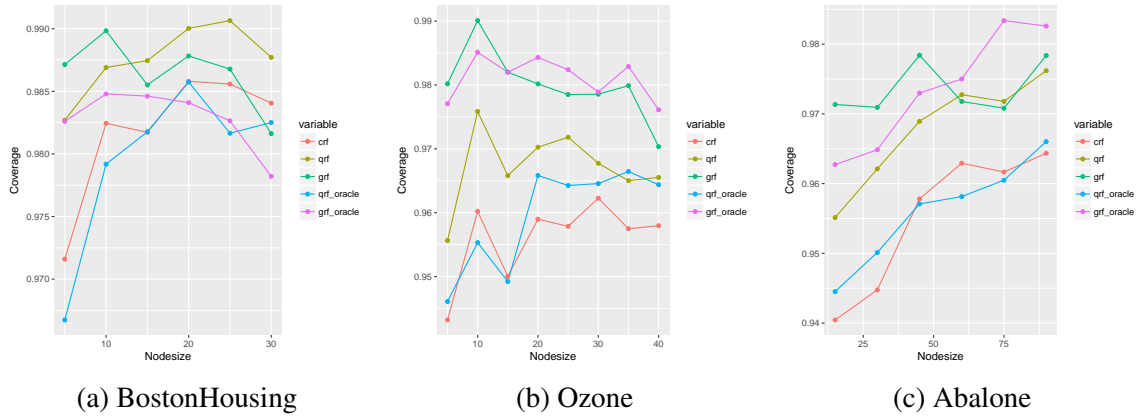


Figure 3.16: Confidence intervals with different node sizes on real data.

3.6 Acknowledgement

Chapter 3, in full, is currently being prepared for submission for publication of the material. Li, Hanbo; Bradic, Jelena. Forest-type quantile regression for randomly censored data.

The dissertation/thesis author is the primary investigator and author of this material.

Chapter 4

Estimation and Inference for High-dimensional Left-censored Quantiles

4.1 Introduction

In this chapter, we present a quantile regression framework for high-dimensional left-censored linear models. Comparing to the forest-type regression framework in Chapter 3, the method introduced below is tailored towards high-dimensional quantile regression and statistical inference on left-censored data. A different approach, namely redistribution of mass, was adopted in the initial estimation here. This creates new challenges in estimation and inference of the problem. In return, the optimization problem can be transformed from a nonconvex optimization involving left-censored data into a modified quantile regression, which then greatly relieves computational burden.

4.1.1 Contributions

We develop methodology for the quantile estimation and inference under high-dimensional and left-censoring settings. In details, the work provides a τ -quantile estimator and confidence

intervals for high-dimensional left-censored regression, for any $\tau \in (0, 1)$, along with the theoretical guarantees. We modify a quantile regression estimation approach for right-censored data to accommodate the left-censored nature of our problem, and further extend the recently developed de-biasing techniques to derive an improved estimator suitable for high-dimensional inference.

4.1.2 Related Work

Quantile regression, as an robust alternative to ordinary linear regression, has received great attention since its introduction in [KBJ78]. The concept has then been taken to settings with heteroskedastic errors [KBJ82] and non-linear regression model [Obe82]. [Pow86] first studied censored quantile regression, where the method was first applied under fixed left-censored data setting, with known censoring levels. Despite of the difficulties present in the censored nature of the data, Powell showed that the proposed natural estimator is consistent and asymptotically normal. However, many works, including [KP96], [Fit97], [BH98] and [FW07], have discussed computational burden due to the nonconvexity nature of the minimization objective function involved in Powell's estimator.

Meanwhile, progress has been made in application of survival analysis. Under right-censored data settings, both [KG01] and [Por03] have studied quantile regression with random right-censored data in details. Moreover, [Por03] proposed a recursively reweighted estimator of the regression quantile process, which generalized the Kaplan-Meier estimating scheme. Based on the redistribution of mass idea of [Efr67], the method in [Por03] recursively updates the weight of censored cases. Similarly, motivated by the same idea, [WW12] proposed a method, such that the weights of the censored observations are estimated in a single step. We extend the idea to high-dimensional left-censored models.

4.1.3 Content

In Section 4.2, the methodology is presented with both procedures for deriving the initial estimator and the de-biased estimator. In Section 4.3, we study the conditions and asymptotic theory of the proposed method. Numerical simulations and a real data application are presented in Section 4.4. Finally, lemmas and their proofs are provided in Section 4.5 and 4.6, and the proofs of theorems are provided in Section 4.7.

4.2 Methodology

We start with the problem setup with model description. Then we lay out the methodology in two parts. In the first subsection, we describe our proposal for initial estimator, and in the second subsection we present the details of bias correction for the initial estimator.

4.2.1 Model Description

We consider the problem in the context of left-censored linear models. Let T_i be an underlying response variable, which is uncensored. We also denote \mathbf{x}_i as our covariates vector of length p . The underlying latent quantile regression model for some quantile $\tau \in (0, 1)$ comes in the form of

$$T_i = \mathbf{x}_i \boldsymbol{\beta}^o(\tau) + \varepsilon_i(\tau), \quad i = 1, \dots, n, \quad (4.1)$$

where $\varepsilon_i(\tau)$ is a random error, whose τ -th quantile conditional on \mathbf{x}_i we assume is at 0. Due to left-censoring, however, we only observe the triplet (\mathbf{x}_i, Y_i, C_i) , where

$$Y_i = \max(T_i, C_i), \quad \text{and let } \delta_i = \mathbb{1}(T_i > C_i), \quad (4.2)$$

and $i = 1, \dots, n$. Together (4.1) and (4.2) specify a left-censored quantile regression model. As C_i is observed, one can always reduce (4.2) to a constant-censored model, also known as Type-I Tobit model, in which the censoring vector is a constant c across i . Our interest lies in obtaining confidence intervals for the quantile coefficient $\beta^o(\tau)$ for various τ , under high-dimensional settings with $p \gg n$. Bearing the high-dimensionality in mind, we denote $S_{\beta^o} = \{j | \beta_j^o \neq 0\}$ as the active set of variables of the coefficients and denote its cardinality by $s_{\beta^o} = |S_{\beta^o}|$.

4.2.2 Initial Estimator

In the case without censoring, quantile regression is carried out with the specific loss function $\rho_\tau(z) = z(\tau - \mathbb{1}\{z < 0\})$, also known as the check function. In the censoring case, however, directly fitting using the quantile loss results in a nonconvex optimization problem. In addition, simply removing the censored observations results in loss of information and bias. With such consideration, we borrow an algorithm from [WW12]. Specifically, we mimicked the "locally weighted censored quantile regression" method, which is based on Efron's redistribution of mass idea. The method assigns different weights on censored data and non-censored data, and avoids discarding all censored data, while maintaining partial information provided by the non-censored ones.

The method redistributes the mass of each censored observation to some point far on left, which is $-\infty$ in the case of left censoring. Note that if $\mathbf{x}_i\beta(\tau) > C_i$ for all \mathbf{x}_i , then the left censoring at C_i has no impact on our estimate of τ -quantile. This observation comes from the fact that the quantile regression estimator is only determined by the signs of residuals, in another word, we only care about the order of the responses.

We now present the initial estimator $\hat{\beta}$, with the justification of the weights following.

$$\hat{\beta} = \arg \min_{\beta} \frac{1}{n} \sum_{i=1}^n [w_i(F_0)\rho_\tau(Y_i - \mathbf{x}_i\beta) + (1 - w_i(F_0))\rho_\tau(Y_i^{-\infty} - \mathbf{x}_i\beta)] + \lambda_n \sum_{j=1}^p |\beta_j|,$$

where $w_i(F_0)$ is defined as following, F_0 being the distribution of T_i ,

$$w_i(F_0) = \begin{cases} 1 & \text{if } \delta_i = 1 \text{ or } F_0(C_i|\mathbf{x}_i) < \tau \\ 1 - \frac{\tau}{F_0(C_i|\mathbf{x}_i)} & \text{if } \delta_i = 0 \text{ and } F_0(C_i|\mathbf{x}_i) > \tau \end{cases}.$$

Notice that the additional penalty term is added, in order to accommodate the high-dimensional setting. To make sense out of the weights, we begin from the objective function of the underlying model under quantile loss,

$$U_n(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^n \rho_{\tau}(T_i - \mathbf{x}_i \boldsymbol{\beta}).$$

Taking the derivative, we have the first order estimating equation

$$D_n(\boldsymbol{\beta}) = -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^{\top} (\tau - \mathbb{1}\{T_i - \mathbf{x}_i \boldsymbol{\beta} < 0\}).$$

The subgradient condition $D_n(\boldsymbol{\beta}) = o_p(1)$ depends only on $\mathbb{1}\{T_i - \mathbf{x}_i \boldsymbol{\beta} < 0\}$ for each \mathbf{x}_i . Now fix any $\boldsymbol{\beta}$, if an observation is uncensored, then $Y_i = T_i$ is observed, and so is $\mathbb{1}\{T_i - \mathbf{x}_i \boldsymbol{\beta} < 0\}$. For censored observations ($Y_i = C_i > T_i$), if $\mathbf{x}_i \boldsymbol{\beta} > C_i$, we immediately know $T_i < \mathbf{x}_i \boldsymbol{\beta}$. The tricky case is when $\mathbf{x}_i \boldsymbol{\beta} < C_i$, we cannot determine the sign of $T_i - \mathbf{x}_i \boldsymbol{\beta}$. Hence, we look at the expectation

$$\mathbb{E}[\mathbb{1}\{T_i - \mathbf{x}_i \boldsymbol{\beta} > 0\} | T_i < C_i] = \frac{\mathbb{P}(\mathbf{x}_i \boldsymbol{\beta} < T_i < C_i)}{\mathbb{P}(T_i < C_i)},$$

where F_0 is the distribution of T_i . When $\boldsymbol{\beta} = \boldsymbol{\beta}^o(\tau)$,

$$\begin{aligned} \mathbb{E}[\mathbb{1}\{T_i - \mathbf{x}_i \boldsymbol{\beta}^o(\tau) > 0\} | T_i < C_i] &= \frac{\mathbb{P}(\mathbf{x}_i \boldsymbol{\beta}^o(\tau) < T_i < C_i)}{\mathbb{P}(T_i < C_i)} \\ &= \frac{F_0(C_i|\mathbf{x}_i) - \tau}{F_0(C_i|\mathbf{x}_i)}. \end{aligned}$$

The observations above motivated us to assign weight $w_i(F_0) = 1$ to the first two scenarios,

when we have uncensored or $F_0(C_i|\mathbf{x}_i) < \tau$ observations. Note that at the location \mathbf{x}_i , even when a data point is censored, if we believe the quantile of interest is above the censoring level, we still assign full weight to that data. Intuitively, we are only interested in estimating in quantile τ . In terms of a specific data point, our only concern is whether it is above or below the quantile line $\mathbf{x}_i\boldsymbol{\beta}^o$. For censored and ambiguous scenarios which we cannot determine the sign of $T_i - \mathbf{x}_i\boldsymbol{\beta}^o(\tau)$, we assign weight $w_i(F_0) = 1 - \frac{\tau}{F_0(C_i|\mathbf{x}_i)}$. By assigning the complimentary weight to any point below, such as $(\mathbf{x}_i, -\infty)$ or $(\mathbf{x}_i, Y_i^{-\infty})$, the quantile fit remains unaffected. Without loss of generality, we assume fixed censoring level $C_i = 0$ for all i .

Finally, using a consistent plug in estimator \hat{F}_n for F_0 , we have the initial estimator as,

Step 0: Initial estimator

$$\begin{aligned} \hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^n [w_i(\hat{F}_n) \rho_{\tau}(Y_i - \mathbf{x}_i\boldsymbol{\beta}) & \quad (4.3) \\ + (1 - w_i(\hat{F}_n)) \rho_{\tau}(Y_i^{-\infty} - \mathbf{x}_i\boldsymbol{\beta})] + \lambda_n \sum_{j=1}^p |\boldsymbol{\beta}_j|. \end{aligned}$$

We delay the discussion of the estimator \hat{F}_n to Condition 8, where we will lay out the requirement on such estimator.

4.2.3 Bias Correction

With our inference objective, the estimator given in (4.3) needs improvement. As we show later, the initial estimator is consistent. However, as other penalized estimators, our initial estimator is also a biased one. Following classical one-step estimation framework, typically an one-step improvement of the following form is considered. With appropriate estimators plugged in as proxies, we have

Step 1: Bias correction

$$\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\Theta}} S_n(\hat{\boldsymbol{\beta}}, \hat{F}_n), \quad (4.4)$$

where the vector \mathbf{S}_n is the score and the matrix $\hat{\Theta}$ is a proxy to the inverse Hessian matrix \mathbf{H}^{-1} . \mathbf{H} is defined as the subgradient of \mathbf{S}_n .

We first define \mathbf{S}_n , and then provide an explanation for the transition between $\hat{\Theta}$ and \mathbf{H}^{-1} .

$$\mathbf{S}_n(\boldsymbol{\beta}, F) := -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top [w_i(F) \psi_\tau(Y_i - \mathbf{x}_i \boldsymbol{\beta}) + (1 - w_i(F)) \psi_\tau(Y_i^{-\infty} - \mathbf{x}_i \boldsymbol{\beta})]$$

with $\psi_\tau(z) = \tau - \mathbb{1}\{z < 0\}$ being the differential of $\rho_\tau(z)$. Note that $Y_i^{-\infty} - \mathbf{x}_i \boldsymbol{\beta} < 0$ due to our choice of $Y_i^{-\infty} = -\infty$. Therefore, we have $\psi_\tau(Y_i^{-\infty} - \mathbf{x}_i \boldsymbol{\beta}) = \tau - 1$ for all i , and hence

$$\begin{aligned} \mathbf{S}_n(\boldsymbol{\beta}, F) &= -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top [w_i(F) \psi_\tau(Y_i - \mathbf{x}_i \boldsymbol{\beta}) + (1 - w_i(F))(\tau - 1)] \\ &= -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top [w_i(F) \mathbb{1}\{Y_i - \mathbf{x}_i \boldsymbol{\beta} \geq 0\} - (1 - \tau)]. \end{aligned} \quad (4.5)$$

Notice that $\mathbf{S}_n(\hat{\boldsymbol{\beta}}, \hat{F}_n)$ depends on both the initial estimator $\hat{\boldsymbol{\beta}}$ and \hat{F}_n . This imposes an additional challenge on the theory, which we address later in Lemma 7. As for (4.3) being a consistent estimator, only consistency of the estimator \hat{F}_n is required. However, for inference a slightly stronger convergence rate requirement on the error of the estimator \hat{F}_n needs to be imposed, which is summarized in Condition 8.

As for the Hessian matrix \mathbf{H} , we observe that the function ψ_τ is not everywhere differentiable. Hence, we propose to consider another candidate for the subgradient of \mathbf{S}_n . We first compute the expectation of the score $\mathbf{S}_n(\boldsymbol{\beta}, F)$, and then compute its gradient. Thus, for the simplicity of notation, the following expectations are taken with respect to T_i given \mathbf{x} .

Proposition 1. *Assuming the true distribution F_0 , we have*

$$\begin{aligned}\mathbb{E}[\mathbf{S}_n(\boldsymbol{\beta}, F_0)] &= -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \left(\tau - \mathbb{P}(Y_i < \mathbf{x}_i \boldsymbol{\beta}) - \tau \mathbb{1}\{F_0(0|\mathbf{x}_i) > \tau\} \mathbb{1}\{\mathbf{x}_i \boldsymbol{\beta} \leq 0\} \right) \\ &= \begin{cases} -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i (\tau - \mathbb{P}(T_i < \mathbf{x}_i \boldsymbol{\beta})) & \text{if } \mathbf{x}_i \boldsymbol{\beta} > 0 \\ -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i (\tau - \tau \mathbb{1}\{F_0(0|\mathbf{x}_i) > \tau\}) & \text{if } \mathbf{x}_i \boldsymbol{\beta} \leq 0 \end{cases} \end{aligned} \quad (4.6)$$

and hence the Hessian

$$\mathbf{H}(\boldsymbol{\beta}) = \frac{\partial}{\partial \boldsymbol{\beta}} \mathbb{E}[\mathbf{S}_n(\boldsymbol{\beta}, F_0)] = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i f_0(\mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i) \mathbb{1}\{\mathbf{x}_i \boldsymbol{\beta} > 0\} \quad (4.7)$$

where f_0 is the density function of T_i .

Remark 5. *Note that $\mathbb{E}[\mathbf{D}_n(\boldsymbol{\beta})] = -n^{-1} \sum_{i=1}^n \mathbf{x}_i \left(\tau - \mathbb{P}(T_i < \mathbf{x}_i \boldsymbol{\beta}) \right)$. Comparing to (4.6), we know when $\mathbf{x}_i \boldsymbol{\beta} > 0$, $\mathbb{E}[\mathbf{S}_n(\boldsymbol{\beta}, F_0)] = \mathbb{E}[\mathbf{D}_n(\boldsymbol{\beta})]$, and hence when $\boldsymbol{\beta} = \boldsymbol{\beta}_o$, $\mathbb{E}[\mathbf{S}_n(\boldsymbol{\beta}_o, F_0)] = 0$ since $\mathbb{E}[\mathbf{D}_n(\boldsymbol{\beta}_o)] = 0$. Furthermore, $\mathbb{1}\{F_0(0|\mathbf{x}_i) > \tau\} = \mathbb{1}\{\mathbf{x}_i \boldsymbol{\beta}_o \leq 0\}$, if F_0 is strictly increasing, and hence $\mathbb{E}[\mathbf{S}_n(\boldsymbol{\beta}_o, F_0)] = 0$ when $\mathbf{x}_i \boldsymbol{\beta}_o \leq 0$ as well. In summary, at the truth $\boldsymbol{\beta}_o$, the expectation of our score estimator $\mathbb{E}[\mathbf{S}_n(\boldsymbol{\beta}_o, F_0)]$ is indeed zero.*

Note that the matrix $\mathbf{H}(\boldsymbol{\beta})$ is not invertible for general $\boldsymbol{\beta}$ when the number of parameters p exceeds the number of observations n . In fact, with a little abuse of notation, we only assume the existence of \mathbf{H}^{-1} , which is laid out as Condition 14 later in the text (here, the expectation is with respect to \mathbf{x}_i). In the following section, we describe the details in obtaining the proxy $\hat{\Theta}$ for \mathbf{H}^{-1} .

4.2.4 Inverse Hessian Estimator: Nodewise Lasso

Our Inverse Hessian estimator is inspired by the nodewise lasso method proposed in [VdGBRD14]. For notation simplicity, we first rewrite (4.7),

$$\mathbf{H}(\boldsymbol{\beta}) = n^{-1} \sum_{i=1}^n \mathbf{u}_i^\top \mathbf{u}_i = n^{-1} \mathbf{x}_\beta^\top \mathbf{x}_\beta,$$

where $\mathbf{u}_i := \mathbf{x}_i \mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} > 0) \sqrt{f_0(\mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i)}$, $\mathbf{x}_\beta = \mathbf{W}_\beta \mathbf{x}$, and \mathbf{W}_β is defined as

$$\mathbf{W}_\beta = \text{diag} \left(\mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} > 0) \sqrt{f_0(\mathbf{x}_i \boldsymbol{\beta} | \mathbf{x}_i)} \right)_{i=1}^n. \quad (4.8)$$

That is, \mathbf{x}_β is a new design matrix with i -th row to be \mathbf{u}_i , which can also be treated as the product of weighted matrix \mathbf{W}_β and \mathbf{x} . Note that for fixed data, $(\mathbf{W}_\beta)_{jj}$ only depends on $\boldsymbol{\beta}$.

Then we carry out nodewise lasso using \mathbf{x}_β . Note that as we use the initial estimator $\hat{\boldsymbol{\beta}}$ as the plug in for \mathbf{x}_β , we also use a consistent estimator \hat{f}_n in place for f_0 in (4.8). The discussion of the estimator \hat{f}_n is delayed later to Condition 9. We have the nodewise lasso scheme as following. For each $j = 1, \dots, p$, define

$$\hat{\boldsymbol{\gamma}}_j := \arg \min_{\boldsymbol{\gamma} \in \mathbb{R}^{p-1}} \left(n^{-1} \|(\mathbf{x}_{\hat{\boldsymbol{\beta}}})_j - (\mathbf{x}_{\hat{\boldsymbol{\beta}}})_{-j} \boldsymbol{\gamma}\|_2^2 + 2\lambda_j \|\boldsymbol{\gamma}\|_1 \right), \quad (4.9)$$

where $(\mathbf{x}_{\hat{\boldsymbol{\beta}}})_{-j}$ is the design submatrix without the j -th column. Note that (4.9) can be solved using standard lasso regression. We further denote the components of $\hat{\boldsymbol{\gamma}}_j \in \mathbb{R}^{p-1}$ as $\{\hat{\boldsymbol{\gamma}}_{j,k} : k = 1, \dots, p, k \neq j\}$. Then define

$$\hat{\mathbf{C}} := \begin{pmatrix} 1 & -\hat{\boldsymbol{\gamma}}_{1,2} & \cdots & -\hat{\boldsymbol{\gamma}}_{1,p} \\ -\hat{\boldsymbol{\gamma}}_{2,1} & 1 & \cdots & -\hat{\boldsymbol{\gamma}}_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ -\hat{\boldsymbol{\gamma}}_{p,1} & -\hat{\boldsymbol{\gamma}}_{p,2} & \cdots & 1 \end{pmatrix}$$

and

$$\hat{D}^2 := \text{diag}(\hat{d}_1^2, \dots, \hat{d}_p^2),$$

where for $j = 1, \dots, p$,

$$\hat{d}_j^2 := n^{-1} \|(X_{\hat{\beta}})_j - (\mathbf{x}_{\hat{\beta}})_{-j} \hat{\gamma}_j\|_2^2 + \lambda_j \|\hat{\gamma}_j\|_1. \quad (4.10)$$

\hat{d}_j^2 serves as an estimate to the noise level of the regression in (4.9). Finally, our proxy $\hat{\Theta}$ is defined as,

$$\hat{\Theta} := \hat{D}^{-2} \hat{C}. \quad (4.11)$$

In addition, we note that using the KKT conditions, we can show

$$\|\mathbf{H}(\hat{\beta}) \hat{\Theta}_j^\top - e_j\|_\infty \leq \lambda_j / \hat{d}_j^2, \quad (4.12)$$

and

$$\left(\mathbf{H}(\hat{\beta}) \hat{\Theta}^\top \right)_{jj} = 1. \quad (4.13)$$

Finally, we propose the novel High-dimensional Left-censored Quantile Regression in Algorithm 6 and 7.

Algorithm 6 High-dimensional Left-censored Quantile Regression

- 1: Initial Estimation
 - 2: Obtain an estimator \hat{F}_n
 - 3: Plug in \hat{F}_n into (4.3) and obtain $\hat{\beta}$
 - 4: One-step Correction
 - 5: Obtain estimator $\hat{\Theta}$, more details in Algorithm 7
 - 6: Plug in initial estimator $\hat{\beta}$ and \hat{F}_n for $\mathcal{S}_n(\hat{\beta}, \hat{F}_n)$ as in (4.5)
 - 7: Obtain the one-step improved estimator $\tilde{\beta}$ as in (4.4)
-

Algorithm 7 Inverse Hessian estimation $\hat{\Theta}$

- 1: Obtain an estimator \hat{f}_n
 - 2: Plug in initial estimator $\hat{\beta}$ and \hat{f}_n into (4.8)
 - 3: **for** $j = 1, \dots, p$ **do**
 - 4: Obtain $\hat{\gamma}_j$ and \hat{d}_j^2 as in (4.9) and (4.10) respectively
 - 5: **end for**
 - 6: Obtain $\hat{\Theta}$ as described in (4.11)
-

4.3 Theoretical Considerations

In what follows, we briefly discuss the preliminary theoretical results, along with the conditions required. In the first subsection, we address the requirements for the distribution and density estimators. Then we move on to conditions for acquiring consistency using the initial estimator. We are inspired by the consistency result of the penalized censored least absolute deviation estimator in [MvdG16]. Finally, we present the derivation of the normality result for the improved one-step estimator, which follows from the sketch of [BG16]. Under the current context, however, extra challenges surface as both score and inverse Hessian depends on distribution and density estimator in addition to the parameter estimator $\hat{\beta}$.

4.3.1 Distribution and Density Estimators

We impose the following condition on the choice of distribution estimator.

Condition 8 (Distribution estimator condition). *The estimator $\hat{F}_n(t|\mathbf{x})$ is a consistent estimator of the conditional distribution of T , $F_0(t|\mathbf{x})$, for all \mathbf{x} . More precisely, for any $t \in \mathbb{R}$,*

$$\sup_{\mathbf{x} \in \mathbb{R}^p} |\hat{F}_n(t|\mathbf{x}) - F_0(t|\mathbf{x})| = O_p(\delta_{\hat{F}}),$$

where $\delta_{\hat{F}} \xrightarrow{p} 0$ as $n \rightarrow \infty$.

Note that the condition essentially only requires \hat{F}_n to be a consistent estimator. We have

selected the classical Kaplan-Meier estimator for analysis later in the paper. Likewise, we also impose a consistency condition on the density estimator \hat{f}_n as following.

Condition 9 (Density condition). *1. The conditional density function $f_0(y|\mathbf{x})$ is a Lipschitz function in y with a uniform Lipschitz constant L for all \mathbf{x} .*

2. There exists $M > m > 0$ such that $m \leq f_0(y|\mathbf{x}) \leq M$ for all y and \mathbf{x} .

3. The conditional density estimator $\hat{f}_n(y|\mathbf{x})$ is a consistent estimator of $f_0(y|\mathbf{x})$. To be precise,

$$\int \int (\hat{f}_n(y|\mathbf{x}) - f_0(y|\mathbf{x}))^2 d\mu(\mathbf{x})dy = o_p(1),$$

where μ is a measure on the support of \mathbf{x} .

4. $\lim_{\epsilon \rightarrow 0^+} \mathbb{P}(|\mathbf{x}\beta^\circ| > \epsilon) = 1$.

The two conditions above are not restrictive in their nature, though distribution and density estimation in high-dimensional settings remains an active research topic. Nevertheless, we refer one to [HY05], [Efr07] and [IL15] for more discussions on the topic.

4.3.2 Consistency of Initial Estimator

In the section, we present the consistency analysis for the initial estimator. For notational simplicity, throughout this section, \mathbf{x} and \mathbf{x}_i are row vectors. Also, we denote $\hat{w} = w(\hat{F})$ and $w^0 = w(F_0)$. We also define the linear function $f_\beta(\mathbf{x}) = \mathbf{x}\beta$, the reweighted loss function $\rho_f(\mathbf{x}, y, w) = w\rho_\tau(y - f_\beta(\mathbf{x})) + (1 - w)\rho_\tau(y^{-\infty} - f_\beta(\mathbf{x}))$, the risk $\mathcal{P}\rho_f = \mathbb{E}\rho_f(\mathbf{x}, y, w^0)$, the empirical risk $\mathcal{P}_n\rho_f = \frac{1}{n} \sum_{i=1}^n \rho_f(\mathbf{x}_i, y_i, w_i^0)$ at F_0 , and the empirical risk $\hat{\mathcal{P}}_n\rho_f = \frac{1}{n} \sum_{i=1}^n \rho_f(\mathbf{x}_i, y_i, \hat{w}_i)$ at \hat{F} . Then we define f^* as a linear functional such that for all \mathbf{x} ,

$$f^*(\mathbf{x}) = \arg \min_a \mathbb{E} [w^0 \rho_\tau(y - a) + (1 - w^0) \rho_\tau(y^{-\infty} - a) | \mathbf{x}].$$

In order for f^* to be uniquely defined, we need the following censoring condition Condition 10. To see the necessity of this condition, let β^o be the true parameter. By the first order property, $\mathbb{E}[w^0 \psi_\tau(y - a) + (1 - w^0)(\tau - 1) | \mathbf{x}] = 0$. Hence, for all \mathbf{x} ,

$$\mathbb{E}[w^0 \mathbb{1}(y > a) | \mathbf{x}] = 1 - \tau. \quad (4.14)$$

By the definition of weight w^0 , if $F_0(0 | \mathbf{x}) < \tau$, (4.14) means $F_0(a | \mathbf{x}) = \tau$, and hence, $f_0(\mathbf{x}) = f_{\beta^o}(\mathbf{x})$. But if $F_0(0 | \mathbf{x}) > \tau$, then any $a < 0$ is a solution to (4.14). However, we require (4.14) to hold for every \mathbf{x} . So as long as not for all \mathbf{x} , $F_0(0 | \mathbf{x}) > \tau$, then because of the linearity of f^* , there exists a unique solution.

Condition 10 (Censoring condition). *Let μ be measure on \mathcal{X} . There exists a set $E \subset \mathcal{X}$ such that $\mu(E) > 0$ and $F_0(0 | \mathbf{x}) < \tau$ for all $\mathbf{x} \in E$. Furthermore, at the censoring level 0, there exists a constant $0 < M_0 < \tau$ such that $F_0(0 | \mathbf{x}) \geq M_0$ for all \mathbf{x} .*

Some additional conditions also need to be imposed.

Condition 11 (Error condition). *The conditional error distribution function $v_0(t | \mathbf{x})$ is continuously differentiable for all \mathbf{x} , and the first derivative $\dot{v}_0(t | \mathbf{x})$ satisfies Lipschitz condition with constant L uniformly for all \mathbf{x} , and is bounded from above and below. Furthermore, $\dot{v}_0(0 | \mathbf{x}) > 0$ and $\int_0^\varepsilon (\varepsilon - t) dv_0(t | \mathbf{x}) > 0$ for all $\varepsilon > 0$ and \mathbf{x} .*

The above condition is our only limitation on the error distribution. Even though we require bounded first derivative for the error density, which excludes densities with unbounded first moment, the condition still allows for a class of distributions much larger than the Gaussian. Next, we have a condition on the design. First, we denote $\gamma_j := \arg \min_\gamma \mathbb{E} \|\mathbf{X}_j - \mathbf{X}_{-j} \gamma\|_n^2$, and then $\mathbf{X}_{-j} \gamma_j$ is the projection of \mathbf{X}_j into \mathbf{X}_{-j} under the inner product $\langle \mathbf{X}_i, \mathbf{X}_j \rangle = \mathbb{E} \mathbf{X}_i^\top \mathbf{X}_j / n$.

Condition 12 (Design matrix condition). *The design matrix \mathbf{X} satisfies $\|\mathbf{X}\|_\infty = \max_{i,j} |X_{i,j}| = O(1)$, that is, every column $\|\mathbf{X}_j\|_\infty = O(1)$. If furthermore, the projection $\mathbf{X}_{-j} \gamma_j$ is also bounded for all j , i.e. $\|\mathbf{X}_{-j} \gamma_j\|_\infty = O(1)$, we say \mathbf{X} is strongly bounded.*

A bounded condition on design matrix entries \mathbf{X}_{ij} is not uncommon in high-dimensional settings [VdGBRD14]. In fact, in many cases, if \mathbf{X} follows an unbounded distribution, one can always approximate its distribution with a truncated one. The following is the same compatibility condition introduced for linear models [BRT09], which is standard condition when applying lasso estimators.

Condition 13 (Compatibility condition). *There exists some $\phi_0 > 0$ and all β satisfying $\|(\beta - \beta^o)_{S_{\beta^o}^c}\|_1 \leq 3\|(\beta - \beta^o)_{S_{\beta^o}}\|_1$ it holds that*

$$\|(\beta - \beta^o)_{S_{\beta^o}}\|_1^2 \leq \frac{s_{\beta^o}}{\phi_0^2} (\beta - \beta^o)^T \mathbb{E}[\mathbf{x}^T \mathbf{x}] (\beta - \beta^o).$$

Denoting the excess risk as $\mathcal{E}(f) = \mathcal{P}\rho_f - \mathcal{P}\rho_{f_0}$, and the sum of squares norm as $\|f\|^2 = \mathbb{E}f^2(\mathbf{x})$, in the linear case, $\|f_{\beta}\|^2 = \mathbb{E}f_{\beta}^2(\mathbf{x}) = \beta^T \mathbb{E}[\mathbf{x}^T \mathbf{x}] \beta$, we are now ready to present the consistency result.

Theorem 10. *Under Conditions 10 - 13 and define*

$$\lambda(t) = 4K_X \sqrt{\frac{2\log(2p)}{n}} + K_X \sqrt{\frac{32t}{n}}.$$

Then for $\lambda \geq 4\lambda(t)$ with $t = 2\log(p)$ and some constant C , with probability at least $1 - \log_2(8np^2)/p^2$,

$$\|\hat{\beta} - \beta^o\|_1 \leq \frac{6C\lambda s_{\beta^o}}{\phi_0^2}, \quad (4.15)$$

$$(\hat{\beta} - \beta^o)^T \mathbb{E}[\mathbf{x}^T \mathbf{x}] (\hat{\beta} - \beta^o) \leq \frac{9C^2 \lambda^2 s_{\beta^o}}{\phi_0^2}. \quad (4.16)$$

In other words, with $\lambda \asymp \sqrt{\log p/n}$, we have $\|\hat{\beta} - \beta^o\|_1 = O_p\left(s_{\beta^o} \sqrt{\frac{\log(p)}{n}}\right)$ and

$$n^{-1} \|\mathbf{X}(\hat{\beta} - \beta^o)\|_2^2 = O_p\left(s_{\beta^o} \frac{\log(p)}{n}\right).$$

Corollary 1. *Under the assumption $s_{\beta^o} = o\left(\sqrt{n/\log(p)}\right)$, we have consistency for the initial estimator $\hat{\beta}$.*

4.3.3 Asymptotic Normality of One-step Penalized Estimator

This section entails the delicate details of obtaining the asymptotic normality of the improved one-step estimator, with imposed conditions as well as the preliminary lemmas. We start the analysis with the following decomposition of (4.4),

$$\begin{aligned} \sqrt{n}(\tilde{\beta}_j - \beta_j^o) &= \underbrace{\sqrt{n}(\hat{\beta}_j - \beta_j^o)}_{\text{I}} - \underbrace{\sqrt{n}(\hat{\Theta}_j \mathbf{S}_n(\beta^o, F_0))}_{\text{N}} \quad (4.17) \\ &\quad - \underbrace{\sqrt{n}[\hat{\Theta}_j(\mathbf{S}_n(\hat{\beta}, F_0) - \mathbb{E}\mathbf{S}_n(\hat{\beta}, F_0)) - \hat{\Theta}_j(\mathbf{S}_n(\beta^o, F_0) - \mathbb{E}\mathbf{S}_n(\beta^o, F_0))]}_{\text{II}}, \\ &\quad - \underbrace{\sqrt{n}\hat{\Theta}_j(\mathbf{S}_n(\hat{\beta}, \hat{F}_n) - \mathbf{S}_n(\hat{\beta}, F_0))}_{\text{III}} - \underbrace{\sqrt{n}(\hat{\Theta}_j(\mathbb{E}\mathbf{S}_n(\hat{\beta}, F_0) - \mathbb{E}\mathbf{S}_n(\beta^o, F_0))}_{\Delta} \end{aligned}$$

where $\hat{\Theta}_j$ denotes the j -th row of $\hat{\Theta}$. With the help of this decomposition, our aim is to show that part (N) converges to a Normal distribution, while the other terms converge to zero at a faster rate. In order to characterize and bound each individual term, we have lemmas for results leading up to Theorem 11 below. However, for the purpose of presentation, we defer the lemmas to Section 4.5.

Finally, we introduce the last condition we impose. One may also refer to this condition as the restrictive eigenvalue assumption, which requires the population Hessian to be at least invertible. We note that even in linear models without censoring, this is an indispensable condition.

Condition 14. *The smallest eigenvalue Λ_{\min} of $\mathbb{E}\left[X_{\beta^o}^T X_{\beta^o}/n\right]$ is strictly positive and $1/\Lambda_{\min} =$*

$O(1)$.

We are now ready to present the main result.

Theorem 11. *Under Conditions 8 - 14, with $\lambda \asymp \sqrt{\log p/n}$ and $\lambda_j \asymp \sqrt{\log p/n}$, and define $s_j := \left\| \Theta_j^0 \right\|_0 = \left| \{k \neq j : \Theta_{j,k}^0 \neq 0\} \right|$, assuming $Ks_{\beta^o}^2 \log p/n \vee s_{\beta^o}^{1/2} s_j^{1/2} (\log p/n)^{1/4} \vee K \|\hat{\Theta}_j - \Theta_j^0\|_1 = o(1)$, where $K = \sqrt{s_j}$ and in the strongly bounded case, $K = 1$. Let $I_n = (\tilde{\beta}_j - a_n, \tilde{\beta}_j + a_n)$ $a_n = z_\alpha \sqrt{\hat{\Theta}_j \mathbf{Om} \hat{\Theta}_j^\top / n}$, where*

$$\mathbf{Om} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i (\hat{\phi}_i + \hat{\psi}_i)^2,$$

$\hat{\psi}_i := - [w_i(\hat{F}_n) \mathbb{1}\{Y_i - \mathbf{x}_i \hat{\beta} \geq 0\} - (1 - \tau)]$ and

$$\hat{\phi}_i := \tau \mathbb{1}(\mathbf{x}_i \hat{\beta} \leq 0) \mathbb{1}(Y_i = 0) \frac{\mathbb{1}(\hat{F}_n > \tau)}{\hat{F}_n^2} \sum_{\substack{l=1 \\ l \neq i}}^n B_{nl}(\mathbf{x}_i) \left(1 - \frac{\mathbb{1}(Y_l = 0)}{\hat{F}_n} \right).$$

The distribution estimator \hat{F}_n is chosen to be the classical Kaplan-Meier estimator,

$$\hat{F}_n(t|\mathbf{x}) = \prod_{j=1}^n \left(1 - \frac{1}{\sum_{k=1}^n \mathbb{1}(Y_k \leq Y_j)} \right)^{\eta_j(t)}, \quad (4.18)$$

where $\eta_j(t) = \mathbb{1}(Y_j > t, \delta_j = 1)$. For $j \in \{1, \dots, p\}$, when $n, p \rightarrow \infty$, we have

$$\mathbb{P}(\beta_j^o \in I_n) = 1 - 2\alpha.$$

Remark 6. *The quantity s_j quantifies the sparsity nature of the underlying precision matrix Θ^0 , which we aim to estimate with $\hat{\Theta}$. This is a standard assumption in high dimensional inference. Essentially, it restricts the column $(\mathbf{X}_{\beta^o})_j$ to be dependent with only s_j number of columns in $(\mathbf{X}_{\beta^o})_{-j}$.*

4.4 Numerical Experiments and Application

In this section, we present the application our proposed method in details, along with simulation results under various settings and an application in real data study.

4.4.1 Further Details of Algorithm 6 and 7

We start with the definition of $Y^{-\infty}$. In practice, we have taken

$$Y^{-\infty} := -1000 \times \|Y\|_{\infty} = -1000 \times \max_i |Y_i|.$$

For the estimator of conditional distribution of T_i , as mentioned earlier, there are options specifically tailored for distribution estimation in high-dimensions, we provide here a possible estimator \hat{F}_n for line 2 in Algorithm 6 based on the ideas of Kaplan-Meier estimator, which is defined as the following.

$$\hat{F}_n(t|\mathbf{x}) = \prod_{j=1}^n \left(1 - \frac{B_{nj}(\mathbf{x})}{\sum_{k=1}^n \mathbb{1}(Y_k \leq Y_j) B_{nk}(\mathbf{x})} \right)^{\eta_j(t)}, \quad (4.19)$$

where $\eta_j(t) = \mathbb{1}(Y_j > t, \delta_j = 1)$. Choosing $B_{nk}(\mathbf{x}) = 1/n$ results in the classical Kaplan-Meier estimator. We also note that the Nadaraya-Watson's type weights for $B_{nk}(\mathbf{x})$ is also a common choice, which is

$$B_{nk}(\mathbf{x}) = \frac{K\left(\frac{\mathbf{x}-\mathbf{x}_k}{h_n}\right)}{\sum_{i=1}^n K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)}, \quad (4.20)$$

where K is a density kernel function, and $h_n \in \mathbb{R}^+$ is the bandwidth converging to zero as $n \rightarrow \infty$. In the simulations, we have opted for the classical Kaplan-Meier estimator for simplicity. In addition, we have the following density estimator for \hat{f}_n in line 1 in Algorithm 7. For a positive

bandwidth sequence \hat{h}_n ,

$$\hat{f}_n = \hat{h}_n^{-1} \sum_{i=1}^n \frac{\mathbb{1}(x_i \hat{\boldsymbol{\beta}} > 0) \mathbb{1}(0 \leq Y_i - \mathbf{x}_i \hat{\boldsymbol{\beta}} \leq \hat{h}_n)}{\sum_{i=1}^n \mathbb{1}(x_i \hat{\boldsymbol{\beta}} > 0)}. \quad (4.21)$$

This estimator is inspired by the estimator for error density at 0 presented in [BG16], which translates to an estimation for density of T_i at $\mathbf{x}_i \boldsymbol{\beta}^o(\tau)$. For the choice of \hat{h}_n , we also follow the adaptive choice of the bandwidth sequence thereof. Let $u_i := y_i - \mathbf{x}_i \hat{\boldsymbol{\beta}}$,

$$\hat{h}_n = c \left\{ s_{\hat{\boldsymbol{\beta}}} \log p/n \right\}^{-1/3} \text{median} \left\{ u_i : u_i > \sqrt{\log p/n}, x_i \hat{\boldsymbol{\beta}} > 0 \right\},$$

for a constant $c > 0$. Here, $s_{\hat{\boldsymbol{\beta}}}$ denotes the size of the estimated set of the non-zero elements of the initial estimator $\hat{\boldsymbol{\beta}}$, i.e., $s_{\hat{\boldsymbol{\beta}}} = \|\hat{\boldsymbol{\beta}}\|_0$.

An additional note is also in place for line 3 of Algorithm 6. Regarding the computation procedure to obtain the initial estimator, we note that this boils down to a weighted quantile regression problem and is readily solvable using linear programming techniques. The penalty parameter λ in (4.3) is chosen by the minimum of K-fold cross validation statistic, $\arg \min_{\lambda} \sum_{k=1}^K \text{CV}_k(\lambda)$, and

$$\text{CV}_k(\lambda) := n_k^{-1} \sum_{i \in F_k} \left[w_i(\hat{F}_n) \rho_{\tau}(Y_i - \mathbf{x}_i \hat{\boldsymbol{\beta}}^k) + (1 - w_i(\hat{F}_n)) \rho_{\tau}(Y_i^{-\infty} - \mathbf{x}_i \hat{\boldsymbol{\beta}}^k) \right], \quad (4.22)$$

where F_k denotes the k -th fold of the n observations, n_k is the number of observations in F_k , and $\hat{\boldsymbol{\beta}}^k$ is the parameter coefficients fitted on F_k^c observations. Likewise, the choice of λ_j in line 4 of Algorithm 7 is chosen in the same way, except in the cross validation statistic, the squared error loss is used instead of the weighted quantile loss in (4.22).

Table 4.1: $\tau = 0.4$ Coverage Probability for High-dimensional Left-censored Quantile Regression (HLQR) with True F_0 and True f_0

| Distribution of the error term | Simulation Setting for $n = 200, p = 300$ | | | |
|--------------------------------|---|----------------|-----------------|----------------|
| | Toeplitz design $\rho = 0.3$ | | Identity design | |
| | Signal Variable | Noise Variable | Signal Variable | Noise Variable |
| Normal | 0,95 | 0,97 | 0,95 | 0,93 |
| Student's | 0,95 | 0,94 | 0,95 | 0,92 |
| Beta | 0,90 | 0,93 | 0,91 | 0,93 |
| Weibull | 0,94 | 0,97 | 0,98 | 0,94 |

4.4.2 Simulation Data

We are now ready to present the simulation results. The size of the model settings are chosen to be of $n = 200$ for the number of observations, and $p = 300$ for the number of parameters. In addition, the sparsity of the underlying true parameter β^o , denoted as s_{β^o} earlier in the text, is set to be 5. We have also selected four different distributions for the error of the model: standard normal, Student's t with 4 degrees of freedom, Beta distribution with parameters $(2, 3)$ and Weibull distribution with parameters $(1, 1)$. The design matrix \mathbf{X} is generated from a multivariate Normal distribution $\mathcal{N}(\mu, \Sigma)$, where μ is chosen to be the zero vector, and the covariance matrix Σ is taken to be the identity matrix or the Toeplitz matrix such that $\Sigma_{ij} = \rho^{|i-j|}$ for $\rho = 0.3$. The two quantiles of interest are chosen to be $\tau = 0.4$ and $\tau = 0.7$. In the case when τ -th quantile of the error is not zero, we subtract off the τ -th quantile of the error distribution from the model. The censoring level c is chosen such that the proportion of the censoring data is set at 10%. We present simulation results for when the true F_0 and f_0 plugged in, and also when we use our proposed rudimentary estimators \hat{F}_n and \hat{f}_n as described earlier in the section.

Table 4.1 and 4.2 summarize the average coverage probabilities of the constructed 95% level confidence intervals for obtaining $\tau = 0.4$ and 0.7 quantile regression estimators under various settings. We report the signal and noise parameters separately, as the coverage of the signal ones are known to be more difficult. In conjunction, we have also included box plots

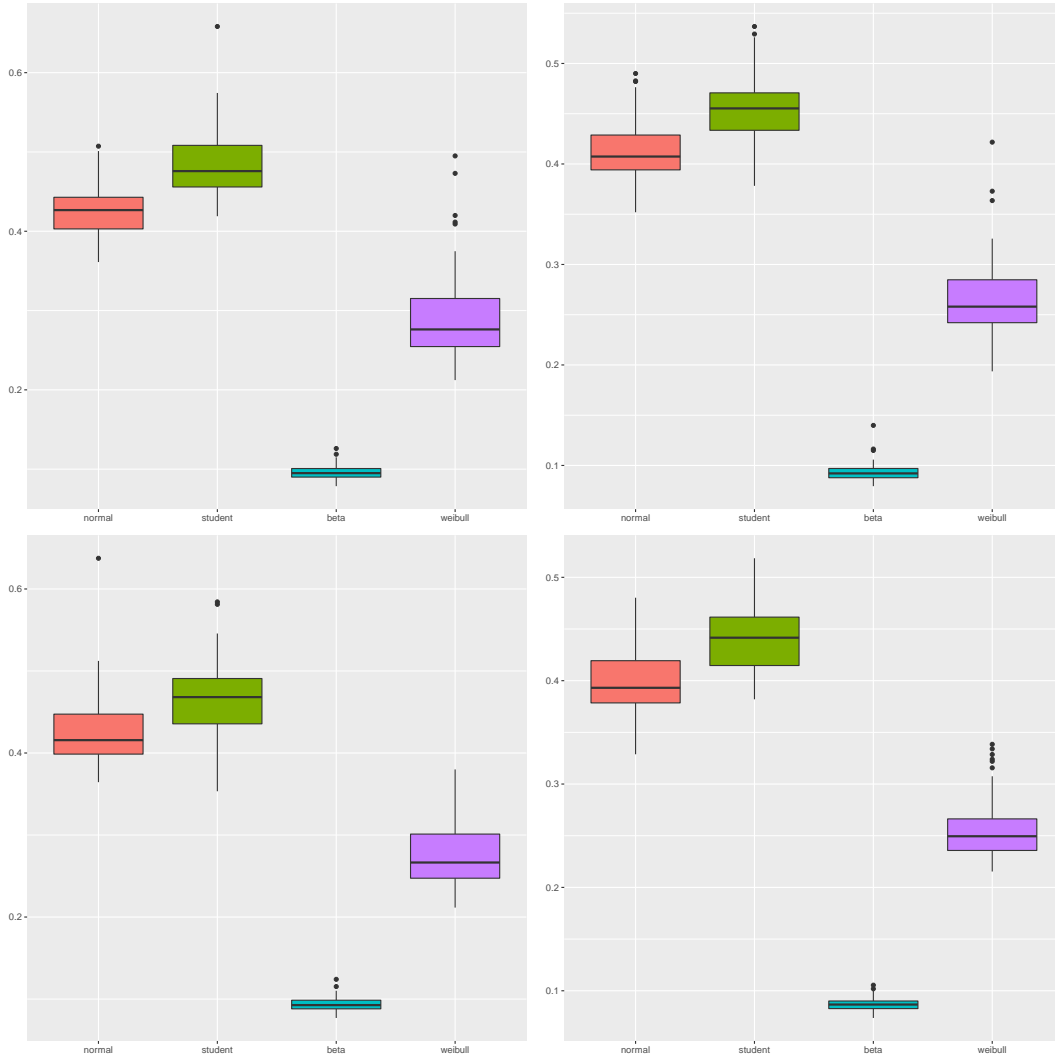


Figure 4.1: $\tau = 0.4$ comparative boxplots of the average interval length (with true F_0 and true f_0). Signal (left) and noise (right) variables, and Toeplitz design with $\rho = 0.3$ (top) and identity design (bottom).

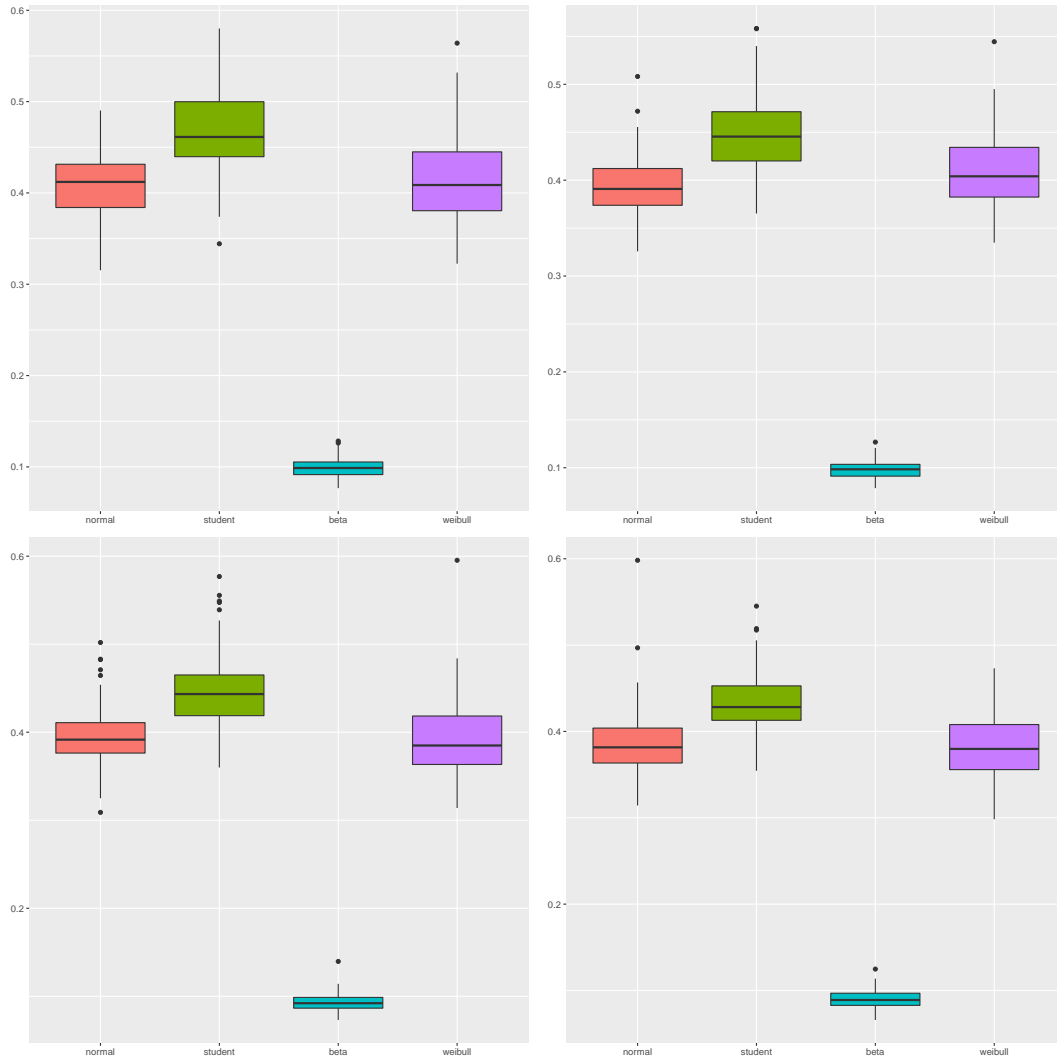


Figure 4.2: $\tau = 0.7$ comparative boxplots of the average interval length (with true F_0 and true f_0). Signal (left) and noise (right) variables, and Toeplitz design with $\rho = 0.3$ (top) and identity design (bottom).

Table 4.2: $\tau = 0.7$ Coverage Probability for High-dimensional Left-censored Quantile Regression (HLQR) with True F_0 and True f_0

| Distribution of the error term | Simulation Setting for $n = 200, p = 300, \tau = 0.7$ | | | |
|--------------------------------|---|----------------|-----------------|----------------|
| | Toeplitz design $\rho = 0.3$ | | Identity design | |
| | Signal Variable | Noise Variable | Signal Variable | Noise Variable |
| Normal | 0,94 | 0,97 | 0,92 | 0,97 |
| Student's | 0,91 | 0,94 | 0,91 | 0,95 |
| Beta | 0,96 | 0,99 | 0,89 | 0,95 |
| Weibull | 0,92 | 0,94 | 0,87 | 0,91 |

of interval widths under these settings (Figure 4.1 and 4.2). From the results of applying our methodology with true F_0 and true f_0 , it is observed that the coverage probabilities are approximately the same and are close to the nominal values. In addition, we noticed that among the four chosen error distributions, our method turns out to be most efficient, in terms of the confidence interval width, when the error distribution is bounded. However, it is observed that our method is sensitive to heavy-tailed distributions, such as the Student's t distribution with degrees of freedom being 4.

The results of plugging in estimators \hat{F}_n and \hat{f}_n are summarized in Table 4.3 and 4.4 for the two quantile settings $\tau = 0.4$ and 0.7 . In terms of coverage probability, we observe similar results as the ones with true F_0 and f_0 , as the probabilities are approximately the same and are close to the nominal values. We notice that the interval widths almost tripled for the cases of error being standard normal and Student's t distribution as seen in Figure 4.3 and 4.4. However, this is not unexpected as we using estimators instead of the true underlying values. With better tailored estimators to the scenario, we believe that the width of the intervals in the two cases can be reduced.

In addition, we have also examined the power of our estimator. Maintaining similar settings as in previous simulations, that is $n = 200$ and $p = 300$, whereas s_{β^0} is also set to be 5. We have our null hypothesis for the coefficients being 1 for the signals and 0 for the noises.

Table 4.3: $\tau = 0.4$ Coverage Probability for High-dimensional Left-censored Quantile Regression (HLQR) with Estimated \hat{F}_n and Estimated \hat{f}_n

| Distribution of the error term | Simulation Setting for $n = 200, p = 300$ | | | |
|--------------------------------|---|----------------|-----------------|----------------|
| | Toeplitz design $\rho = 0.3$ | | Identity design | |
| | Signal Variable | Noise Variable | Signal Variable | Noise Variable |
| Normal | 0,95 | 0,97 | 0,97 | 0,94 |
| Student's | 0,98 | 0,94 | 0,98 | 1,00 |
| Beta | 0,99 | 0,95 | 0,97 | 0,97 |
| Weibull | 0,99 | 0,92 | 0,96 | 0,95 |

Table 4.4: $\tau = 0.7$ Coverage Probability for High-dimensional Left-censored Quantile Regression (HLQR) with Estimated \hat{F}_n and Estimated \hat{f}_n

| Distribution of the error term | Simulation Setting for $n = 200, p = 300, \tau = 0.7$ | | | |
|--------------------------------|---|----------------|-----------------|----------------|
| | Toeplitz design $\rho = 0.3$ | | Identity design | |
| | Signal Variable | Noise Variable | Signal Variable | Noise Variable |
| Normal | 0,89 | 0,99 | 0,96 | 0,97 |
| Student's | 0,93 | 0,93 | 1,00 | 0,96 |
| Beta | 0,96 | 0,97 | 0,91 | 0,96 |
| Weibull | 0,95 | 0,95 | 0,99 | 0,96 |

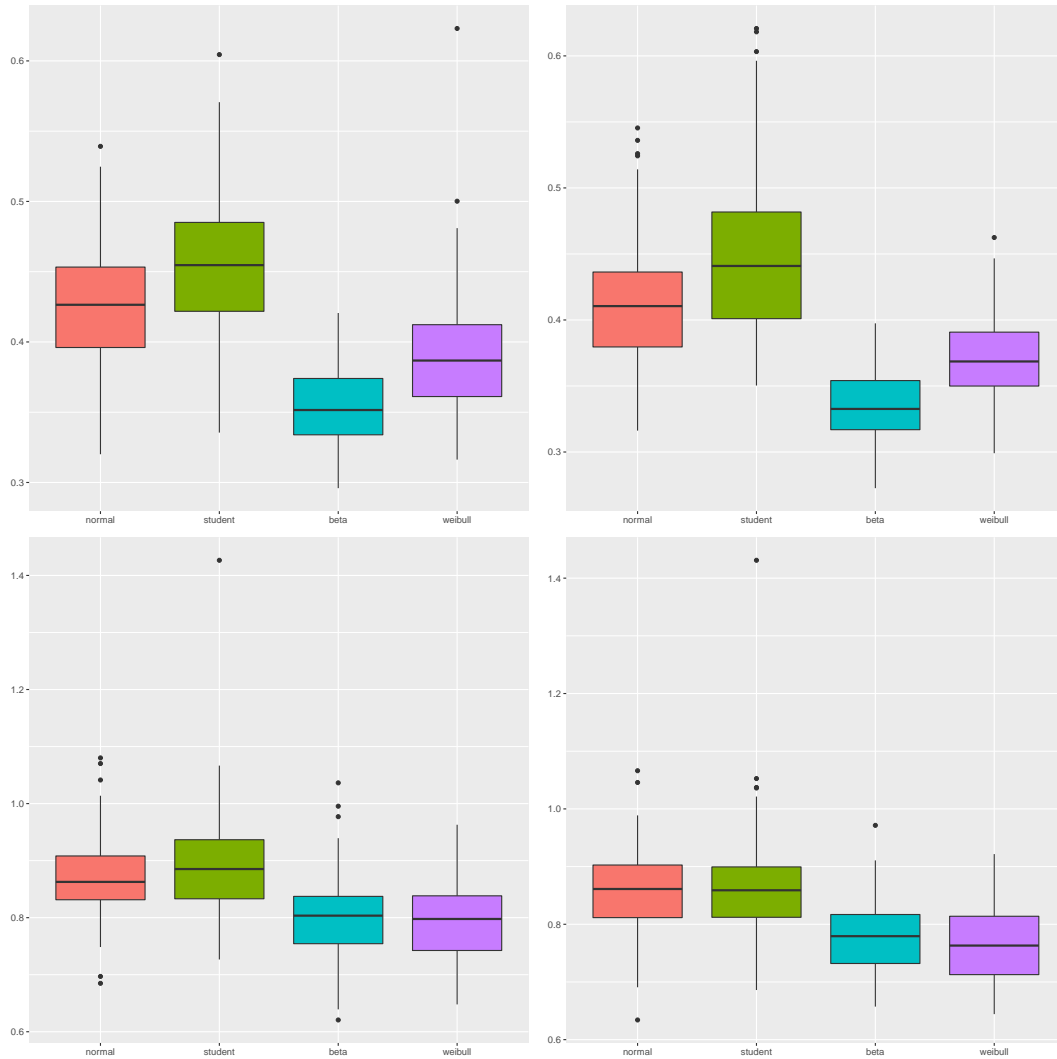


Figure 4.3: $\tau = 0.4$ comparative boxplots of the average interval length (with estimated \hat{F}_n and true \hat{f}_n). Signal (left) and noise (right) variables, and Toeplitz design with $\rho = 0.3$ (top) and identity design (bottom).

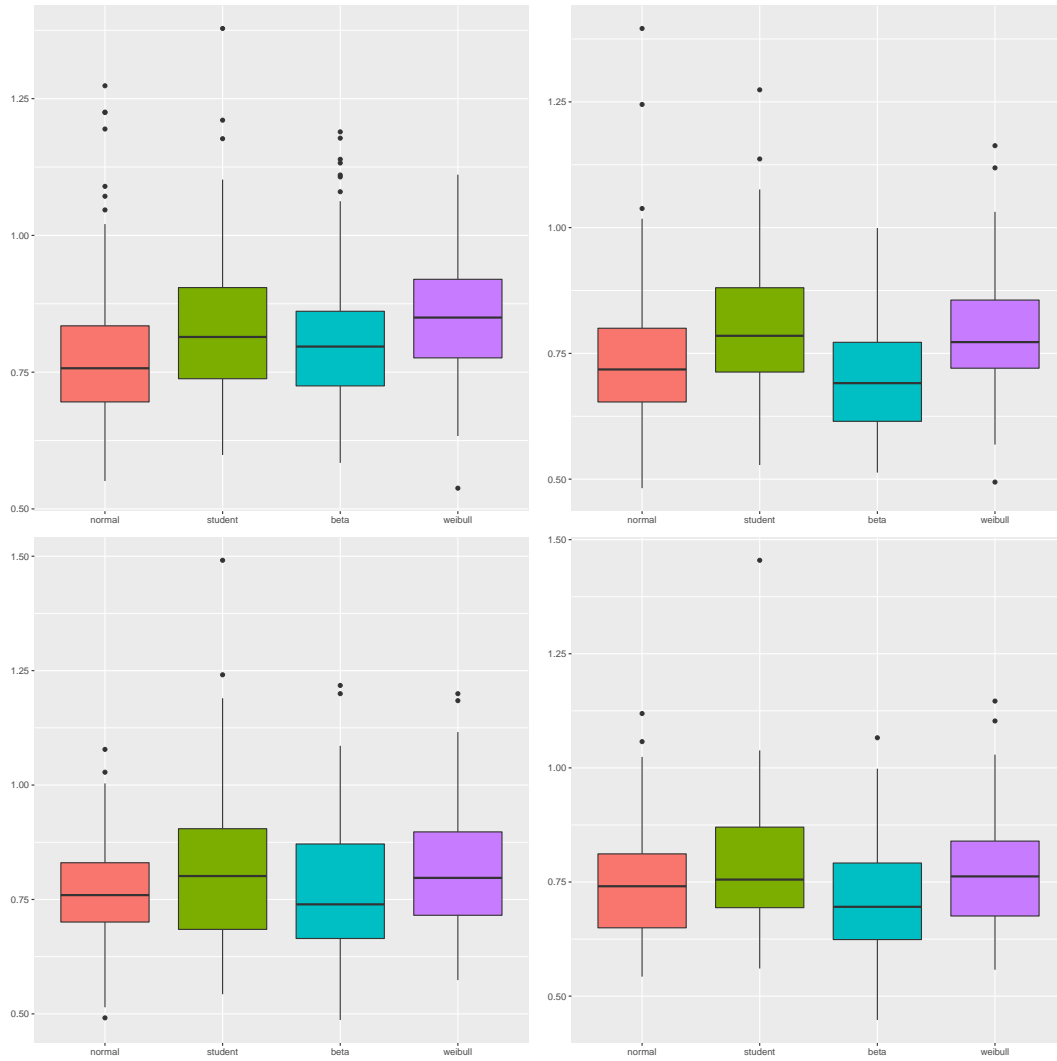


Figure 4.4: $\tau = 0.7$ comparative boxplots of the average interval length (with estimated \hat{F}_n and true \hat{f}_n). Signal (left) and noise (right) variables, and Toeplitz design with $\rho = 0.3$ (top) and identity design (bottom).

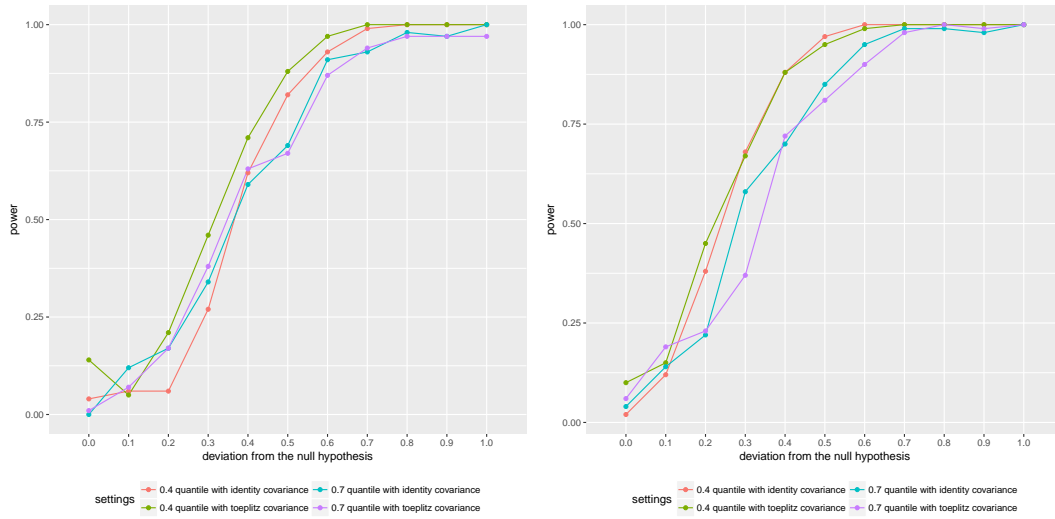


Figure 4.5: Power curve of signal (left) and noise (right) variables under normal errors, $H_0 : \beta_j^o = c$ versus $H_1 : \beta_j^o \neq c$, where the true parameter $\beta_j^o = c + h$. The deviation from the null hypothesis h ranges from 0 to 1.

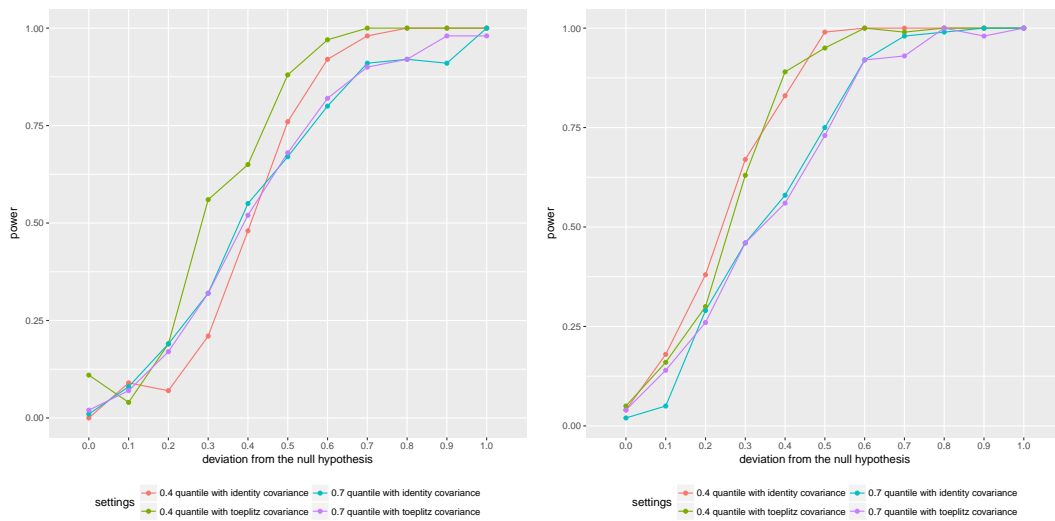


Figure 4.6: Power curve of signal (left) and noise (right) variables under normal errors, $H_0 : \beta_j^o = c$ versus $H_1 : \beta_j^o \neq c$, where the true parameter $\beta_j^o = c + h$. The deviation from the null hypothesis h ranges from 0 to 1.

We test $H_0 : \tilde{\beta}_j = \beta_j^o$ versus $H_1 : \tilde{\beta}_j = \beta_j^o + h$. While keeping the significance level at 0.05, we increase the deviation from the null hypothesis h gradually from 0.1 to 1. We observe that both the signal and noise variables converges to power of 1 quickly for various settings, which testifies the effectiveness of our estimator. The results are summarized in Figure 4.5 and 4.6 below.

4.4.3 Real Data

In this section, we apply our High-dimensional Left-censored Quantile Regression (HLQR) to a microarray dataset of cardiomyopathy in transgenic mice, kindly provided by Professor Mark Segal, who also studied the dataset in [SDC03]. To study human diseases such as chamber dilation and left ventricular conduction delay, a transgenic mouse model of dilated cardiomyopathy was used.

Specifically, [RDK⁺00] proposed to control a G protein-coupled receptor, designated as Ro1, through an inducible expression system. Thirty mice are used for the study, and are divided into four experimental groups. Six transgenic mice expressed Ro1 for two weeks, which did not show symptoms of disease. Nine other transgenic mice expressed Ro1 for eight weeks, and exhibited cardiomyopathy symptoms. The recovery group consists of seven transgenic mice, whose expression of Ro1 was on for eight weeks and off for four weeks. Finally, the control group is made up of non-transgenic mice expressed Ro1 for eight weeks.

The goal is to identify genes involved in the Ro1 expression changes, which may provide new diagnostic markers for cardiomyopathy. To this end, Affymetrix Mu6500 arrays were used for the study, and the response of interest is Ro1, whereas the predictors are 6,319 microarray gene expressions. The dimensionality of the model is then 30 observations ($n = 30$) and 6,319 features ($p = 6319$). In order to verify the effectiveness of our High-dimensional Left-censored Quantile Regression framework, we artificially created a 10% censoring on the response Ro1 value, and fitted the dataset for five quantiles, $\tau = 0.5, 0.75$, and 0.9. The regularization parameter in the initial estimator is chosen using a five-fold cross validation procedure as described in (4.22).

Table 4.5: Gene expressions selected by High-dimensional Left-censored Quantile Regression (HLQR) with 10% censoring in comparison with the ones selected by L_1 norm QR model in [LZ08] (L_1 QR) with no censoring

| GeneBank | $\tau = 0.5$ | | $\tau = 0.75$ | | $\tau = 0.9$ | |
|----------|----------------|----------|----------------|----------|----------------|----------|
| | HLQR | L_1 QR | HLQR | L_1 QR | HLQR | L_1 QR |
| D31717 | (97,68, 97.92) | ✓ | (97,65, 97.96) | ✓ | (97,61, 97.91) | ✓ |
| U73744 | (20,09, 20.32) | ✓ | (20,08, 20.29) | ✓ | (20,06, 20.31) | ✓ |
| U25708 | (46,61, 46.82) | ✓ | (46,60, 46.83) | ✓ | (46,60, 46.90) | |
| AA061310 | (9,07, 9.26) | ✓ | (9,07, 9.22) | | (9,05, 9.29) | |
| M30127 | (−0,04, 0.06) | ✓ | (−0,03, 0.05) | ✓ | (−0,04, 0.06) | ✓ |
| L38971 | (20,36, 20.54) | ✓ | (20,35, 20.54) | | (20,34, 20.58) | |
| Z32675 | (25,07, 25.28) | ✓ | (25,03, 25.15) | | (25,02, 25.36) | |
| W75373 | (41,96, 42.17) | ✓ | (41,94, 42.20) | | (41,94, 42.16) | |
| AA044561 | (0,02, 0.18) | | (−0,01, 0.28) | ✓ | (−0,05, 0.33) | |
| AA111168 | (−0,12, 0.22) | | (−0,10, 0.17) | | (−0,13, 0.21) | ✓ |
| M18194 | (−0,04, 0.10) | | (−0,12, 0.15) | | (−0,04, 0.09) | |

The gene expressions deemed to be significant by the confidence intervals are summarized in Table 4.5. We also noticed that the same dataset has also been studied in both [LZ08]. Thereby, we included real data results therein for comparison.

As one can see from Table 4.5, there are quite a few overlaps between the gene expressions selected in [LZ08] and the ones selected by our High-dimensional Left-censored Quantile Regression method, even with 10% of censoring introduced. In addition to merely identifying the significant genes, our methodology is capable of providing a precise confidence interval for the significant gene expressions. Moreover, we notice that the sets of selected genes by models across various quantiles, i.e. $\tau = 0.5, 0.75$, and 0.9 , using our HLQR are more consistent than the sets reported for models with different quantiles from L_1 QR. In other words, our methodology tends to agree on a common set of significant gene expressions across models with different quantile levels.

The starkest contrast between the gene expressions reported can be seen in M30127 (Mouse MHC class I tum-transplantation antigen P35B gene), whose importance has been noted consistently across quantiles in L_1 QR, whereas our HLQR procedure does not find the expression significant. Instead, we do notice that our resulting confidence interval does suggest

the significance of another gene expression M20985 (Mouse MHC class I H2-Qa-Mb1 gene). The confidence intervals for M20985 is as following (91.14, 91.32) in $\tau = 0.5$, (91.14, 91.30) in $\tau = 0.75$, and (91.11, 91.35) in $\tau = 0.9$. Whereas as of date the M30127 expression's role in the cardiomyopathy development is yet to be determined, [PSA⁺10] has confirmed that M20985 is part of a locus that confers susceptibility of viral-induced chronic myocarditis. In such case, our methodology has correctly identified a substantial gene candidate for further study of the disease.

Last but not the least, we would like to emphasize on the necessity of considering censoring data cases. In fact, it is difficult to accurately measure absolute expression levels and reliably detect low abundance genes [DKES06]. Thus, we believe our method would be a great asset for researchers analyzing datasets, which have observations with lower detection limit.

4.5 Lemmas

The following result gives a bound on the estimation error of our inverse Hessian estimator $\hat{\Theta}_j$ to the underlying population quantity Θ_j^0 .

Lemma 6. *Under Conditions 8 - 14,*

$$\|\hat{\Theta}_{\hat{\beta},j} - \Theta_{\beta^o,j}\|_1 = O_p(\lambda_j s_j) + O_p(K\sqrt{\lambda s \beta^o s_j}) + O_p(K(\lambda s \beta^o s_j^2/n)^{1/4}) + O_p(\sqrt{s_j \delta_{f,n} K}),$$

where $\delta_{f,n} := n^{-1} \sum_{i=1}^n (\hat{f}(\mathbf{x}_i \hat{\beta} | \mathbf{x}_i) - f_0(\mathbf{x}_i \beta^o | \mathbf{x}_i))^2$. For bounded case, $K = \sqrt{s_j}$, and $K = 1$ in the strongly bounded case.

Remark 7. *In particular, in the bounded case, if we choose $\lambda \asymp \sqrt{\log(p)/n}$, $\lambda_j \asymp \sqrt{\log(p)/n}$, $s_j^2 s \beta^o \sqrt{\log(p)/n} = o_p(1)$, $s \beta^o s_j^4 \sqrt{\log(p)/n^3} = o_p(1)$ and $s_j \sqrt{\delta_{f,n}} = o_p(1)$, then*

$$\|\hat{\Theta}_{\hat{\beta},j} - \Theta_{\beta^o,j}\|_1 = o_p(1).$$

In the strongly bounded case, we only require $\lambda \asymp \sqrt{\log(p)/n}$, $\lambda_j \asymp \sqrt{\log(p)/n}$,

$$s_j s_{\beta^o} \sqrt{\log(p)/n} = o_p(1)$$

and $s_j \delta_{f,n} = o_p(1)$.

Finally, we begin presenting preliminary results for each term in the decomposition (4.17). We start with term (III), which measures the error of the one-step improvement quantity using the estimator \hat{F}_n .

Lemma 7. *Under Condition 8 - 14, for \hat{F}_n chosen to be as in (4.18)*

$$III = -\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \phi_i + O_p \left(\frac{K}{n} + K \left(\frac{\log n}{n} \right)^{3/4} \right),$$

where $K = \sqrt{s_j}$, and in the strongly bounded case, $K = 1$, and

$$\phi_i := \tau \mathbb{1}(\mathbf{x}_i \beta^o \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} \sum_{\substack{l=1 \\ l \neq i}}^n B_{nl}(\mathbf{x}_i) \left(\frac{\mathbb{1}(Y_l > 0, \delta_l = 1)}{F_0(Y_l | \mathbf{x})} - \int_{\max\{0, Y_l\}}^{\infty} \frac{dF_0(s | \mathbf{x})}{F_0^2(s | \mathbf{x})} \right).$$

Furthermore,

$$\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \phi_i \xrightarrow{d} \mathcal{N} \left(0, \frac{\sigma_\phi^2}{n} \right),$$

where $\sigma_\phi^2 = \mathbb{E} \hat{\Theta}_j \Omega_\phi \hat{\Theta}_j^\top$ and $\Omega_\phi := \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i \phi_i^2 / n$.

Remark 8. *Lemma 7 implies that an additional normality term results from using the classical Kaplan-Meier estimator as a proxy for the true distribution F . Such a term can be understood as the extra variability due to the missing information regarding underlying distribution.*

In the following, we apply linearization on the term (Δ) and then combine the term together with (I), which then gives us the following Lemma. The rationale behind such arrangement is that

the term (Δ) describes the difference in the one-step correction with expectation of score using initial estimator $\hat{\beta}$, whereas the term (I) is exactly the difference of $\hat{\beta}$ and β^o .

Lemma 8. *Under Conditions 8 - 14, when $\left\| \hat{\Theta}_{\hat{\beta},j} - \Theta_{\beta^o,j} \right\|_1 = o_p(1)$,*

$$|I - \Delta| = O_p(K\lambda_j\lambda s_{\beta^o}) + O_p(K\lambda^2 s_{\beta^o}^2),$$

where $K = \sqrt{s_j}$, and in the strongly bounded case, $K = 1$.

For part (II), we have the following lemma, which aims to bound the difference of a empirical process.

Lemma 9. *Under Conditions 8 - 14,*

$$|II| = O_p\left(\sqrt{\lambda s_{\beta^o} s_j / n}\right).$$

Last but not the least, we show the normality of the term $\sqrt{n}\hat{\Theta}\mathbb{S}_n(\beta^o)$ for part (N). The lemma shows that the leading term of the Bahadur decomposition (4.17) follows a normal distribution.

Lemma 10. *Assuming Conditions 8 - 14,*

$$N = -\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \Psi_i \xrightarrow{d} \mathcal{N}\left(0, \frac{\sigma_\Psi^2}{n}\right),$$

where $\Psi_i = -[w_i(F_0) \mathbb{1}\{Y_i - \mathbf{x}_i \beta^o \geq 0\} - (1 - \tau)]$, and $\sigma_\Psi^2 = \mathbb{E} \hat{\Theta}_j \Omega_\Psi \hat{\Theta}_j^\top$ and

$$\Omega_\Psi := \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i \Psi_i^2 / n.$$

4.6 Proofs of Lemmas

Proof of Lemma 6. Let \mathbf{w}_β be the diagonal of the weighted matrix \mathbf{w}_β . Denote

$$\mathbf{x}_{\beta^o, j} = \mathbf{x}_{\beta^o, -j} \gamma_{\beta^o, j} + \boldsymbol{\eta}_{\beta^o, j}, \quad (4.23)$$

and

$$\hat{\gamma}_{\hat{\beta}, j} = \arg \min_{\gamma} \|\mathbf{x}_{\hat{\beta}, j} - \mathbf{x}_{\hat{\beta}, -j} \gamma\|_n^2 + 2\lambda_j \|\gamma\|_1,$$

where $\gamma_{\beta^o, j} = \arg \min_{\gamma} \mathbb{E} \|\mathbf{x}_{\beta^o, j} - \mathbf{x}_{\beta^o, -j} \gamma\|_n^2$. Define

$$\boldsymbol{\eta}_j := \mathbf{x}_j - \mathbf{x}_{-j} \gamma_{\beta^o, j}, \quad (4.24)$$

we can rewrite equation (4.23) as

$$\mathbf{w}_{\beta^o} \mathbf{x}_j = \mathbf{w}_{\beta^o} \mathbf{x}_{-j} \gamma_{\beta^o, j} + \mathbf{w}_{\beta^o} \boldsymbol{\eta}_j,$$

and similarly by (4.24), we also have

$$\mathbf{w}_{\hat{\beta}} \mathbf{x}_j = \mathbf{w}_{\hat{\beta}} \mathbf{x}_{-j} \hat{\gamma}_{\hat{\beta}, j} + \mathbf{w}_{\hat{\beta}} \boldsymbol{\eta}_j. \quad (4.25)$$

By the definition of $\hat{\gamma}_{\hat{\beta}, j}$,

$$\begin{aligned} & \|\mathbf{x}_{\hat{\beta}, j} - \mathbf{x}_{\hat{\beta}, -j} \hat{\gamma}_{\hat{\beta}, j}\|_n^2 + 2\lambda_j \|\hat{\gamma}_{\hat{\beta}, j}\|_1 \\ & \leq \|\mathbf{x}_{\hat{\beta}, j} - \mathbf{x}_{\hat{\beta}, -j} \gamma_{\beta^o, j}\|_n^2 + 2\lambda_j \|\gamma_{\beta^o, j}\|_1. \end{aligned}$$

Replacing $\hat{\mathbf{x}}_{\hat{\beta},j}$ by (4.25) and rearranging terms, we get

$$\begin{aligned}
& \|\mathbf{x}_{\hat{\beta},-j}(\hat{\gamma}_{\hat{\beta},j} - \gamma_{\beta^o,j})\|_n^2 + 2\lambda_j \|\hat{\gamma}_{\hat{\beta},j}\|_1 \\
& \leq \frac{2}{n} \left(\mathbf{w}_{\hat{\beta}}^2 \boldsymbol{\eta}_j \right)^\top \mathbf{x}_{-j}(\hat{\gamma}_{\hat{\beta},j} - \gamma_{\beta^o,j}) + 2\lambda_j \|\gamma_{\beta^o,j}\|_1 \\
& = \frac{2}{n} \boldsymbol{\eta}_{\beta^o,j}^\top \mathbf{x}_{\beta^o,-j}(\hat{\gamma}_{\hat{\beta},j} - \gamma_{\beta^o,j}) + 2\lambda_j \|\gamma_{\beta^o,j}\|_1 + \text{Rem},
\end{aligned}$$

where the remainder $\text{Rem} = (2/n) \left((\mathbf{w}_{\hat{\beta}}^2 - \mathbf{w}_{\beta^o}^2) \boldsymbol{\eta}_j \right)^\top \mathbf{x}_{-j}(\hat{\gamma}_{\hat{\beta},j} - \gamma_{\beta^o,j})$. Note that by Condition 12, $\|\boldsymbol{\eta}_j\|_\infty \leq \|\mathbf{x}_j\|_\infty + \|\mathbf{x}_{-j}\gamma_{\beta^o,j}\|_\infty = O_p(\sqrt{s_j})$. In the strongly bounded case, we have the projection $\|\mathbf{x}_{\beta^o,-j}\gamma_{\beta^o,j}\|_\infty = O_p(1)$, hence $\|\boldsymbol{\eta}_j\|_\infty = O_p(1)$. In the following, we write $\|\boldsymbol{\eta}_j\|_\infty = O_p(K)$ where $K = \sqrt{s_j}$ in general case, and $K = 1$ when data is strongly bounded.

We can bound the remainder term

$$|\text{Rem}| \leq \frac{2}{n} \|(\mathbf{w}_{\hat{\beta}}^2 - \mathbf{w}_{\beta^o}^2) \boldsymbol{\eta}_j\|_2 \|\mathbf{x}_{-j}(\hat{\gamma}_{\hat{\beta},j} - \gamma_{\beta^o,j})\|_2.$$

Therefore,

$$\begin{aligned}
& n^{-1} \|(\mathbf{w}_{\hat{\beta}}^2 - \mathbf{w}_{\beta^o}^2) \boldsymbol{\eta}_{\beta^o, j}\|_2^2 \\
& \leq \frac{1}{n} \|\boldsymbol{\eta}_{\beta^o, j}\|_\infty^2 \sum_{i=1}^n (\mathbf{w}_{\hat{\beta}, i}^2 - \mathbf{w}_{\beta^o, i}^2)^2 \\
& = \frac{1}{n} \|\boldsymbol{\eta}_{\beta^o, j}\|_\infty^2 \sum_{i=1}^n (\hat{f}(\mathbf{x}_i \hat{\beta} | \mathbf{x}_i) \mathbb{1}(\mathbf{x}_i \hat{\beta} > 0) - f_0(\mathbf{x}_i \beta^o | \mathbf{x}_i) \mathbb{1}(\mathbf{x}_i \beta^o > 0))^2 \\
& \leq \frac{1}{n} \|\boldsymbol{\eta}_{\beta^o, j}\|_\infty^2 \left\{ \sum_{i=1}^n (\hat{f}(\mathbf{x}_i \hat{\beta} | \mathbf{x}_i) - f_0(\mathbf{x}_i \hat{\beta} | \mathbf{x}_i))^2 + \sum_{i=1}^n (f_0(\mathbf{x}_i \hat{\beta} | \mathbf{x}_i) - f_0(\mathbf{x}_i \beta^o | \mathbf{x}_i))^2 \right\} \\
& \quad + \frac{1}{n} \|\boldsymbol{\eta}_{\beta^o, j}\|_\infty \sum_{i=1}^n f_0(\mathbf{x}_i \beta^o | \mathbf{x}_i)^2 (\mathbb{1}(\mathbf{x}_i \hat{\beta} > 0) - \mathbb{1}(\mathbf{x}_i \beta^o > 0))^2 \\
& = \delta_{f, n} O_p(K^2) + \frac{1}{n} \|\mathbf{x}(\hat{\beta} - \beta^o)\|_2^2 O_p(K^2) + \frac{1}{n} \sum_{i=1}^n (\mathbb{1}(\mathbf{x}_i \hat{\beta} > 0) - \mathbb{1}(\mathbf{x}_i \beta^o > 0))^2 O_p(K^2) \\
& = O(\delta_{f, n} K^2) + O_p(\lambda^2 s_{\beta^o} K^2) + O_p(K^2) \frac{1}{n} \sum_{i=1}^n \mathcal{B}_i(\hat{\beta})
\end{aligned}$$

where $\delta_{f, n} = n^{-1} \sum_{i=1}^n (\hat{f}(\mathbf{x}_i \hat{\beta} | \mathbf{x}_i) - f_0(\mathbf{x}_i \beta^o | \mathbf{x}_i))^2$ and $\mathcal{B}_i(\beta) = (\mathbb{1}(\mathbf{x}_i \beta > 0) - \mathbb{1}(\mathbf{x}_i \beta^o > 0))^2$. Observe that for any fixed β , $\mathcal{B}_i(\beta)$ is Bernoulli random variable. Let $\mathcal{P} = \mathbb{P}(\mathcal{B}_i = 1)$. Note that

$$\max_i |\mathbf{x}_i \beta - \mathbf{x}_i \beta^o| = \|\mathbf{x} \beta - \mathbf{x} \beta^o\|_\infty \leq \|\mathbf{x}\|_\infty \|\beta - \beta^o\|_1 \leq K_X \|\beta - \beta^o\|_1,$$

and

$$\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i \beta - \mathbf{x}_i \beta^o)^2 = n^{-1} \|\mathbf{x}(\beta - \beta^o)\|_2^2.$$

Therefore, $\mathcal{P} \leq \mathbb{P}(|\mathbf{x}_i \beta^o| \leq K_X \|\beta - \beta^o\|_1) = O(\|\beta - \beta^o\|_1)$ by the boundedness of density f_0 .

By Chernoff inequality,

$$\left| \frac{1}{n} \sum_{i=1}^n \mathcal{B}_i(\beta) \right| = O_p(\mathcal{P}) + O_p\left(\frac{\sqrt{\mathcal{P}(1-\mathcal{P})}}{\sqrt{n}}\right).$$

Hence, we have

$$\left| \frac{1}{n} \sum_{i=1}^n \mathcal{B}_i(\hat{\beta}) \right| = O_p(\lambda s \beta^o) + O_p\left(\frac{\sqrt{\lambda s \beta^o}}{\sqrt{n}}\right).$$

Therefore, for any $\delta > 0$,

$$\begin{aligned} |\text{Rem}| &= \delta \|\mathbf{x}_{\hat{\beta}, -j}(\hat{\gamma}_{\hat{\beta}, j} - \gamma_{\beta^o, j})\|_n^2 + O(\delta_{f,n} K^2) \\ &\quad + O_p(\lambda^2 s \beta^o K^2) + O_p(K^2 \lambda s \beta^o) + O_p(K^2 \sqrt{\lambda s \beta^o / n}). \end{aligned}$$

By the standard arguments, choosing $\lambda_j \asymp \sqrt{\log(p)/n}$, we get

$$\|\mathbf{x}_{\hat{\beta}, -j}(\hat{\gamma}_{\hat{\beta}, j} - \gamma_{\beta^o, j})\|_n^2 = O_p(\lambda_j^2 s_j) + O(\delta_{f,n} K^2) + O_p(\lambda s \beta^o K^2) + O_p(K^2 \sqrt{\lambda s \beta^o / n})$$

and

$$\|\hat{\gamma}_{\hat{\beta}, j} - \gamma_{\beta^o, j}\|_1 = O_p(\lambda_j s_j) + O(\sqrt{\delta_{f,n} K \sqrt{s_j}}) + O_p(K \sqrt{\lambda s \beta^o s_j}) + O_p(K(\lambda s \beta^o s_j^2 / n)^{1/4}).$$

Using (4.24) again, we get

$$\hat{d}_{\hat{\beta}, j}^2 - d_{\beta^o, j}^2 = \underbrace{\mathbf{x}_{\beta^o, j}^\top (\mathbf{x}_{\beta^o, j} - \mathbf{x}_{\beta^o, -j} \hat{\gamma}_{\hat{\beta}, j}) / n - d_{\beta^o, j}^2}_{(i)} + \underbrace{\mathbf{x}_j^\top (\mathbf{w}_{\hat{\beta}}^2 - \mathbf{w}_{\beta^o}^2) (\mathbf{x}_j - \mathbf{x}_{-j} \hat{\gamma}_{\hat{\beta}, j}) / n}_{(ii)}.$$

By Theorem 2.4 in [VdGBRD14], we have (i) = $O_p(\lambda_j \sqrt{s_j})$. For the second part (ii), by Condition 12,

$$\begin{aligned} (ii) &= O_p(K) \frac{1}{n} \sum_{i=1}^n |\hat{f}(\mathbf{x}_i \hat{\beta} | \mathbf{x}_i) \mathbb{1}(\mathbf{x}_i \hat{\beta} > 0) - f_0(\mathbf{x}_i \beta^o | \mathbf{x}_i) \mathbb{1}(\mathbf{x}_i \beta^o > 0)| \\ &= O_p(\sqrt{\delta_{f,n}} K) + O_p(\lambda \sqrt{s \beta^o} K) + O_p(K(\lambda s \beta^o / n)^{1/4}). \end{aligned}$$

Therefore,

$$\left| \hat{d}_{\hat{\beta},j}^2 - d_{\beta^o,j}^2 \right| = O_p(\lambda_j \sqrt{s_j}) + O_p(\sqrt{\delta_{f,n} K}) + O_p(\lambda \sqrt{s \beta^o} K) + O_p(K(\lambda s \beta^o / n)^{1/4}).$$

Combining all previous results,

$$\begin{aligned} & \|\hat{\Theta}_{\hat{\beta},j} - \Theta_{\beta^o,j}\|_1 \\ & \leq \|\hat{\gamma}_{\hat{\beta},j} - \gamma_{\beta^o,j}\|_1 / \hat{d}_{\hat{\beta},j}^2 + \|\gamma_{\beta^o,j}\|_1 \left(1 / \hat{d}_{\hat{\beta},j}^2 - 1 / d_{\beta^o,j}^2 \right) \\ & = O_p(\lambda_j s_j) + O_p(K \sqrt{\lambda s \beta^o s_j}) + O_p(K(\lambda s \beta^o s_j^2 / n)^{1/4}) + O_p(\sqrt{s_j \delta_{f,n} K}). \end{aligned}$$

□

Proof of Lemma 7. We begin with expanding on the following difference,

$$\hat{\Theta}_j (\mathbb{S}_n(\hat{\beta}, \hat{F}_n) - \mathbb{S}_n(\hat{\beta}, F_0)) = \hat{\Theta}_j \left. \frac{\partial \mathbb{S}_n(\hat{\beta}, F)}{\partial F} \right|_{F=F_0} (\hat{F}_n - F_0) + \frac{1}{2} \left. \frac{\partial^2 \mathbb{S}_n(\hat{\beta}, F)}{\partial F^2} \right|_{F=\tilde{F}} (\hat{F}_n - F_0)^2 \quad (4.26)$$

for some \tilde{F} between \hat{F}_n and F_0 . We then work on rewriting the terms in the summation of $\mathbb{S}_n(\beta, F)$.

Let $\mathbb{S}_n(\boldsymbol{\beta}, F) := n^{-1} \sum_{i=1}^n \mathbb{S}_i(\boldsymbol{\beta}, F)$,

$$\begin{aligned}
\mathbb{S}_i(\boldsymbol{\beta}, F) &= -\mathbf{x}_i^\top [w_i(F) \mathbb{1}(Y_i - \mathbf{x}_i \boldsymbol{\beta} \geq 0) + \tau - 1] \\
&= -\mathbf{x}_i^\top [\mathbb{1}(T_i \leq 0) (\tau - 1 + \frac{\tau}{F} \mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} \leq 0) \mathbb{1}(F > \tau)) \\
&\quad + \mathbb{1}(T_i > 0) (\tau - 1 + \mathbb{1}(T_i \geq \mathbf{x}_i \boldsymbol{\beta}))] \\
&= -\mathbf{x}_i^\top [\mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} \leq 0) (\mathbb{1}(T_i \leq 0)(\tau - 1) + \mathbb{1}(T_i \leq 0) \frac{\tau}{F} \mathbb{1}(F > \tau) + \tau \mathbb{1}(T_i > 0)) \\
&\quad + \mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} > 0) (\mathbb{1}(T_i \leq 0)(\tau - 1) + \mathbb{1}(T_i > 0)(\tau - 1) + \mathbb{1}(T_i \geq \mathbf{x}_i \boldsymbol{\beta}))] \\
&= -\mathbf{x}_i^\top [\mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} \leq 0) (\tau - \mathbb{1}(T_i \leq 0) + \mathbb{1}(T_i \leq 0) \frac{\tau}{F} \mathbb{1}(F > \tau)) \\
&\quad + \mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} > 0) (\tau - 1 + \mathbb{1}(T_i \geq \mathbf{x}_i \boldsymbol{\beta}))] \\
&= -\mathbf{x}_i^\top [\tau - \mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} \leq 0, T_i \leq 0) + \mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} \leq 0, T_i \leq 0) \frac{\tau}{F} \mathbb{1}(F > \tau) \\
&\quad - \mathbb{1}(\mathbf{x}_i \boldsymbol{\beta} > 0) + \mathbb{1}(T_i \geq \mathbf{x}_i \boldsymbol{\beta}, \mathbf{x}_i \boldsymbol{\beta} > 0)].
\end{aligned}$$

We derive the first derivative of \mathbb{S}_n with respect to F at F_0 ,

$$\begin{aligned}
\left. \frac{\partial \mathbb{S}_n(\hat{\boldsymbol{\beta}}, F)}{\partial F} \right|_{F=F_0} &= \lim_{\varepsilon \rightarrow 0} -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\boldsymbol{\beta}} \leq 0, T_i \leq 0) \\
&\quad \times \frac{1}{\varepsilon(F-F_0)} \left(\frac{\mathbb{1}(F_0 + \varepsilon(F-F_0) > \tau)}{F_0 + \varepsilon(F-F_0)} - \frac{\mathbb{1}(F_0 > \tau)}{F_0} \right) \\
&= -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\boldsymbol{\beta}} \leq 0, T_i \leq 0) \\
&\quad \times \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon(F-F_0)} \left(\frac{\mathbb{1}(F_0 + \varepsilon(F-F_0) > \tau)}{F_0 + \varepsilon(F-F_0)} - \frac{\mathbb{1}(F_0 > \tau)}{F_0 + \varepsilon(F-F_0)} \right. \\
&\quad \left. + \frac{\mathbb{1}(F_0 > \tau)}{F_0 + \varepsilon(F-F_0)} - \frac{\mathbb{1}(F_0 > \tau)}{F_0} \right) \\
&= \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\boldsymbol{\beta}} \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2},
\end{aligned}$$

where the details of taking the limit is as the following.

$$\begin{aligned}
& \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon(F-F_0)} \left(\frac{\mathbb{1}(F_0+\varepsilon(F-F_0) > \tau)}{F_0+\varepsilon(F-F_0)} - \frac{\mathbb{1}(F_0 > \tau)}{F_0+\varepsilon(F-F_0)} + \frac{\mathbb{1}(F_0 > \tau)}{F_0+\varepsilon(F-F_0)} - \frac{\mathbb{1}(F_0 > \tau)}{F_0} \right) \\
&= \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon(F-F_0)} \left(\frac{\mathbb{1}(F_0+\varepsilon(F-F_0) > \tau) - \mathbb{1}(F_0 > \tau)}{F_0+\varepsilon(F-F_0)} - \frac{\varepsilon(F-F_0)}{F_0(F_0+\varepsilon(F-F_0))} \mathbb{1}(F_0 > \tau) \right) \\
& \quad = -\frac{\mathbb{1}(F_0 > \tau)}{F_0^2},
\end{aligned}$$

since F_0 is bounded away from τ . Likewise, we have the second derivative of \mathbb{S}_n with respect to F at \tilde{F} as

$$\left. \frac{\partial^2 \mathbb{S}_n(\hat{\beta}, F)}{\partial F^2} \right|_{F=\tilde{F}} = -\frac{2}{n} \sum_{i=1}^n \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\beta} \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(\tilde{F} > \tau)}{\tilde{F}^3},$$

as for F close to F_0 , \tilde{F} is also bounded away from τ .

Plugging the derivatives into (4.26), we have

$$\begin{aligned}
\hat{\Theta}_j(\mathbb{S}_n(\hat{\beta}, \hat{F}_n) - \mathbb{S}_n(\hat{\beta}, F_0)) &= \underbrace{\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\beta} \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} (\hat{F}_n - F_0)}_{(i)} \\
&\quad - \underbrace{\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\beta} \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(\tilde{F} > \tau)}{\tilde{F}^3} (\hat{F}_n - F_0)^2}_{(ii)}.
\end{aligned}$$

Following the framework of Theorem 1 of [LS86] and Theorem 2.3 of [GMCS94] that for the classical Kaplan-Meier estimator \hat{F}_n as defined in (4.18), we have the following linearization.

$$\hat{F}_n(0|\mathbf{x}) - F_0(0|\mathbf{x}) = \frac{1}{n} \sum_{l=1}^n \zeta(Y_l, \delta_l, \mathbf{x}) + O_p \left(\left(\frac{\log n}{n} \right)^{3/4} \right) = O_p \left(\frac{1}{\sqrt{n}} + \left(\frac{\log n}{n} \right)^{3/4} \right)$$

for some Θ_i between $(\mathbf{x} - \mathbf{x}_i)/h_n$ and $(\mathbf{x} - \mathbf{x}_l)/h_n$, where

$$\zeta(Y_l, \delta_l, \mathbf{x}) = \frac{\mathbb{1}(Y_l > 0, \delta_l = 1|\mathbf{x})}{F_0(Y_l|\mathbf{x})} - \int_{\max\{0, Y_l\}}^{\infty} \frac{dF_0(s|\mathbf{x})}{F_0^2(s|\mathbf{x})}.$$

In fact, for $i \neq l$, $\mathbb{1}(T_i \leq 0) \zeta(Y_l, \delta_l, \mathbf{x})$ are independent random variables with mean zero and finite variances for any given \mathbf{x} .

Replacing the term $(\hat{F}_n - F_0)$ with its linearization, and separating the terms of $i = l$ from $i \neq l$, for term (i), we have

$$(i) = \frac{1}{n^2} \sum_{i=1}^n \sum_{\substack{l=1 \\ l \neq i}}^n \hat{\Theta}_j \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \beta^o \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} B_{nl}(\mathbf{x}_i) \zeta(Y_l, \delta_l, \mathbf{x}_i) \\ + \frac{1}{n^2} \sum_{i=1}^n \sum_{\substack{l=1 \\ l \neq i}}^n \hat{\Theta}_j \mathbf{x}_i^\top \tau (\mathbb{1}(\mathbf{x}_i \hat{\beta} \leq 0) - \mathbb{1}(\mathbf{x}_i \beta^o \leq 0)) \quad (4.27)$$

$$\times \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} B_{nl}(\mathbf{x}_i) \zeta(Y_l, \delta_l, \mathbf{x}_i) \\ + \frac{1}{n^2} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\beta} \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} B_{ni}(\mathbf{x}_i) \zeta(Y_i, \delta_i, \mathbf{x}_i) \\ + \left(\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\beta} \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} \right) \cdot O_p \left(\left(\frac{\log n}{n} \right)^{3/4} \right) \\ = \frac{1}{n^2} \sum_{i=1}^n \sum_{\substack{l=1 \\ l \neq i}}^n \hat{\Theta}_j \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \beta^o \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} B_{nl}(\mathbf{x}_i) \zeta(Y_l, \delta_l, \mathbf{x}_i) \quad (4.28)$$

$$+ O_p \left(\frac{K \lambda_S \beta^o}{n} \right) + O_p \left(\frac{K}{n^{3/2}} \right) + O_p \left(K \left(\frac{\log n}{n} \right)^{3/4} \right), \quad (4.29)$$

where $K = \sqrt{s_j}$, and in the strongly bounded case, $K = 1$. The order in (4.29) results from the condition that $\|\hat{\Theta}_{\hat{\beta}, j} - \Theta_{\beta^o, j}\|_1 = o_p(1)$, and similar arguments as in Lemma 6. For the other term (ii), we can bound it as following,

$$(ii) = \left(\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \tau \mathbb{1}(\mathbf{x}_i \hat{\beta} \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(\hat{F} > \tau)}{\hat{F}^3} \right) \cdot O_p \left(\frac{1}{n} + \left(\frac{\log n}{n} \right)^{3/2} + \frac{\log^{3/4} n}{n^{4/5}} \right) \\ = O_p \left(\frac{K}{n} + K \left(\frac{\log n}{n} \right)^{3/2} \right).$$

For convenience in notations, define random variables ϕ_i as following,

$$\phi_i := \tau \mathbb{1}(\mathbf{x}_i \beta^o \leq 0) \mathbb{1}(T_i \leq 0) \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} \sum_{\substack{l=1 \\ l \neq i}}^n B_{nl}(\mathbf{x}_i) \zeta(Y_l, \delta_l, \mathbf{x}_i).$$

Then $\{\hat{\Theta}_j \mathbf{x}_i^\top \phi_i\}_{i=1}^n$ are i.i.d. mean zero random variables with finite variance. Thus, by the central

limit theorem, (4.28) $\xrightarrow{d} \mathcal{N}(0, \sigma_1^2/n)$, where $\sigma_1^2 = \mathbb{E} \hat{\Theta}_j \Omega_1 \hat{\Theta}_j^\top$, and $\Omega_1 := \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i \phi_i^2/n$.

□

Lemma 11 (Preliminary Result for Lemma 8). *By the construction of inverse matrix Θ^0 and $\hat{\Theta}$, we have $1/\hat{d}_j^2 = O(1)$.*

Proof of Lemma 11. First, we note that $\Theta_{j,j}^0 = 1/d_j^2$, which is a result of the KKT condition following similar arguments as in 2.3.1 of [BG16]. Second, following the proof of lemma 5.3 in [VdGBRD14], we can show $\hat{d}_j^2 = d_j^2 + o_p(1)$. Then the results follows from Condition 14. □

Proof of Lemma 8. We will suppress F_0 in the argument of \mathbb{S}_n for the proof, and start by first examining part of Δ . Denote $\mathbb{H}() = [\partial \mathbb{E} \mathbb{S}_n(\beta) / \partial \beta]_{\beta=}$,

$$\begin{aligned} \mathbb{E} \mathbb{S}_n(\hat{\beta}) - \mathbb{E} \mathbb{S}_n(\beta^o) &= \mathbb{H}() (\hat{\beta} - \beta^o) \\ &= \mathbb{H}(\hat{\beta}) (\hat{\beta} - \beta^o) + (\mathbb{H}() - \mathbb{H}(\hat{\beta})) (\hat{\beta} - \beta^o). \end{aligned}$$

Thus, we can rewrite Δ as

$$\Delta = \hat{\Theta}_j \mathbb{H}(\hat{\beta}) (\hat{\beta} - \beta^o) + \hat{\Theta}_j (\mathbb{H}() - \mathbb{H}(\hat{\beta})) (\hat{\beta} - \beta^o).$$

Subtracting (Δ) from (I) , we have

$$\begin{aligned} I - \Delta &= \hat{\beta}_j - \beta_j^o - \hat{\Theta}_j \mathbb{H}(\hat{\beta}) (\hat{\beta} - \beta^o) - \hat{\Theta}_j (\mathbb{H}() - \mathbb{H}(\hat{\beta})) (\hat{\beta} - \beta^o) \\ &= \underbrace{(e_j^T - \hat{\Theta}_j \mathbb{H}(\hat{\beta})) (\hat{\beta} - \beta^o)}_{(i)} + \underbrace{\hat{\Theta}_j (\mathbb{H}(\hat{\beta}) - \mathbb{H}()) (\hat{\beta} - \beta^o)}_{(ii)} \end{aligned}$$

Using the KKT condition described in (4.12), we could work out a bound for (i). In more

detail,

$$\begin{aligned}
|(e_j^T - \hat{\Theta}_j \mathbb{H}(\hat{\beta}))(\hat{\beta} - \beta^o)| &\leq \|(e_j^T - \hat{\Theta}_j \mathbb{H}(\hat{\beta}))\|_\infty \|\hat{\beta} - \beta^o\|_1 \\
&\leq \frac{\lambda_j}{\hat{d}_j^2} \|\hat{\beta} - \beta^o\|_1 \\
&= O_p(\lambda_j \lambda s_{\beta^o})
\end{aligned}$$

where the last inequality is due to the consistency result of Theorem 10 and the fact that $1/\hat{d}_j^2$ is bounded, which is shown in Lemma 11. Now for part (ii),

$$\begin{aligned}
&|\hat{\Theta}_j(\mathbb{H}(\cdot) - \mathbb{H}(\hat{\beta}))(\hat{\beta} - \beta^o)| \\
\leq &\left| \frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^T \cdot \mathbf{x}_i (\mathbb{1}(\mathbf{x}_i > 0) f_0(\mathbf{x}_i | \mathbf{x}_i) - \mathbb{1}(\mathbf{x}_i \hat{\beta} > 0) f_0(\mathbf{x}_i | \hat{\beta} | \mathbf{x}_i)) (\hat{\beta} - \beta^o) \right| \\
\leq &\left| \frac{L}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^T (\mathbf{x}_i (\hat{\beta} - \beta^o))^2 \right| + M \left| \frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^T \mathbf{x}_i (\hat{\beta} - \beta^o) (\mathbb{1}(\mathbf{x}_i > 0) - \mathbb{1}(\mathbf{x}_i \hat{\beta} > 0)) \right| \\
\leq &L \|\mathbf{x} \hat{\Theta}_j^\top\|_\infty \|\mathbf{x}(\hat{\beta} - \beta^o)\|_2 / n + MK_X \|\mathbf{x} \hat{\Theta}_j^\top\|_\infty \|\hat{\beta} - \beta^o\|_1 \frac{1}{n} \sum_{i=1}^n |\mathbb{1}(\mathbf{x}_i > 0) - \mathbb{1}(\mathbf{x}_i \hat{\beta} > 0)| \\
= &O_p(K \lambda^2 s_{\beta^o}) + O_p(K \lambda s_{\beta^o}) \frac{1}{n} \sum_{i=1}^n \mathcal{B}_i.
\end{aligned}$$

When $\|\hat{\Theta}_j - \Theta_j^0\|_1 = o_p(1)$, the term $\|\mathbf{x} \hat{\Theta}_j^\top\|_\infty$ is $O_p(K)$, where $K = \sqrt{s_j}$ in the bounded case, and $K = 1$ in the strongly bounded case. By similar argument in Lemma 6, $n^{-1} \sum_{i=1}^n \mathcal{B}_i = O_p(\lambda s_{\beta^o})$.

Putting parts of (i) and (ii) together, we have

$$|I - \Delta| = O_p(\lambda_j \lambda s_{\beta^o}) + O_p(K \lambda^2 s_{\beta^o}^2).$$

□

Proof of Lemma 9. Suppressing the argument F_0 for simplicity of notation, define

$$\begin{aligned}\Xi(\boldsymbol{\beta}) &= \boldsymbol{\Theta}_j [\mathbb{S}_n(\boldsymbol{\beta}) - \mathbb{S}_n(\boldsymbol{\beta}^o)] - \boldsymbol{\Theta}_j [\mathbb{E}\mathbb{S}_n(\boldsymbol{\beta}) - \mathbb{E}\mathbb{S}_n(\boldsymbol{\beta}^o)] \\ &= \underbrace{\boldsymbol{\Theta}_j [\mathbb{S}_n(\boldsymbol{\beta}) - \mathbb{S}_n(\boldsymbol{\beta}^o)]}_{\bar{\boldsymbol{x}}\boldsymbol{i}_n} - \mathbb{E}\boldsymbol{\Theta}_j [\mathbb{S}_n(\boldsymbol{\beta}) - \mathbb{S}_n(\boldsymbol{\beta}^o)],\end{aligned}$$

where the expectation is with respect to response variables T_i and $\boldsymbol{\Theta}$ is any p by p matrix with $\|\boldsymbol{\Theta}_j\| = O(\sqrt{s_j})$ (s_j is still the j -th row cardinality of $\boldsymbol{\Theta}^o$). So in another word, $\boldsymbol{\Theta}$ is any matrix with the same row cardinality as $\boldsymbol{\Theta}^o$). Then the term (II) is just $\Xi(\hat{\boldsymbol{\beta}})$ with $\boldsymbol{\Theta} = \hat{\boldsymbol{\Theta}}$. Note that

$$\bar{\boldsymbol{x}}\boldsymbol{i}_n = \frac{\sqrt{s_j}}{n} \sum_{i=1}^n s_j^{-1/2} \underbrace{\boldsymbol{\Theta}_j \boldsymbol{x}_i^T w_i [\mathbb{1}(Y_i \geq \boldsymbol{x}_i \boldsymbol{\beta}^o) - \mathbb{1}(Y_i \geq \boldsymbol{x}_i \boldsymbol{\beta})]}_{\xi_i}.$$

Now for any i , without loss of generality, assume $\boldsymbol{x}_i \boldsymbol{\beta} > \boldsymbol{x}_i \boldsymbol{\beta}^o \geq 0$. Then $\xi_i = \tilde{\xi}_i / \boldsymbol{\Theta}_j \boldsymbol{x}_i^T w_i$ is a Bernoulli random variable

$$\tilde{\xi}_i = \begin{cases} 1, & \text{if } \boldsymbol{x}_i \boldsymbol{\beta}^o \leq Y_i < \boldsymbol{x}_i \boldsymbol{\beta} \\ 0, & \text{elsewhere} \end{cases}$$

and $\mathbb{P}(\xi_i = 1) = F_0(\boldsymbol{x}_i \boldsymbol{\beta} | \boldsymbol{x}_i) - F_0(\boldsymbol{x}_i \boldsymbol{\beta}^o | \boldsymbol{x}_i) = f_0(\boldsymbol{x}_i | \boldsymbol{x}_i) \boldsymbol{x}_i (\boldsymbol{\beta} - \boldsymbol{\beta}^o)$ for some $\boldsymbol{x}_i \boldsymbol{\beta}^o < \boldsymbol{x}_i < \boldsymbol{x}_i \boldsymbol{\beta}$. Therefore, $\text{Var}(\xi_i) \leq \mathbb{P}(\xi_i = 1) = O_p(\|\boldsymbol{\beta} - \boldsymbol{\beta}^o\|_1)$ by Condition 9 and 12, and so is the variance of $\tilde{\xi}_i$ because $\|s_j^{-1/2} \boldsymbol{\Theta}_j(\boldsymbol{\beta}) \boldsymbol{x}_i^T w_i\|_\infty$ is bounded. Furthermore, it is easy to see that $\tilde{\xi}_i$ is a stochastically bounded random variable, say $|\tilde{\xi}_i| \leq a$ almost surely. Then $\text{Var}(\tilde{\xi}_i)/a = O_p(\|\boldsymbol{\beta} - \boldsymbol{\beta}^o\|_1)$ and this holds true for all $\boldsymbol{\beta}$. Invoking Bennett's inequality and the fact $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^o\|_1 = O_p(s_{\boldsymbol{\beta}^o} \lambda)$, we have $\Xi(\hat{\boldsymbol{\beta}}) = O_p(\sqrt{\lambda s_{\boldsymbol{\beta}^o} s_j / n})$, and hence Lemma 9. \square

Proof of Lemma 10. We start by rewriting part of term (N), we note that

$$\begin{aligned}\mathbb{S}_n(\beta^o, F_0) &= -\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top [w_i(F_0) \mathbb{1}\{Y_i - \mathbf{x}_i \beta^o \geq 0\} - (1 - \tau)] \\ &= \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \boldsymbol{\psi}_i\end{aligned}$$

where $\boldsymbol{\psi}_i = -[w_i(F_0) \mathbb{1}\{Y_i - \mathbf{x}_i \beta^o \geq 0\} - (1 - \tau)]$. It is easy to show that, for each i ,

$$\mathbb{E}[\boldsymbol{\psi}_i | \mathbf{x}_i] = -(\tau - \mathbb{P}(Y_i < \mathbf{x}_i \beta^o) - \tau(\mathbb{1}(\mathbf{x}_i \beta^o \leq 0))^2) = 0.$$

Furthermore, $|\boldsymbol{\psi}_i| \leq 1$. Then we can apply Lindeberg central limit theorem to random variable $\{\hat{\boldsymbol{\Theta}}_j \mathbf{x}_i^\top \boldsymbol{\psi}_i\}_{i=1}^n$. We have

$$\hat{\boldsymbol{\Theta}}_j \mathbb{S}_n(\beta^o, F_0) = \frac{1}{n} \sum_{i=1}^n \hat{\boldsymbol{\Theta}}_j \mathbf{x}_i^\top \boldsymbol{\psi}_i \xrightarrow{d} \mathcal{N}\left(0, \frac{\sigma_\Psi^2}{n}\right),$$

where $\sigma_2^2 = \mathbb{E} \hat{\boldsymbol{\Theta}}_j^\top [n^{-1} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i \boldsymbol{\psi}_i^2] \hat{\boldsymbol{\Theta}}_j = \mathbb{E} \hat{\boldsymbol{\Theta}}_j \Omega_\Psi \hat{\boldsymbol{\Theta}}_j^\top$ and $\Omega_\Psi := \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i \boldsymbol{\psi}_i^2 / n$.

□

Proof of Lemma 12. Assume $f_\beta(x) = a > 0$. Let the distribution function of error at x be

$$v_0(t|\mathbf{x}) = \mathbb{P}(\varepsilon \leq t|\mathbf{x}).$$

$$\begin{aligned}
\mathcal{P}\rho_f|\mathbf{x} &= \mathbb{E}[w\rho_\tau(y-a) + (1-w)\rho_\tau(y^{-\infty}-a)|\mathbf{x}] \\
&= \int_a^\infty [w(t)\rho_\tau(t-a) + (1-w(t))\rho_\tau(y^{-\infty}-a)] dF_0(t|\mathbf{x}) \\
&\quad + \int_0^a [w(t)\rho_\tau(t-a) + (1-w(t))\rho_\tau(y^{-\infty}-a)] dF_0(t|\mathbf{x}) \\
&\quad + \int_{-\infty}^0 [w(t)\rho_\tau(t-a) + (1-w(t))\rho_\tau(y^{-\infty}-a)] dF_0(t|\mathbf{x}) \\
&= \int_a^\infty \tau(t-a)dF_0(t|\mathbf{x}) + \int_0^a (\tau-1)(t-a)dF_0(t|\mathbf{x}) \\
&\quad + \int_{-\infty}^0 \left[\left(1 - \frac{\tau}{F_0(0|\mathbf{x})}\right)(\tau-1)(t-a) + \frac{\tau}{F_0(0|\mathbf{x})}(\tau-1)(y^{-\infty}-a) \right] dF_0(t|\mathbf{x}) \\
&= \tau \int_0^\infty t dF_0 - \int_0^a t dF_0 + (\tau F_0(0|\mathbf{x}) - \tau - F_0(0|\mathbf{x}))a + aF_0(a|\mathbf{x}) \\
&\quad + \int_{-\infty}^0 \left[\left(1 - \frac{\tau}{F_0(0|\mathbf{x})}\right)(\tau-1)(t-a) + \frac{\tau}{F_0(0|\mathbf{x})}(\tau-1)(y^{-\infty}-a) \right] dF_0(t|\mathbf{x}).
\end{aligned}$$

$$\begin{aligned}
\mathcal{P}\rho_{f_0}|\mathbf{x} &= \mathbb{E}[w\rho_\tau(y - \mathbf{x}\beta^o) + (1-w)\rho_\tau(y^{-\infty} - \mathbf{x}\beta^o)|\mathbf{x}] \\
&= \tau \int_0^\infty t dF_0 - \int_0^{\mathbf{x}\beta^o} t dF_0 + (\tau F_0(0|\mathbf{x}) - \tau - F_0(0|\mathbf{x}))\mathbf{x}\beta^o + \mathbf{x}\beta^o F_0(\mathbf{x}\beta^o|\mathbf{x}) \\
&\quad + \int_{-\infty}^0 \left[\left(1 - \frac{\tau}{F_0(0|\mathbf{x})}\right)(\tau-1)(t - \mathbf{x}\beta^o) + \frac{\tau}{F_0(0|\mathbf{x})}(\tau-1)(y^{-\infty} - \mathbf{x}\beta^o) \right] dF_0(t|\mathbf{x}).
\end{aligned}$$

$$\begin{aligned}
\mathcal{P}\rho_f|\mathbf{x} - \mathcal{P}\rho_{f_0}|\mathbf{x} &= -\int_{\mathbf{x}\beta^o}^a t dF_0 + (\tau F_0(0|\mathbf{x}) - \tau - F_0(0|\mathbf{x}))(a - \mathbf{x}\beta^o) + aF_0(a|\mathbf{x}) - \tau\mathbf{x}\beta^o \\
&\quad + \int_{-\infty}^0 \left[\left(1 - \frac{\tau}{F_0(0|\mathbf{x})}\right)(\tau - 1)(\mathbf{x}\beta^o - a) \right. \\
&\quad \left. + \frac{\tau}{F_0(0|\mathbf{x})}(\tau - 1)(\mathbf{x}\beta^o - a) \right] dF_0(t|\mathbf{x}) \\
&= -\int_{\mathbf{x}\beta^o}^a t dF_0 + (\tau F_0(0|\mathbf{x}) - \tau - F_0(0|\mathbf{x}))(a - \mathbf{x}\beta^o) + aF_0(a|\mathbf{x}) - \tau\mathbf{x}\beta^o \\
&\quad + (\mathbf{x}\beta^o - a)(\tau - 1) \int_{-\infty}^0 \left[\left(1 - \frac{\tau}{F_0(0|\mathbf{x})}\right) + \frac{\tau}{F_0(0|\mathbf{x})} \right] dF_0(t|\mathbf{x}) \\
&= -\int_{\mathbf{x}\beta^o}^a t dF_0 + (\tau F_0(0|\mathbf{x}) - \tau - F_0(0|\mathbf{x}))(a - \mathbf{x}\beta^o) + aF_0(a|\mathbf{x}) - \tau\mathbf{x}\beta^o \\
&\quad + (\mathbf{x}\beta^o - a)(\tau - 1)F_0(0|\mathbf{x}) \\
&= -\int_0^{a-\mathbf{x}\beta^o} (t + \mathbf{x}\beta^o) d\nu_0(t|\mathbf{x}) \\
&\quad + (\tau\nu_0(-\mathbf{x}\beta^o|\mathbf{x}) - \tau - \nu_0(-\mathbf{x}\beta^o|\mathbf{x}))(a - \mathbf{x}\beta^o) \\
&\quad + a\nu_0(a - \mathbf{x}\beta^o|\mathbf{x}) - \tau\mathbf{x}\beta^o + (\mathbf{x}\beta^o - a)(\tau - 1)\nu_0(-\mathbf{x}\beta^o|\mathbf{x}) \\
&= -\int_0^{a-\mathbf{x}\beta^o} t d\nu_0(t|\mathbf{x}) + (a - \mathbf{x}\beta^o)(\nu_0(a - \mathbf{x}\beta^o|\mathbf{x}) - \tau). \quad (4.30)
\end{aligned}$$

Let $z := a - \mathbf{x}\beta^o$, then:

$$\begin{aligned}
(4.30) &= -\int_0^z t d\nu_0(t|\mathbf{x}) + z(\nu_0(z|\mathbf{x}) - \tau) \\
&= -\int_0^z t d\nu_0(t|\mathbf{x}) + \int_0^z z d\nu_0(t|\mathbf{x}) \\
&= \int_0^z (z-t) d\nu_0(t|\mathbf{x}) \\
&= \int_0^z (z-t) \dot{\nu}_0(t|\mathbf{x}) dt \\
&= \int_0^z (z-t) \dot{\nu}_0(0|\mathbf{x}) dt + \int_0^z (z-t) (\dot{\nu}_0(t|\mathbf{x}) - \dot{\nu}_0(0|\mathbf{x})) dt \\
&\geq \int_0^z (z-t) \dot{\nu}_0(0|\mathbf{x}) dt - \int_0^{|z|} (|z|-t) |\dot{\nu}_0(t|\mathbf{x}) - \dot{\nu}_0(0|\mathbf{x})| dt \\
&\stackrel{(i)}{\geq} \int_0^z (z-t) \dot{\nu}_0(0|\mathbf{x}) dt - L \int_0^{|z|} (|z|-t) t dt \\
&= \frac{1}{2} \dot{\nu}_0(0|\mathbf{x}) z^2 - \frac{1}{6} L |z|^3. \tag{4.31}
\end{aligned}$$

In (i), we use the Lipschitz condition of the density function of error. Because of (4.31) and Condition 11, we can then use the Lemma in Stadler (2010) to conclude that there exists $C_1 > 0$ s.t. $\mathcal{E}(f_\beta) \geq C_1^2 \|f_\beta - f_0\|^2$. \square

Proof of Lemma 13.

$$\begin{aligned}
|\gamma_\beta(y, \mathbf{x})| &= |w\rho_\tau(y - \mathbf{x}\beta) + (1-w)\rho_\tau(y^{-\infty} - \mathbf{x}\beta) - w\rho_\tau(y - \mathbf{x}\beta^o) - (1-w)\rho_\tau(y^{-\infty} - \mathbf{x}\beta^o)| \\
&= |w\rho_\tau(y - \mathbf{x}\beta) - w\rho_\tau(y - \mathbf{x}\beta^o) + (1-w)\rho_\tau(y^{-\infty} - \mathbf{x}\beta) - (1-w)\rho_\tau(y^{-\infty} - \mathbf{x}\beta^o)| \\
&= |w(\rho_\tau(y - \mathbf{x}\beta) - \rho_\tau(y - \mathbf{x}\beta^o)) + (1-w)(\tau - 1)\mathbf{x}(\beta^o - \beta)| \\
&\leq_{(i)} w|\max(\tau, 1 - \tau)\mathbf{x}(\beta - \beta^o)| + (1-w)|(\tau - 1)\mathbf{x}(\beta^o - \beta)| \\
&= \{w\max(\tau, 1 - \tau) + (1-w)(1 - \tau)\} |\mathbf{x}(\beta - \beta^o)| \\
&\leq \max(\tau, 1 - \tau) |\mathbf{x}(\beta - \beta^o)| \\
&\leq \max(\tau, 1 - \tau) \|\mathbf{x}\|_\infty \|\beta - \beta^o\|_1 \\
&\leq_{(ii)} \max(\tau, 1 - \tau) K_X \|\beta - \beta^o\|_1.
\end{aligned}$$

for all x, y, β in the range. The inequality (i) is from triangle inequality and property of loss function ρ_τ , and (ii) is because of Condition 12. Therefore, we have

$$|\gamma_\beta(y_i, \mathbf{x}_i) - \mathbb{E}\gamma_\beta(y_i, \mathbf{x}_i)| \leq 2 \max(\tau, 1 - \tau) \|\beta - \beta^o\|_1 K_X.$$

Denote $c_{i,\beta} := 2 \max(\tau, 1 - \tau) \|\beta - \beta^o\|_1 K_X$, it is easy to show that

$$\sup_{\|\beta - \beta^o\|_1 \leq M} \sum_{i=1}^n c_{i,\beta}^2 \leq (4 \max(\tau, 1 - \tau)^2 M^2 K_X^2) n \leq 4M^2 K_X^2 n.$$

By the concentration theorem (Massart, 2000), we have

$$\mathbb{P}(Z_M \geq \mathbb{E}Z_M + t) \leq \exp\left(-\frac{nt^2}{32M^2 K_X^2}\right).$$

Therefore,

$$\mathbb{P}\left(Z_M \geq \mathbb{E}Z_M + MK_X \sqrt{\frac{32t}{n}}\right) \leq e^{-t}.$$

By the contraction inequality (Lemma 14.20 in Buhlmann and van de Geer (2011)), we have

$$\mathbb{E}Z_M \leq 4MK_X \sqrt{\frac{2\log(2p)}{n}}.$$

Consequently, for all $t > 0$ and $M > 0$,

$$\mathbb{P}\left(Z_M \geq 4MK_X \sqrt{\frac{2\log(2p)}{n}} + MK_X \sqrt{\frac{32t}{n}}\right) \leq e^{-t}.$$

Let

$$\lambda(t) = 4K_X \sqrt{\frac{2\log(2p)}{n}} + K_X \sqrt{\frac{32t}{n}}, \quad (4.32)$$

we have

$$\mathbb{P}(Z_M \geq M\lambda(t)) \leq e^{-t}.$$

□

4.7 Proofs of Theorems

Proof of Theorem 10.

Lemma 12. *Assuming Conditions 10 and 13, there exists some constant C_1 such that*

$$\mathcal{E}(f_\beta) \geq C_1^2 \|f_\beta - f_0\|^2.$$

Lemma 13 (Concentration inequality). *Define*

$$\gamma_\beta(y, \mathbf{x}) := \rho_{f_\beta}(y, \mathbf{x}, w) - \rho_{f_{\beta^0}}(y, \mathbf{x}, w),$$

$$Z_M := \sup_{\|\beta - \beta^0\|_1 \leq M} \left| \frac{1}{n} \sum_{i=1}^n \gamma_\beta(y_i, \mathbf{x}_i) - \mathbb{E} \gamma_\beta(y_i, \mathbf{x}_i) \right|,$$

$$\lambda(t) := 4K_X \sqrt{\frac{2 \log(2p)}{n}} + K_X \sqrt{\frac{32t}{n}}.$$

Then we have

$$\mathbb{P}(Z_M \geq M\lambda(t)) \leq e^{-t}.$$

The following argument follows Muller and van der Geer (2014). We start with bounding

the excess risk for $f_{\hat{\beta}}$,

$$\begin{aligned}\mathcal{E}(f_{\hat{\beta}}) &= \mathcal{P}\rho_{f_{\hat{\beta}}} - \mathcal{P}\rho_{f_0} \\ &= -(\mathcal{P}_n - \mathcal{P})(\rho_{f_{\hat{\beta}}} - \rho_{f_0})\end{aligned}\tag{4.33}$$

$$+ \hat{\mathcal{P}}_n(\rho_{f_{\hat{\beta}}}) + \lambda\|\hat{\beta}\|_1 - \left(\hat{\mathcal{P}}_n(\rho_{f_0}) + \lambda\|\beta^o\|_1\right)\tag{4.34}$$

$$+ \lambda\|\beta^o\|_1 - \lambda\|\hat{\beta}\|_1\tag{4.35}$$

$$+ \mathcal{P}_n(\rho_{f_{\hat{\beta}}}) - \hat{\mathcal{P}}_n(\rho_{f_{\hat{\beta}}}) + \mathcal{P}_n(\rho_{f_0}) - \hat{\mathcal{P}}_n(\rho_{f_0}).\tag{4.36}$$

The plan is that, for equation (4.33), the empirical process part, we bound the term using concentration inequality. While equation (4.34) is negative by the definition of $\hat{\beta}$, equation (4.35) can be bounded using triangular inequality. Finally, for equation (4.36), it is negligible because $\|w^0 - \hat{w}\|_\infty = o_p(1)$, which is shown in the proof of Lemma 7.

We then bound (4.33), (4.34), (4.35) separately. For (4.35), it is easy to show:

$$\lambda\|\beta^o\|_1 - \lambda\|\hat{\beta}\|_1 \leq \lambda \sum_{j \in S(\beta^o)} |\hat{\beta}_j - \beta_j^o| - \lambda \sum_{j \in S^c(\beta^o)} |\hat{\beta}_j|.$$

For (4.33), we have

$$-(\mathcal{P}_n - \mathcal{P})(\rho_{f_{\hat{\beta}}} - \rho_{f_0}) = -(\mathcal{P}_n - \mathcal{P})\gamma_{\hat{\beta}},$$

and

$$Z_M = \sup_{\|\beta - \beta^o\|_1 \leq M} |(\mathcal{P}_n - \mathcal{P})\gamma_\beta|.$$

Now define

$$Z_M^\delta := \sup_{\|\beta - \beta^o\|_1 \leq M} \frac{|(\mathcal{P}_n - \mathcal{P})\gamma_\beta|}{\|\beta - \beta^o\|_1 \vee \delta}.$$

We have

$$\begin{aligned}
\mathbb{P}(Z_M^\delta > 2\lambda(t)) &= \mathbb{P}\left(\sup_{\|\beta - \beta^o\|_1 \leq M} \frac{|(\mathcal{P}_n - \mathcal{P})\gamma_\beta|}{\|\beta - \beta^o\|_1 \vee \delta} > 2\lambda(t)\right) \\
&\leq \sum_{j=\lfloor -\log_2 M \rfloor}^{\lceil -\log_2 \delta - 1 \rceil} \mathbb{P}\left(\sup_{2^{-j-1} \leq \|\beta - \beta^o\|_1 \leq 2^{-j}} \frac{|(\mathcal{P}_n - \mathcal{P})\gamma_\beta|}{\|\beta - \beta^o\|_1 \vee \delta} > 2\lambda(t)\right) \\
&\quad + \mathbb{P}\left(\sup_{\|\beta - \beta^o\|_1 \leq \delta} \frac{|(\mathcal{P}_n - \mathcal{P})\gamma_\beta|}{\|\beta - \beta^o\|_1 \vee \delta} > 2\lambda(t)\right) \\
&\leq \sum_{j=\lfloor -\log_2 M \rfloor}^{\lceil -\log_2 \delta - 1 \rceil} \mathbb{P}\left(\sup_{2^{-j-1} \leq \|\beta - \beta^o\|_1 \leq 2^{-j}} \frac{|(\mathcal{P}_n - \mathcal{P})\gamma_\beta|}{2^{-j-1}} > 2\lambda(t)\right) \\
&\quad + \mathbb{P}\left(\sup_{\|\beta - \beta^o\|_1 \leq \delta} \frac{|(\mathcal{P}_n - \mathcal{P})\gamma_\beta|}{\delta} > 2\lambda(t)\right) \\
&= \sum_{j=\lfloor -\log_2 M \rfloor}^{\lceil -\log_2 \delta - 1 \rceil} \mathbb{P}\left(\sup_{2^{-j-1} \leq \|\beta - \beta^o\|_1 \leq 2^{-j}} |(\mathcal{P}_n - \mathcal{P})\gamma_\beta| > 2^{-j}\lambda(t)\right) \\
&\quad + \mathbb{P}\left(\sup_{\|\beta - \beta^o\|_1 \leq \delta} |(\mathcal{P}_n - \mathcal{P})\gamma_\beta| > 2\delta\lambda(t)\right) \\
&\leq \sum_{j=\lfloor -\log_2 M \rfloor}^{\lceil -\log_2 \delta - 1 \rceil} \mathbb{P}(Z_{2^{-j}} > 2^{-j}\lambda(t)) + e^{-t} \\
&\leq \sum_{j=\lfloor -\log_2 M \rfloor}^{\lceil -\log_2 \delta - 1 \rceil} e^{-t} + e^{-t} \\
&= (\lceil -\log_2 \delta - 1 \rceil - \lfloor -\log_2 M \rfloor + 2) e^{-t} \\
&= (\lceil \log_2 M \rceil - \lfloor \log_2 \delta + 1 \rfloor + 2) e^{-t} \\
&\leq (\lceil \log_2 M \rceil - \lceil \log_2 \delta \rceil + 2) e^{-t} \\
&\leq \log_2 \left(\frac{8M}{\delta} \right) e^{-t}.
\end{aligned}$$

Therefore, for any β with $\|\beta - \beta^o\|_1 \leq M$, we have

$$|(\mathcal{P}_n - \mathcal{P})\gamma_\beta| \leq 2\lambda(t) (\|\beta - \beta^o\|_1 \vee \delta)$$

with probability at least $1 - \log_2\left(\frac{8M}{8}\right) e^{-t}$.

It is easy to show that $\|\hat{\beta} - \beta^o\|_1 \ll n$. Then let $\delta = p^{-2}$, $t = 2\log(p)$ we have

$$|(\mathcal{P}_n - \mathcal{P})\gamma_{\hat{\beta}}| \leq 2\lambda(t) (\|\hat{\beta} - \beta^o\|_1 \vee p^{-2})$$

with probability at least $1 - \log_2(8np^2)/p^2$.

If $\|\hat{\beta} - \beta^o\|_1 \leq p^{-2}$, trivially we have consistency.

If $\|\hat{\beta} - \beta^o\|_1 > p^{-2}$, then because (4.34) is always non-positive by the definition of $\hat{\beta}$, we have

$$\begin{aligned} \mathcal{E}(f_{\hat{\beta}}) &\leq -(\mathcal{P}_n - \mathcal{P})(\rho_{f_{\hat{\beta}}} - \rho_{f_0}) + \lambda\|\beta^o\|_1 - \lambda\|\hat{\beta}\|_1 \\ &\leq 2\lambda(t)\|\hat{\beta} - \beta^o\|_1 + \lambda\|\beta^o\|_1 - \lambda\|\hat{\beta}\|_1 \\ &= 2\lambda(t) \left(\sum_{j \in \mathcal{S}(\beta^o)} |\hat{\beta}_j - \beta_j^o| + \sum_{j \in \mathcal{S}^c(\beta^o)} |\hat{\beta}_j| \right) \\ &\quad + \lambda \left(\sum_{j \in \mathcal{S}(\beta^o)} |\beta_j^o| - \sum_{j \in \mathcal{S}(\beta^o)} |\hat{\beta}_j| - \sum_{j \in \mathcal{S}^c(\beta^o)} |\hat{\beta}_j| \right) \\ &\leq 2\lambda(t) \left(\sum_{j \in \mathcal{S}(\beta^o)} |\hat{\beta}_j - \beta_j^o| + \sum_{j \in \mathcal{S}^c(\beta^o)} |\hat{\beta}_j| \right) + \lambda \left(\sum_{j \in \mathcal{S}(\beta^o)} |\hat{\beta}_j - \beta_j^o| - \sum_{j \in \mathcal{S}^c(\beta^o)} |\hat{\beta}_j| \right) \\ &= (2\lambda(t) + \lambda) \sum_{j \in \mathcal{S}(\beta^o)} |\hat{\beta}_j - \beta_j^o| + (2\lambda(t) - \lambda) \sum_{j \in \mathcal{S}^c(\beta^o)} |\hat{\beta}_j|. \end{aligned} \quad (4.37)$$

Since $\mathcal{E}(f_{\hat{\beta}}) \geq 0$ and $\lambda \geq 4\lambda(t)$, from (4.37), we know

$$\|\hat{\beta}_{\mathcal{S}^c}\|_1 \leq \frac{\lambda + 2\lambda(t)}{\lambda - 2\lambda(t)} \|(\hat{\beta} - \beta^o)_{\mathcal{S}^c}\|_1 \leq 3\|(\hat{\beta} - \beta^o)_{\mathcal{S}^c}\|_1 \quad (4.38)$$

which allows us to use the compatibility and censoring conditions. And again by (4.37) and $\lambda \geq 4\lambda(t)$, we have

$$\mathcal{E}(f_{\hat{\beta}}) \leq (2\lambda(t) + \lambda) \sum_{j \in \mathcal{S}(\beta^o)} |\hat{\beta}_j - \beta_j^o|. \quad (4.39)$$

By Lemma 12, equation (4.39), the censoring condition and the compatibility condition,

we have

$$\begin{aligned}
(2\lambda(t) + \lambda) \|(\hat{\beta} - \beta^o)_{S_o}\|_1 &\geq C_1^2 \|f_{\hat{\beta}} - f_0\|_2^2 \\
&= C_1^2 (\hat{\beta} - \beta^o)^T \mathbb{E}[\mathbf{x}^T \mathbf{x}] (\hat{\beta} - \beta^o) \\
&\geq_{(i)} C_1^2 \frac{\phi_0^2}{s_{\beta^o}} \|(\hat{\beta} - \beta^o)_{S_o}\|_1^2
\end{aligned} \tag{4.40}$$

where (i) is from the compatibility condition.

By (4.40),

$$\|(\hat{\beta} - \beta^o)_{S_o}\|_1 \leq \frac{s_{\beta^o} (2\lambda(t) + \lambda)}{C_1^2 \phi_0^2}. \tag{4.41}$$

Equation (4.38) implies that $\|\hat{\beta} - \beta^o\|_1 \leq 4 \|(\hat{\beta} - \beta^o)_{S_o}\|_1$, and hence by (4.41),

$$\begin{aligned}
\|\hat{\beta} - \beta^o\|_1 &\leq \frac{4s_{\beta^o} (2\lambda(t) + \lambda)}{C_1^2 \phi_0^2} \\
&\leq \frac{6\lambda s_{\beta^o}}{C_1^2 \phi_0^2}.
\end{aligned} \tag{4.42}$$

With $C = 1/C_1^2$, we have Theorem 10. Furthermore, by (4.40), we have

$$(\hat{\beta} - \beta^o)^T \mathbb{E}[\mathbf{x}^T \mathbf{x}] (\hat{\beta} - \beta^o) \leq \frac{3\lambda C}{2} \|\hat{\beta} - \beta^o\|_1. \tag{4.43}$$

□

Proof of Theorem 11. Following results from Lemmas 7 - 10, when $\left\| \hat{\Theta}_{\hat{\beta},j} - \Theta_{\beta^o,j} \right\|_1 = o_p(1)$,

the representation (4.17) can be simplified as

$$\begin{aligned}
\sqrt{n}(\tilde{\beta}_j - \beta_j^o) &= \sqrt{n} \left(\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \Psi_i + \frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top \Phi_i \right) \\
&+ O_p \left(K \lambda_j \lambda s_{\beta^o} \sqrt{n} + K \lambda^2 s_{\beta^o}^2 \sqrt{n} + \sqrt{\lambda s_{\beta^o} s_j} + \frac{K}{\sqrt{n}} + K \frac{\log^{3/4} n}{\sqrt{n}} \right) \\
&= \sqrt{n} \left(\frac{1}{n} \sum_{i=1}^n \hat{\Theta}_j \mathbf{x}_i^\top (\Psi_i + \Phi_i) \right) + o_p(1).
\end{aligned}$$

The last line follows from assuming both λ and λ_j are of order $O(\sqrt{\log p/n})$, and $K s_{\beta^o}^2 \log p/n \vee s_{\beta^o}^{1/2} s_j^{1/2} (\log p/n)^{1/4} = o(1)$. Then we have that

$$\sqrt{n}(\tilde{\beta}_j - \beta_j^o) \xrightarrow{d} \mathcal{N}(0, \sigma_j^2),$$

where $\sigma_j^2 = \mathbb{E} \hat{\Theta}_j \Omega \hat{\Theta}_j^\top$ and $\Omega := \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i (\Psi_i + \Phi_i)^2 / n$.

The only missing part of the proof is the bound on the estimation error for $\hat{\sigma}_j^2 := \hat{\Theta}_j \hat{\Omega} \hat{\Theta}_j^\top$ from $\mathbb{E} \hat{\Theta}_j \Omega \hat{\Theta}_j^\top$, where $\hat{\Omega} = \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i (\hat{\Psi}_i + \hat{\Phi}_i)^2 / n$. We start with rewriting the estimation error,

$$\begin{aligned}
|\hat{\sigma}_j^2 - \sigma_j^2| &= \underbrace{\left| \hat{\Theta}_j \hat{\Omega} \hat{\Theta}_j^\top - \Theta_j^0 \hat{\Omega} \Theta_j^{0,\top} \right|}_{T_1} + \underbrace{\left| \Theta_j^0 \hat{\Omega} \Theta_j^{0,\top} - \Theta_j^0 \Omega \Theta_j^{0,\top} \right|}_{T_2} \\
&+ \underbrace{\left| \Theta_j^0 \Omega \Theta_j^{0,\top} - \Theta_j^0 \mathbb{E} \Omega \Theta_j^{0,\top} \right|}_{T_3} + \underbrace{\left| \mathbb{E} \left(\Theta_j^0 \Omega \Theta_j^{0,\top} - \hat{\Theta}_j \Omega \hat{\Theta}_j^\top \right) \right|}_{T_4}
\end{aligned}$$

For the term T_1 , we can further decompose it as

$$\begin{aligned}
T_1 &\leq \left| (\Theta_j^0 - \hat{\Theta}_j) \Omega \Theta_j^{0,\top} \right| + \left| \hat{\Theta}_j \Omega (\Theta_j^{0,\top} - \hat{\Theta}_j^\top) \right| \\
&\leq 2 \left| \Theta_j^0 \Omega (\Theta_j^0 - \hat{\Theta}_j)^\top \right| + \left| (\Theta_j^0 - \hat{\Theta}_j) \Omega (\Theta_j^0 - \hat{\Theta}_j)^\top \right| \\
&\leq 2 \|\Theta_j^0 \Omega\|_\infty \|\hat{\Theta}_j - \Theta_j^0\|_1 + \|\Omega\|_\infty \|\hat{\Theta}_j - \Theta_j^0\|_1^2.
\end{aligned}$$

Because $\left| \Theta_j^0 \mathbf{x}_i^\top \right| = O(K)$ and $\|\mathbf{x}_i\|_\infty = O(1)$, we know $\|\Theta_j^0 \Omega\|_\infty = O(K)$ and $\|\Omega\|_\infty = O(1)$. Therefore, $T_1 = o_p(1)$ if $K \|\hat{\Theta}_j - \Theta_j^0\|_1 = o_p(1)$. We note that term T_4 can be bounded similarly.

For the term $T_2 + T_3$, denote $\hat{\xi}_i = \hat{\psi}_i + \hat{\phi}_i$, then

$$T_2 + T_3 = \underbrace{\left| \Theta_j^0 \left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i (\hat{\xi}_i^2 - \xi_i^2) \right) \Theta_j^{0,\top} \right|}_{T_2} + \underbrace{\left| \Theta_j^0 \left(\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i^\top \mathbf{x}_i \xi_i^2 - \mathbb{E} \mathbf{x}_i^\top \mathbf{x}_i \xi_i^2) \right) \Theta_j^{0,\top} \right|}_{T_3}.$$

For term T_3 , since $\|\mathbf{x}\|_\infty = O(1)$ and $|\xi_i| \leq 1$, by Hoeffding's inequality, we have

$$\frac{1}{n} \sum_{i=1}^n \left(\Theta_j^0 \mathbf{x}_i^\top \mathbf{x}_i \Theta_j^{0,\top} \xi_i^2 - \mathbb{E} \Theta_j^0 \mathbf{x}_i^\top \mathbf{x}_i \Theta_j^{0,\top} \xi_i^2 \right) = O_p \left(\frac{K^2}{\sqrt{n}} \right).$$

Next, note that for T_2

$$\begin{aligned} \left| \frac{1}{n} \sum_{i=1}^n \hat{\xi}_i^2 - \xi_i^2 \right| &= \left| \frac{1}{n} \sum_{i=1}^n (\hat{\xi}_i + \xi_i)(\hat{\xi}_i - \xi_i) \right| \\ &= \left| \frac{1}{n} \sum_{i=1}^n (\hat{\psi}_i + \hat{\phi}_i + \psi_i + \phi_i) (\hat{\psi}_i + \hat{\phi}_i - \psi_i - \phi_i) \right| \\ &\leq 4 \left(\left| \frac{1}{n} \sum_{i=1}^n (\hat{\psi}_i - \psi_i) \right| + \left| \frac{1}{n} \sum_{i=1}^n (\hat{\phi}_i - \phi_i) \right| \right). \end{aligned}$$

For the first difference, we have

$$\begin{aligned} \left| \frac{1}{n} \sum_{i=1}^n (\hat{\psi}_i - \psi_i) \right| &= \left| \frac{1}{n} \sum_{i=1}^n (w_i(\hat{F}_n) \mathbb{1}(Y_i - \mathbf{x}_i \hat{\beta} \geq 0) - w_i(F_0) \mathbb{1}(Y_i - \mathbf{x}_i \beta^o \geq 0)) \right| \\ &\leq \left| \frac{1}{n} \sum_{i=1}^n (w_i(\hat{F}_n) - w_i(F_0)) \right| + \left| \frac{1}{n} \sum_{i=1}^n (\mathbb{1}(Y_i - \mathbf{x}_i \hat{\beta} \geq 0) - \mathbb{1}(Y_i - \mathbf{x}_i \beta^o \geq 0)) \right| \\ &= O_p(1/\sqrt{n}) + O_p(\lambda s \beta^o), \end{aligned}$$

following results in Lemma 7. In addition,

$$\begin{aligned}
|\hat{\phi}_i - \phi_i| &\leq \tau \left| \frac{1}{n} \sum_{l=1, l \neq i}^n \frac{\mathbb{1}(\hat{F}_n > \tau)}{\hat{F}_n^2} \left(1 - \frac{\mathbb{1}(Y_l=0)}{\hat{F}_n} \right) - \frac{1}{n} \sum_{l=1, l \neq i}^n \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} \left(1 - \frac{\mathbb{1}(Y_l=0)}{F_0} \right) \right| \\
&= \frac{\tau}{n} \left| \sum_{l=1, l \neq i}^n \left(\frac{\mathbb{1}(\hat{F}_n > \tau)}{\hat{F}_n^2} - \frac{\mathbb{1}(F_0 > \tau)}{F_0^2} \right) \right. \\
&\quad \left. + \sum_{l=1, l \neq i}^n \left(\frac{\mathbb{1}(F_0 > \tau) \mathbb{1}(Y_l=0)}{F_0^3} - \frac{\mathbb{1}(\hat{F}_n > \tau) \mathbb{1}(Y_l=0)}{\hat{F}_n^3} \right) \right| \\
&= O_p(1/\sqrt{n}),
\end{aligned}$$

which then gives that $\left[\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}_i (\hat{\xi}_i^2 - \xi_i^2) \right]_{j,k} = O_p(1/\sqrt{n} + \lambda s_{\beta^o})$. Then we conclude that $T_3 = O_p(K^2/\sqrt{n} + K^2 \lambda s_{\beta^o})$.

Finally, when $K s_{\beta^o}^2 \log p/n \vee s_{\beta^o}^{1/2} s_j^{1/2} (\log p/n)^{1/4} \vee K \|\hat{\Theta}_j - \Theta_j^0\|_1 \vee K^2/\sqrt{n} \vee K^2 \lambda s_{\beta^o} = o(1)$, we have that $\hat{\sigma}_j = \sigma_j + o(1)$, which then completes the proof. \square

4.8 Acknowledgement

Chapter 4, in full, is currently being prepared for submission for publication of the material. Li, Hanbo; Guo, Jiaqi; Bradic, Jelena. High-dimensional covariate effects on left-censored quantile event times. The dissertation/thesis author is one of the principal investigators and author of this material.

Bibliography

- [Akr94] Michael G Akritas. Nearest neighbor estimation of a bivariate distribution under random censoring. *The Annals of Statistics*, pages 1299–1327, 1994.
- [ATHW14] Sunil Aryal, Kai Ming Ting, Gholamreza Haffari, and Takashi Washio. mp-dissimilarity: A data dependent dissimilarity measure. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 707–712. IEEE, 2014.
- [ATW16] Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *arXiv preprint arXiv:1610.01271*, 2016.
- [BB06] MAJ John R Brence and Donald E Brown. Improving the robust random forest regression algorithm. *Systems and Information Engineering Technical Papers, Department of Systems and Information Engineering, University of Virginia*, 2006.
- [Ber81] Rudolf Beran. Nonparametric regression with randomly censored survival data. Technical report, Technical Report, Univ. California, Berkeley, 1981.
- [BG16] Jelena Bradic and Jiaqi Guo. Robust confidence intervals in high-dimensional left-censored regression. *arXiv preprint arXiv:1609.07165*, 2016.
- [BH98] Moshe Buchinsky and Jinyong Hahn. An alternative estimator for the censored quantile regression model. *Econometrica*, pages 653–671, 1998.
- [BIM86] A Ben-Israel and B Mond. What is invexity? *The ANZIAM Journal*, 28(1):1–9, 1986.
- [BJM06] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.
- [BK13] Jakramate Bootkrajang and Ata Kabán. Boosting in the presence of label noise. *arXiv preprint arXiv:1309.6818*, 2013.
- [Bre96] Leo Breiman. Arcing classifiers. *Annals of Statistics*, 26, 1996.

- [Bre99] Leo Breiman. Prediction games and arcing algorithms. *Neural computation*, 11(7):1493–1517, 1999.
- [Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [Bre04] Leo Breiman. Population theory for boosting ensembles. *The Annals of Statistics*, 32(1):1–11, 2004.
- [BRT09] Peter J Bickel, Ya’acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.
- [BT07] Peter L Bartlett and Mikhail Traskin. Adaboost is consistent. *Journal of Machine Learning Research*, 8(Oct):2347–2368, 2007.
- [CBFAB97] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on image processing*, 6(2):298–311, 1997.
- [CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.
- [CSS02] Michael Collins, Robert E Schapire, and Yoram Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3):253–285, 2002.
- [Dab87] Dorota M Dabrowska. Non-parametric regression with censored survival time data. *Scandinavian Journal of Statistics*, pages 181–197, 1987.
- [Dab89] Dorota M Dabrowska. Uniform consistency of the kernel conditional kaplan-meier estimate. *The Annals of Statistics*, pages 1157–1167, 1989.
- [DG05] P Laurie Davies and Ursula Gather. Breakdown and groups. *The Annals of Statistics*, 33(3):977–1035, 2005.
- [DH83] David L Donoho and Peter J Huber. The notion of breakdown point. *A festschrift for Erich L. Lehmann*, 157184, 1983.
- [DKES06] Sorin Draghici, Purvesh Khatri, Aron C Eklund, and Zoltan Szallasi. Reliability and reproducibility issues in dna microarray measurements. *TRENDS in Genetics*, 22(2):101–109, 2006.
- [DM13] Amit G Deshwar and Quaid Morris. Plida: cross-platform gene expression normalization using perturbed topic models. *Bioinformatics*, 30(7):956–961, 2013.
- [DW00] Carlos Domingo and Osamu Watanabe. Madaboost: A modification of adaboost. In *COLT*, pages 180–189, 2000.

- [Efr67] Bradley Efron. The two sample problem with censored data. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 4, pages 831–853, 1967.
- [Efr07] Sam Efromovich. Conditional density estimation in a regression setting. *The Annals of Statistics*, pages 2504–2535, 2007.
- [FG96] Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, volume 66. CRC Press, 1996.
- [FHT00] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.
- [FHT01] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.
- [Fit97] Bernd Fitzenberger. Computational aspects of censored quantile regression. *Lecture Notes-Monograph Series*, pages 171–186, 1997.
- [Fre95] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.
- [Fre01] Yoav Freund. An adaptive version of the boost by majority algorithm. *Machine learning*, 43(3):293–318, 2001.
- [Fre09] Yoav Freund. A more robust boosting algorithm. *arXiv preprint arXiv:0905.2138*, 2009.
- [Fri01] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [FS95] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.
- [FS96] Yoav Freund and Robert E Schapire. Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Citeseer, 1996.
- [FS97] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
- [FSA99] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

- [FW07] Bernd Fitzenberger and Peter Winker. Improving the computation of censored quantile regressions. *Computational Statistics & Data Analysis*, 52(1):88–108, 2007.
- [GD04] Peter D Grünwald and A Philip Dawid. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *the Annals of Statistics*, 32(4):1367–1433, 2004.
- [Gen03] Claudio Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003.
- [GMCS94] W Gonzalez-Manteiga and C Cadarso-Suarez. Asymptotic properties of a generalized kaplan-meier estimator with some applications. *Communications in Statistics-Theory and Methods*, 4(1):65–78, 1994.
- [GPS07] Giuliano Galimberti, Marilena Pillati, and Gabriele Soffritti. Robust regression trees based on m-estimators. *Statistica*, 67(2):173–190, 2007.
- [Ham68] Frank R Hampel. Contribution to the theory of robust estimation. *Ph. D. Thesis, University of California, Berkeley*, 1968.
- [Ham74] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.
- [HBD⁺05] Torsten Hothorn, Peter Bühlmann, Sandrine Dudoit, Annette Molinaro, and Mark J Van Der Laan. Survival ensembles. *Biostatistics*, 7(3):355–373, 2005.
- [HJCP⁺82] Frank E Harrell Jr, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.
- [HL93] Trevor Hastie and Clive Loader. Local regression: Automatic kernel carpentry. *Statistical Science*, pages 120–129, 1993.
- [HLBRT04] Torsten Hothorn, Berthold Lausen, Axel Benner, and Martin Radespiel-Tröger. Bagging survival trees. *Statistics in medicine*, 23(1):77–91, 2004.
- [HRRS11] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 114. John Wiley & Sons, 2011.
- [Hub64] Peter J Huber. Robust estimation of a location parameter. *The annals of mathematical statistics*, 35(1):73–101, 1964.
- [Hub11] Peter J Huber. *Robust statistics*. Springer, 2011.
- [HY05] Peter Hall and Qiwei Yao. Approximating conditional distribution functions using dimension reduction. *Annals of statistics*, pages 1404–1421, 2005.

- [HZ05] Patrick J Heagerty and Yingye Zheng. Survival model predictive accuracy and roc curves. *Biometrics*, 61(1):92–105, 2005.
- [IKBL08] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *The annals of applied statistics*, pages 841–860, 2008.
- [IL15] Rafael Izbicki and Ann B Lee. Nonparametric conditional density estimation in a high-dimensional regression setting. *Journal of Computational and Graphical Statistics*, (just-accepted), 2015.
- [JP73] Raymond Austin Jarvis and Edward A Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on computers*, 100(11):1025–1034, 1973.
- [KBJ78] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.
- [KBJ82] Roger Koenker and Gilbert Bassett Jr. Robust tests for heteroscedasticity based on regression quantiles. *Econometrica: Journal of the Econometric Society*, pages 43–61, 1982.
- [KG01] Roger Koenker and Olga Geling. Reappraising medfly longevity: a quantile regression survival analysis. *Journal of the American Statistical Association*, 96(454):458–468, 2001.
- [KL93] Michael Kearns and Ming Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.
- [KM58] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.
- [Koe05] Roger Koenker. *Quantile regression*. Number 38. Cambridge university press, 2005.
- [Koe08] Roger Koenker. Censored quantile regression redux. *Journal of Statistical Software*, 27(6):1–25, 2008.
- [KP96] Roger Koenker and Beum J Park. An interior point algorithm for nonlinear quantile regression. *Journal of Econometrics*, 71(1):265–283, 1996.
- [KP02] Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.
- [KS03] Adam Kalai and Rocco A Servedio. Boosting in the presence of noise. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 195–205. ACM, 2003.

- [KTEM07] Takafumi Kanamori, Takashi Takenouchi, Shinto Eguchi, and Noboru Murata. Robust loss functions for boosting. *Neural computation*, 19(8):2183–2244, 2007.
- [LB16] Alexander Hanbo Li and Jelena Bradic. Boosting in the presence of outliers: adaptive classification with non-convex loss functions. *Journal of the American Statistical Association*, (just-accepted), 2016.
- [LD95] Gang Li and Hani Doss. An approach to nonparametric regression for life history data using local linear fitting. *The Annals of Statistics*, pages 787–823, 1995.
- [Lic13] M. Lichman. UCI machine learning repository, 2013.
- [Lin98] Dekang Lin. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304. Citeseer, 1998.
- [Lit91] Nicholas Littlestone. Redundant noisy attributes, attribute errors, and linear-threshold learning using winnow. In *Proceedings of the fourth annual workshop on Computational learning theory*, pages 147–156. Morgan Kaufmann Publishers Inc., 1991.
- [LJ06] Yi Lin and Yongho Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474):578–590, 2006.
- [LKB08] Roman Werner Lutz, Markus Kalisch, and Peter Bühlmann. Robustified l2 boosting. *Computational Statistics & Data Analysis*, 52(7):3331–3341, 2008.
- [LM17] Alexander Hanbo Li and Andrew Martin. Forest-type regression with general losses and robust forest. In *International Conference on Machine Learning*, pages 2091–2100, 2017.
- [Loa06] Clive Loader. *Local regression and likelihood*. Springer Science & Business Media, 2006.
- [LS86] Shaw-Hwa Lo and Kesar Singh. The product-limit estimator and the bootstrap: some asymptotic representations. *Probability Theory and Related Fields*, 71(3):455–465, 1986.
- [LS10] Philip M Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. *Machine learning*, 78(3):287–304, 2010.
- [LZ08] Youjuan Li and Ji Zhu. L 1-norm quantile regression. *Journal of Computational and Graphical Statistics*, 17(1):163–185, 2008.
- [MBBF99a] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus Frean. Functional gradient techniques for combining hypotheses. *Advances in Neural Information Processing Systems*, pages 221–246, 1999.

- [MBBF99b] Llew Mason, Jonathan Baxter, Peter L Bartlett, and Marcus R Freen. Boosting algorithms as gradient descent. In *NIPS*, pages 512–518, 1999.
- [MCX⁺16] Qiguang Miao, Ying Cao, Ge Xia, Maoguo Gong, Jiachen Liu, and Jianfeng Song. Rboost: label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners. *IEEE transactions on neural networks and learning systems*, 27(11):2216–2228, 2016.
- [Mei06] Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun):983–999, 2006.
- [MG16] Waldyn Martinez and J Brian Gray. Noise peeling methods to improve boosting algorithms. *Computational Statistics & Data Analysis*, 93:483–497, 2016.
- [MO97] Richard Maclin and David Opitz. An empirical evaluation of bagging and boosting. *AAAI/IAAI*, 1997:546–551, 1997.
- [MSV09] Hamed Masnadi-Shirazi and Nuno Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *Advances in neural information processing systems*, pages 1049–1056, 2009.
- [MvdG16] Patric Müller and Sara van de Geer. Censored linear model in high dimensions. *Test*, 25(1):75–92, 2016.
- [NB04] John Ashworth Nelder and R Jacob Baker. Generalized linear models. *Encyclopedia of statistical sciences*, 4, 2004.
- [NDRT13] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.
- [New94] Whitney K Newey. Kernel estimation of partial means and a general variance estimator. *Econometric Theory*, 10(02):1–21, 1994.
- [NL02] Richard Nock and Patrice Lefaucheur. A robust boosting algorithm. In *European Conference on Machine Learning*, pages 319–331. Springer, 2002.
- [Obe82] Walter Oberhofer. The consistency of nonlinear regression minimizing the l_1 -norm. *The Annals of Statistics*, pages 316–319, 1982.
- [PH08] Limin Peng and Yijian Huang. Survival analysis with quantile regression models. *Journal of the American Statistical Association*, 103(482):637–649, 2008.
- [Por03] Stephen Portnoy. Censored regression quantiles. *Journal of the American Statistical Association*, 98(464):1001–1012, 2003.
- [Pow84] James L Powell. Least absolute deviations estimation for the censored regression model. *Journal of Econometrics*, 25(3):303–325, 1984.

- [Pow86] James L Powell. Censored regression quantiles. *Journal of econometrics*, 32(1):143–155, 1986.
- [PSA⁺10] Maya C Poffenberger, Iryna Shanina, Connie Aw, Nahida El Wharry, Nadine Straka, Dianne Fang, Annie E Baskin-Hill, Sabrina H Spiezio, Joseph H Nadeau, and Marc S Horwitz. Novel nonmajor histocompatibility complex–linked loci from mouse chromosome 17 confer susceptibility to viral-mediated chronic autoimmune myocarditisclinical perspective. *Circulation: Cardiovascular Genetics*, 3(5):399–408, 2010.
- [RDK⁺00] Charles H Redfern, Michael Y Degtyarev, Andrew T Kwa, Nathan Salomonis, Nathalie Cotte, Tania Nanevich, Nick Fidelman, Kavin Desai, Karen Vranizan, and Elena K Lee. Conditional expression of a gi-coupled receptor causes ventricular conduction delay and a lethal cardiomyopathy. *Proceedings of the National Academy of Sciences*, 97(9):4826–4831, 2000.
- [RL12] Marie-Hélène Roy and Denis Larocque. Robustness of random forests for regression. *Journal of Nonparametric Statistics*, 24(4):993–1006, 2012.
- [ROM01] Gunnar Rätsch, Takashi Onoda, and K-R Müller. Soft margins for adaboost. *Machine learning*, 42(3):287–320, 2001.
- [Ros05] Saharon Rosset. Robust boosting and its relation to bagging. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 249–255. ACM, 2005.
- [Rou84] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.
- [RW94] David Ruppert and Matthew P Wand. Multivariate locally weighted least squares regression. *The annals of statistics*, pages 1346–1370, 1994.
- [RW01] AF Ruckstuhl and AH Welsh. Robust fitting of the binomial model. *The Annals of Statistics*, 29(4):1117–1136, 2001.
- [SBH13] Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference On Learning Theory*, pages 489–511, 2013.
- [Sch13] Robert E Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013.
- [SDC03] Mark R Segal, Kam D Dahlquist, and Bruce R Conklin. Regression approaches for microarray data analysis. *Journal of Computational Biology*, 10(6):961–980, 2003.

- [Ser03] Rocco A Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4(Sep):633–648, 2003.
- [SH06] Tao Shi and Steve Horvath. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1):118–138, 2006.
- [SR92] Arnold J Stromberg and David Ruppert. Breakdown in nonlinear regression. *Journal of the American Statistical Association*, 87(420):991–997, 1992.
- [Sta89] Joan G Staniswalis. The kernel estimate of a regression function in likelihood-based models. *Journal of the American Statistical Association*, 84(405):276–283, 1989.
- [SWW14] LA Stefanski, Yichao Wu, and Kyle White. Variable selection in nonparametric classification via measurement error model selection likelihoods. *Journal of the American Statistical Association*, 109(506):574–589, 2014.
- [TH87] Robert Tibshirani and Trevor Hastie. Local likelihood estimation. *Journal of the American Statistical Association*, 82(398):559–567, 1987.
- [TPC06] Alexey Tsymbal, Mykola Pechenizkiy, and Pádraig Cunningham. Dynamic integration with random forests. In *European conference on machine learning*, pages 801–808. Springer, 2006.
- [Tyl94] David E Tyler. Finite sample breakdown points of projection based multivariate location and scatter statistics. *The Annals of Statistics*, pages 1024–1044, 1994.
- [TZC⁺16] Kai Ming Ting, Ye Zhu, Mark Carman, Yue Zhu, and Zhi-Hua Zhou. Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1205–1214. ACM, 2016.
- [VdGBRD14] Sara Van de Geer, Peter Bühlmann, Ya’acov Ritov, and Ruben Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202, 2014.
- [VDVW96] Aad W Van Der Vaart and Jon A Wellner. Weak convergence. In *Weak convergence and empirical processes*, pages 16–28. Springer, 1996.
- [VKV96] Ingrid Van Keilegom and Noël Veraverbeke. Uniform strong convergence results for the conditional kaplan-meier estimator and its quantiles. *Communications in Statistics—Theory and Methods*, 25(10):2251–2265, 1996.
- [WW12] Huixia Judy Wang and Lan Wang. Locally weighted censored quantile regression. *Journal of the American Statistical Association*, 2012.

- [Yeh98] I-C Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, 28(12):1797–1808, 1998.
- [ZWWL16] Xiang Zhang, Yichao Wu, Lan Wang, and Runze Li. Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):53–76, 2016.
- [ZY05] Tong Zhang and Bin Yu. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33(4):1538–1579, 2005.