Value-at-Risk (VaR), a widely used measure in risk management and portfolio valuation in many financial institutions, is the conditional quantile of the return distribution given the past information. Because financial time series data usually have a non-Gaussian conditional distribution, quantile regression has become very popular in VaR estimation due to its advantage of being distribution free. Especially since the CAViaR model introduced by Engle and Manganelli (2004), it has attracted much attention because the model takes a recursive fashion like GARCH model: instead of recursively regressing on volatilities, it regresses on implicit quantile functions. My work suggests to use additive smoothing splines / regression splines to approximate the unknown quantile functions. In fact, my model can also be considered as a generalized form of a class of well-studied models, e.g. TS-GARCH. Under certain assumptions, the estimation can be obtained by a two-stage procedure, including solving a modified nonparametric quantile regression problem and optimizing a nonlinear function over $\mathbb{R}^n$ space (usually $n = 1$). On selecting the best smoothing parameter, we suggest to use SIC as the criterion for our model by a simulation study, because SIC can perform almost as well as Gold Criterion (defined as minimum expected prediction error) at 5% level and relatively better than improved-AIC at both 5% and 1% levels. The proposed models are implemented on S&P 500 and IBM stock prices to estimate 5% and 1% VaR and the performance of the estimations are especially satisfactory with recursive smoothing splines.
DEDICATION

To my parents and my love.
BIOGRAPHY

Wanying Li was born to parents Runrong Li and Xiaofeng Chen on October 8th, 1982 in Shenzhen, a beautiful city in south China. She received a B.S. degree in Mathematics and Applied Mathematics in Sun Yat-Sen(Zhongshan) University in June 2005. After that, she came to U.S. and started her graduate study in Statistics Department of North Carolina State University. Inspired by the area of financial risk, she worked on her dissertation under the direction of Dr. Peter Bloomfield. In the meantime of pursuing her Ph.D., she worked as an intern in SAS Institute, Inc. and GlaxoSmithKline PLC in succession since 2007.
ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Dr. Bloomfield for all his patience and generous help throughout my study in North Carolina State University. His rich knowledge and profound insight in Statistics is really a big help to me. Nevertheless, his rigorous attitude in academic and profession is and will be the guideline in my research and career life. I am very fortunate to have him as my advisor.

My appreciation also goes to Dr. Pantula, Dr. Wang and Dr. Pelletier for kindly serving me as the committee. Their comments and critiques raised from the perspective of their own expertise are very valuable and helpful. Especially I want to thank Dr. Pantula for taking a long way back to North Carolina to attend my defense.

I can’t say too many thanks to Dr. Zudi Lu for providing me his research paper for my reference, which was helped a lot in removing blocks in my research.

In addition, I really appreciate all the professors I took courses with, because I would not be able to finish my Ph.D. without the knowledge learned from them.

Last but not least, I want to thank my parents, my love and my friends for always being around and supporting me throughout my life. Their encourage and caring is always the source of my persistence and spirit.
# TABLE OF CONTENTS

List of Tables ............................................................................. vii

List of Figures ............................................................................. viii

Chapter 1 Introduction ............................................................. 1
1.1 Value at Risk ................................................................. 1
1.2 Quantile Regression ....................................................... 4
   1.2.1 Introduction .......................................................... 4
   1.2.2 Models and Estimation ............................................ 4
1.3 Motivation and Organization ........................................... 6

Chapter 2 Additive Quantile Regression with Spline Smoothing .... 8
2.1 Introduction ..................................................................... 8
2.2 Additive Regression Spline ........................................... 8
   2.2.1 Introduction .......................................................... 8
   2.2.2 Extension to Additive Model .................................. 9
   2.2.3 Natural Cubic Regression spline ............................... 9
2.3 Additive Smoothing Spline ............................................ 10
   2.3.1 Introduction .......................................................... 10
   2.3.2 Extension to Additive Model .................................. 11

Chapter 3 Local Linear Additive Quantile Regression ................. 13
3.1 Introduction ..................................................................... 13
3.2 Local Linear Regression with Kernel Weighting ................. 14
   3.2.1 Introduction .......................................................... 14
3.3 Back-fitting Algorithm .................................................. 15
   3.3.1 Introduction .......................................................... 15
   3.3.2 Monte Carlo Simulation ......................................... 15
   3.3.3 Problem with Convergence ..................................... 17

Chapter 4 Recursive Quantile Regression Model ....................... 19
4.1 Model Motivation .......................................................... 19
4.2 Recursive model .......................................................... 20
   4.2.1 Estimation with Regression Spline ......................... 20
   4.2.2 Estimation with Smoothing Spline ......................... 21
   4.2.3 Extension to Higher Order ..................................... 24
4.3 Simulation Study .......................................................... 26

Chapter 5 Selection of Smoothing Parameter ......................... 30
5.1 Introduction ................................................................. 30
5.2 Knot Selection in Regression Spline ................................ 32
   5.2.1 Introduction .......................................................... 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.2 Selection in Recursive Model</td>
<td>33</td>
</tr>
<tr>
<td>5.2.3 Simulation Study</td>
<td>34</td>
</tr>
<tr>
<td>5.3 Selection of $\lambda$ in Smoothing Spline</td>
<td>39</td>
</tr>
<tr>
<td>5.3.1 Introduction</td>
<td>39</td>
</tr>
<tr>
<td>5.3.2 Selection in Recursive Model</td>
<td>39</td>
</tr>
<tr>
<td>5.3.3 Simulation Study</td>
<td>39</td>
</tr>
<tr>
<td>Chapter 6 Model Evaluation</td>
<td>42</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>42</td>
</tr>
<tr>
<td>6.2 Empirical Data</td>
<td>44</td>
</tr>
<tr>
<td>Chapter 7 Summary</td>
<td>67</td>
</tr>
<tr>
<td>7.1 Results</td>
<td>67</td>
</tr>
<tr>
<td>7.2 Future Work</td>
<td>68</td>
</tr>
<tr>
<td>References</td>
<td>69</td>
</tr>
<tr>
<td>Appendix</td>
<td>73</td>
</tr>
<tr>
<td>Appendix A Codes of Modified \texttt{rqss} Functions</td>
<td>74</td>
</tr>
<tr>
<td>Table</td>
<td>Title</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>3.1</td>
<td>Average Estimated ISE of Component Functions Across 300 Simulation Replicates</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of Estimation Performance between CAViaR and Linear Regression Spline at 5% VaR</td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of Estimation Performance between CAViaR and Linear Regression Spline at 1% VaR</td>
</tr>
<tr>
<td>5.1</td>
<td>Knot Selections Results w/ Different Criteria at 5% Level</td>
</tr>
<tr>
<td>5.2</td>
<td>Knot Selections Results w/ Different Criteria at 1% Level</td>
</tr>
<tr>
<td>5.3</td>
<td>Lambda Selection Results w/ Different Criteria at 5% Level</td>
</tr>
<tr>
<td>5.4</td>
<td>Lambda Selection Results w/ Different Criteria at 1% Level</td>
</tr>
<tr>
<td>6.1</td>
<td>Estimated Exception Rate and P-values of Backtesting on Predicting 5% VaR</td>
</tr>
<tr>
<td>6.2</td>
<td>Estimated Exception Rate and P-values of Backtesting on Predicting 1% VaR</td>
</tr>
</tbody>
</table>
Figure 3.1  Plots from Simulation Study 1 ............................................... 18

Figure 5.1  The Scatter Plot of Simulated Data by $l_1$ and $l_2$ and Link Functions ... 36

Figure 6.1  Daily Close Price for S&P 500 and IBM Stock Price 12/01/2004 - 11/15/2010 46
Figure 6.2  Daily Log Returns on Close Price for S&P 500 and IBM Stock .............. 47
Figure 6.3  Log Return of Today vs. Previous Day ...................................... 48
Figure 6.4  One-step Ahead Predictions of 5% VaR(quantile) on S&P 500 Daily Log Returns ................................................................. 52
Figure 6.5  One-step Ahead Predictions of 5% VaR(quantile) on S&P 500 Daily Log Returns ................................................................. 53
Figure 6.6  One-step Ahead Predictions of 1% VaR(quantile) on S&P 500 Daily Log Returns ................................................................. 54
Figure 6.7  One-step Ahead Predictions of 1% VaR(quantile) on S&P 500 Daily Log Returns ................................................................. 55
Figure 6.8  One-step Ahead Predictions of 5% VaR(quantile) on IBM Stock Daily Log Returns ................................................................. 56
Figure 6.9  One-step Ahead Predictions of 5% VaR(quantile) on IBM Stock Daily Log Returns ................................................................. 57
Figure 6.10 One-step Ahead Predictions of 1% VaR(quantile) on IBM Stock Daily Log Returns ............................................................... 58
Figure 6.11 One-step Ahead Predictions of 1% VaR(quantile) on IBM Stock Daily Log Returns ............................................................... 59
Figure 6.12 Estimated Link Functions by Asymmetric Slope at 5% VaR on S&P 500 Data ................................................................. 60
Figure 6.13 Estimated Link Functions by Linear Regression Spline at 5% VaR on S&P 500 Data ................................................................. 61
Figure 6.14 Estimated Link Functions by Natural Cubic Spline at 5% VaR on S&P 500 Data ................................................................. 62
Figure 6.15 Estimated Link Functions by Smoothing Spline at 5% VaR on S&P 500 Data ................................................................. 63
Figure 6.16 Estimated Link Functions by Asymmetric Slope at 1% VaR on S&P 500 Data ................................................................. 64
Figure 6.17 Estimated Link Functions by Linear Regression Spline at 1% VaR on S&P 500 Data ................................................................. 65
Figure 6.18 Estimated Link Functions by Natural Cubic Regression Spline at 1% VaR on S&P 500 Data ................................................................. 66
Chapter 1

Introduction

1.1 Value at Risk

In financial markets, Value at Risk (VaR) is widely used to measure the market risk, i.e. unexpected changes in portfolio values. It is defined as a threshold beyond which the probability of the loss in a portfolio over a given period of time is at a given level (mostly 1% or 5%). As VaR is usually a predictive quantity, from a statistical viewpoint, it is equivalent to a conditional quantile of the loss distribution given past information. Because of the advantage in reducing the association between probability and loss to a single number, VaR is often used as a standard measure in risk management and risk evaluations in banks and other financial institutions. A comprehensive overview on VaR was given in Duffie and Pan [1997] and Jorion [2001].

Instead of studying the threshold alone, another popular market risk measure, known as Expected Shortfall (ES), summarizes the losses beyond a threshold. It is defined as the expected loss conditional on the loss being greater than a given percentile over a given period of time.

Because the shape of the loss distribution can be reconstructed by calculating VaRs at different probability levels, ES can be derived through VaR estimations. Because a good estimate of VaR will lead to a good estimate of ES, this work focuses on discussing VaR estimation. Currently, there are several common methods to calculate VaR with historical financial data (Tsay [2005]):

1. RiskMetrics™ developed by J.P. Morgan (Morgan [1996]), i.e. IGARCH (1,1):

Let \( \{ r_t; t = 1, \ldots, T \} \) be the observations of return. Assume \( r_t | F_{t-1} \sim N(\mu_t, \sigma_t^2) \), where \( \mu_t \) is conditional mean and \( \sigma_t^2 \) is the conditional variance of return \( r_t \), and

\[
\mu_t = 0, \quad \sigma_t^2 = \alpha \sigma_{t-1}^2 + (1 - \alpha) r_{t-1}^2, \quad 1 > \alpha > 0
\]

and the value of \( \alpha \) is typically 0.94.
2. Econometric approach: parametric model such as GARCH model and other volatility models. A general model is
\[ r_t = \phi_0 + \sum_{i=1}^{k} \phi_i r_{t-i} + e_t - \sum_{j=1}^{q} \theta_j e_{t-j}, \]
\[ e_t = \sigma_t \epsilon_t, \]
\[ \sigma_t^2 = \alpha_0 + \sum_{i=1}^{u} \alpha_i e_{t-i}^2 + \sum_{j=1}^{v} \beta_j \sigma_{t-j}^2. \]
The distribution of \( \epsilon_t \) can be i.i.d. normal or \( t \) distribution, or other parametric distributions.

3. Quantile estimation: Instead of making assumptions on the distribution types as in the other methods, this method builds models for the quantiles directly.

(a) Empirical quantile estimation (Dowd [2001]): The assumption is that the returns over a period of time are independently drawn from the same continuous distribution; that is to say, the distribution of the returns is constant from the sample period to the prediction period.

We know the order statistics
\[ r_{(1)} \leq r_{(2)} \leq \cdots \leq r_{(n)}. \]
Then
i. If \( l = n \tau \), where \( l \) is an integer, \( r_{(l)} \) is the estimated \( \tau \)th-quantile.
ii. If \( l_1 < n \tau < l_2 \), \( r_{(l_i)} \) is the estimation for the \( (l_i/n) \)th quantile, then the estimated \( \tau \)th-quantile is given by
\[ \frac{\tau_2 - \tau}{\tau_2 - \tau_1} r_{(l_1)} + \frac{\tau - \tau_1}{\tau_2 - \tau_1} r_{(l_2)} \] (interpolation between data points)
where \( \tau_i = l_i/n. \)

(b) Quantile regression: First introduced by Koenker and Bassett [1978] for independent data and later was extended to time series data (Weiss [1991], Koenker et al. [1994] and Cai [2002]). The idea is to use the explanatory variables \( x_t \) to model the quantile of the returns \( r_t \), as say \( g_\tau(x_t; \beta) \). Given \( \tau \)th quantile, the estimation is to find \( \beta \) that minimizes
\[ \sum_{t=1}^{n} \rho_\tau (r_t - g_\tau(x_t; \beta)) \] (1.1)
where
\[
\rho_\tau(u) = \tau u I(u > 0) - (1 - \tau) u I(u < 0)
\] (1.2)
is called the check function. One special type of quantile regression takes the lagged quantiles as the explanatory variables, termed conditional autoregressive value at risk (CAViaR) by Engle and Manganelli [2004]): Given the fact that the volatilities of portfolio returns are autocorrelated over time, like GARCH, and VaR is often a function of volatilities, this method assumes that VaR also has an autocorrelated
structure.
Let \( \beta_\tau = (\beta_0, \beta_1, \cdots, \beta_p) \) be a \((p + 1)\)-vector of unknown parameters, \( q_t(\beta_\tau) \) be the time-\( t \) \( \tau \)th-quantile of the conditional distribution of returns given information up to time \( t - 1 \), and \( r_t \) be the portfolio return at time \( t \) (it can also be a vector containing any observable explanatory variables at time \( t \)). The model has the form as follows:
\[
q_t(\beta_\tau) = \beta_0 + \sum_{i=1}^u \beta_i q_{t-i}(\beta_\tau) + \sum_{j=1}^v \beta_{u+j} l(r_{t-j}),
\] (1.3)
where \( p = u + v + 1 \). The role of \( l(x) \) is to link \( q_t(\beta_\tau) \) to returns \( r_t \), so it controls how the VaR would change according to the past returns \( r_{t-j} \). With this form, the quantile function is smooth.
Various CAViaR processes have been considered:
Adaptive:
\[
q_t(\beta_\tau) = q_{t-1}(\beta_1) + \beta_1 \left\{ 1 + \exp \left( G[r_{t-1} - q_{t-1}(\beta_1)] \right) \right\}^{-1} - \theta
\]
where \( G \) is some positive finite number.
Symmetric absolute value:
\[
q_t(\beta_\tau) = \beta_1 + \beta_2 q_{t-1}(\beta_\tau) + \beta_3 |r_{t-1}|
\]
Asymmetric slope:
\[
q_t(\beta_\tau) = \beta_1 + \beta_2 q_{t-1}(\beta_\tau) + \beta_3 (r_{t-1})^+ + \beta_4 (r_{t-1})^-
\]
Indirect GARCH (1,1):
\[
q_t(\beta_\tau) = (\beta_1 + \beta_2 q_{t-1}^2(\beta_\tau) + \beta_3 r_{t-1}^2)^{1/2}
\]
Because CAViaR has a recursive form, the estimation is more complicated than the other quantile regression models. Engle and Manganelli (2004) proposed to use Genetic Algorithm for optimizing the sum of the check functions. Rossi and Harvey [2009] discussed an iterative Kalman filter method. And Xiao [2006] developed an effective two-step approach that considered both local and global structures.

Except for the quantile estimation method, the other methods inevitably have to estimate the VaR from an estimated distribution and have a restriction that different levels of VaR
are differed by only a constant multiplier, i.e. 1% and 5% VaR always moving in the same direction. By contrast, quantile estimation does not suffer from this because different levels of quantiles are estimated separately. In fact, one may also need to be very careful if the estimate is based on some distributional assumptions, because a wrong assumption may result in very biased estimates. For example, the normality assumption is already found problematic by Hull and White [1998]: the distribution of log returns in financial data is usually fat-tailed and asymmetric. Therefore, due to the advantage of distributional assumption free, quantile regression has become more and more widely used in VaR estimation. Details are discussed in the next section.

1.2 Quantile Regression

1.2.1 Introduction

Conditional mean models were first developed and extensively studied much earlier than quantile models. However, the classical mean model fails if the assumptions on the error distribution are violated, e.g. non-i.i.d. normal distribution. To deal with data that do not fit the classical assumption, quantile regression was first introduced in Koenker and Bassett [1978]. Unlike the conditional mean models, quantile regression can adaptively reflect the effects of covariates on the conditional distribution, and provides a more complete view of the response variables. Since then, a rapidly growing literature is devoted to extending its application to various types of data: data with heteroscedastic errors (Koenker and Bassett [1982]), time series data (Bloomfield and Steiger [1983]), data with dependent errors (Portnoy [1991]), censored data (Powell [1986]) and etc.

Quantile regression has been applied in many areas, such as survival analysis, environmental modeling, financial economics, and so on. Especially in financial economics, quantile regression is one of the most discussed topics in VaR estimation: see Taylor [1999], Chernozhukov and Umanstev [2001], and Engle and Manganelli [2004].

1.2.2 Models and Estimation

Given a data set \( \{x_i, y_i\}_{i=1}^n \), one of the conditional mean estimation method is the least square error estimation, which solves

\[
\min_m \sum_{i=1}^n (y_i - m(x_i))^2.
\]
Similarly, conditional quantile estimation solves the optimization problem

$$
\min_g \sum_{i=1}^{n} \rho_{\tau}(y_i - g_{\tau}(x_i))
$$

(1.4)

where $\rho_{\tau}(\cdot)$, called the “check function”, is used instead of the least square penalty.

Quantile regression can be categorized into three types, parametric, nonparametric and semiparametric quantile regression:

1. Parametric Regression: The form of the function $g_{\tau}(x)$ can be explicitly written down with respect to $\beta$, i.e. $g_{\tau}(x; \beta)$.

   Unlike the least square error problem, its solution has no explicit form because the check function is not differentiable. Barrodale and Roberts [1974] developed the exterior point algorithms to solve this optimization problem. Another popular algorithm to derive the solution is Frisch-Newton, which is particularly effective with large scale problems. Koenker and Portnoy [1997] advanced the interior point method by combining it with effective problem preprocessing. Their method drastically reduces time cost when processing large data sets.

   In applying parametric regression to time series data, Koenker and Zhao [1996] proposed to fit a linear quantile model with the lagged observations as predictors. Koenker and Xiao [2006] discussed a quantile autoregression model that allows different coefficients at different quantile levels. Taylor [2008] introduced weighted quantile regression, with a weight function decaying exponentially by time differences.

2. Nonparametric Regression: Instead of imposing an explicit form on the quantile function, nonparametric regression only requires smoothness. When parametric regression fails to capture the correlation between covariates and quantiles, the nonparametric method excels due to its virtue of flexibility to adapt any type of smooth function. Many studies have been devoted to exploring nonparametric regression: Antoch and Janssen [1989], Chaudhuri [1991] discussed on kernel estimators, Hendricks and Koenker [1992] considered regression splines and applied them to electricity demand data. Koenker et al. [1994] claimed to use total variation of the first derivative of the regression function as a smoothing penalty in order to derive a smoothing spline solution to the quantile regression problem.

Because nonparametric method suffers from “curse-of-dimensionality”, discussions were triggered in approaches to deal with high-dimensional data. Stone [1985] and Stone [1986] introduced a dimension-reduction method by assuming additivity of the effects of covariates on the response. Since then, the additive model has become popular in mean
and quantile regression. Without additivity assumptions, He et al. [1998], He and Ng [1999], and He and Portnoy [2000] used spline methods to address interactions. Other dimension reduction methods include index models (Chaudhuri et al. [1997], Khan [2001]) and partially linear models.

Because time series often have long-term dependence in the past information, it is common to see a time series model of more than two lagged observations. Therefore, nonparametric quantile regressions with time series data often need to solve a high-dimensional problem. One of the recent advancement is Cai and Xu [2009] discussing a linear time series model with dynamic coefficients, where the coefficients are nonparametric functions of past information.

3. Semiparametric Regression: It is proposed to incorporate the efficiency of parametric regression and the flexibility of nonparametric regression. For reference, see S. Chen [2001] and Khan [2001]. The model assumes that part of the covariates is linearly correlated with the quantile function, but not all, so the quantile function takes the form

$$g_\tau(x_i, z_i) = x_i^T \beta + m(z_i),$$

and estimates $\beta$ and $m$ by minimizing

$$\sum_{i=1}^{n} \rho_\tau(y_i - x_i^T \beta - m(z_i)) + \lambda \left( \int |m''(z)|^p dz \right)^{(1/p)}.$$

As mentioned in the previous section, CAViaR can also be regarded as a type of semiparametric regression, when the link function $l(x)$ is nonparametric, and the lagged quantiles are regarded as covariates.

1.3 Motivation and Organization

Many studies have found that in financial time series data, the current observation is often influenced by more than 5 lagged values. Therefore, instead of fitting many lagged variables into a model, an alternative way is to use recursive models. For example, the GARCH model has proved to be very successful and is well accepted in many financial applications. Instead of estimating volatility, CAViaR is one of the most popular models in estimating quantiles, as it recursively regresses the current quantile on the past ones. When used with different types of link function $l(x)$, the CAViaR model can address different relationships between the quantile and past information. However, due to the non-linearity and the difficulty in estimation, the link functions are often taken to be parametric functions. CAViaR also implies an assumption
that the effect of time $t - 1$ variable on time $t$ quantile is only different from the effect of time $t - j (j \neq 1)$ variable by a constant multiplier $\beta_{u+1}/\beta_{u+j}$. While nonparametric methods are much more flexible and require no prior knowledge, one question arises: Can an extended CAViaR model gain some benefits if the link functions are allowed to be different nonparametric functions on different lagged variables? When exploring the data, can we use the extended CAViaR model to identify the appropriate form of the link functions? Therefore, this work will mainly discuss an extended model from (1.3):

$$q_t(\beta_\tau; \tilde{l}) = \beta_0 + \sum_{i=1}^{u} \beta_i q_{t-i}(\beta_\tau) + \sum_{j=1}^{v} l_j(r_{t-j}),$$  

(1.5)

where $\tilde{l} = (l_1, l_2, ..., l_v)$ are univariate nonparametric functions on lagged variables, and its optimal estimation is to satisfy a global criterion,

$$\min_{\beta_\tau, \tilde{l}} \sum_{t} \rho_\tau \left(y_t - q_t(\beta_\tau; \tilde{l})\right).$$  

(1.6)

Following is the organization of this work. To work on the recursive model (1.5), one may need to start from the non-recursive version, i.e. when $u = 0$,

$$q_t(\beta_\tau; \tilde{l}) = \beta_0 + \sum_{j=1}^{v} l_j(r_{t-j}).$$  

(1.7)

Chapter 2 and Chapter 3 will discuss the estimation of this simplified version of additive nonparametric quantile regression model. In detail, Chapter 2 will give an overview of two spline smoothing methods: regression spline and smoothing spline, and their extensions to high dimensions. Chapter 3 will focus on local linear regression with kernel weighting and its estimation methods, including a simulation study on the backfitting algorithm. Chapter 4 will discuss how to use spline methods to estimate the recursive quantile regression model when $u = 1$ and $v = 1$ in equation (1.5). Chapter 5 gives an overview of the current methods of selecting the smoothing parameters and discussed its application in the recursive model. A simulation study is performed to justify the smoothing parameter selection criterion. Chapter 6 evaluates our models in the context of real data application in comparison with existing VaR models. This paper will finalize all the results and findings in Chapter 7 and opens up a list of questions for future work following this paper.
Chapter 2

Additive Quantile Regression with Spline Smoothing

2.1 Introduction

In nonparametric mean regression, kernel estimation, local polynomial (linear) regression and spline smoothing are the most popular methods. In last decades, all of these methods have also been extensively applied in the area of quantile regression. Particularly, spline smoothing has attracted more and more attention in the research and applications because of its computational efficiency in estimation and its adaptive problem structure that accommodates many variations in definitions. There are two most popular methods in spline smoothing, regression splines and smoothing splines.

A spline is a piecewise smooth function with continuous and smooth connections. By assuming the splines are connected at specific knots (the locations where the piecewise functions may have a change in structure), we have polynomial regression spline. On the other hand, smoothing spline does not assume specific locations of structural changes but penalizes non-smoothness along the spline.

2.2 Additive Regression Spline

2.2.1 Introduction

As mentioned above, regression spline assumes structural changes at specified knots. It fits smooth functions (e.g. polynomials) between knots, and imposes some extra conditions in order to make sure the piecewise splines are joined smoothly. In practice, instead of fitting piecewise polynomial splines with the power bases, B-spline bases (de Boor [1978]) are more often used because of computational convenience and estimation stability. The basis functions
of a $p$-degree B-spline are defined as follows (A $p$-degree polynomials spline is equivalent to a $p$-degree B-spline):

Given knots $\{\xi_{-p}, \xi_{-p+1}, ..., \xi_0, \xi_1, ..., \xi_{K+p+1}\}$, for $i = -p, -p+1, ..., K+p$,

$$B_{i,0}(x) = \begin{cases} 1 & x \in [\xi_i, \xi_{i+1}), \\ 0 & \text{o.w.} \end{cases}$$

Then $i$th B-spline basis of degree $j$, $j = 1, 2, ..., p$ is

$$B_{i,j}(x) = \frac{x - \xi_i}{\xi_{i+j} - \xi_i} B_{i,j-1}(x) + \frac{\xi_{i+j+1} - x}{\xi_{i+j+1} - \xi_{i+1}} B_{i+1,j-1}(x)$$

for $i = -p, -p+1, ..., K+p - j$, and the fitted function is

$$\hat{m}(x) = \sum_{i=-p}^{K} \alpha_i B_{i,p}(x).$$

Now the problem of fitting splines is equivalent to solving a linear quantile regression on design points $B_{i,p}(x)$ and its degrees of freedom is $K + p + 1$.

### 2.2.2 Extension to Additive Model

If model (1.7) has each component $l_j(\cdot)$ estimated by B-spline regression, then it can be reduced to

$$q(\beta_0, \alpha^{(1)}, \alpha^{(2)}, ..., \alpha^{(v)}) = \beta_0 + \sum_{j=1}^{v} \sum_{i=-p}^{K^{(j)}} \alpha^{(j)}_{i} B_{i,p}^{(j)}(r_{t-j}),$$

where $\alpha^{(j)} = \{\alpha^{(j)}_1, \alpha^{(j)}_2, ..., \alpha^{(j)}_v\}$. This is actually a quantile regression and can be implemented with the existing R packages quantreg and splines. The estimation is to satisfy the global criterion (1.6)

$$\min_{\beta_0, \alpha^{(1)}, ..., \alpha^{(v)}} \sum_{t=j+1}^{T} \rho_{\tau} \left( r_t - (\beta_0 + \sum_{j=1}^{v} \sum_{i=-p}^{K^{(j)}} \alpha^{(j)}_{i} B_{i,p}^{(j)}(r_{t-j})) \right). \quad (2.1)$$

### 2.2.3 Natural Cubic Regression spline

A natural cubic regression spline is a 3-degree polynomial regression spline with an additional condition of linearity at the boundary, i.e. $g''(x) = g'''(x) = 0$. Nonparametric estimations usually have boundary effects, that is, estimations have high variations on both ends of the value range, especially in the financial data, we can see in the Chapter (6.2) that they are extremely sparse near the boundary. Therefore, it is natural to impose linearity condition to
reduce the estimation uncertainty. Thanks to its computational stability and reduced boundary
effects in estimations, the natural cubic spline is also one of the most popular regression spline
basis. Chambers and Hastie (1982) have shown that a natural cubic regression spline also
has an equivalent B-spline representation. The regression procedure is therefore very similar
to B-spline regression, which can be naturally extended to additive models estimation and be
implemented with existing R packages. Due to the additional constraints on the boundary,
given \( K \) knots, a natural cubic spline has 4 degrees of freedom less than a 3-degree B-spline
regression, that is \( K \) d.f.

### 2.3 Additive Smoothing Spline

Though regression splines have many attractive properties, e.g. low computational complexity
and easy extension to multivariate regression, the goodness of fit is strongly affected by the
number of knots and knot locations. When knot locations are unknown, the procedure to select
the best locations is to search in an \( n \)-dimensional space. Unlike regression spline, smoothing
spline defines every observation point as a potential knot, but imposes a penalty on the
smoothness of the curve. A univariate quantile regression solves

\[
\min_g \sum \rho_\tau \{ y_i - g(x_i) \} + \lambda \left( \int |g''(x)|^p dx \right)^{1/p},
\]

where \( p \geq 1 \). The first part controls the goodness of fit to the quantile, while the second part
controls the smoothness of the fit. Then selection of knot locations is replaced by selecting a \( \lambda \)
value.

#### 2.3.1 Introduction

Koenker et al. [1994] showed that a solution to problem (2.2) given \( p = 1 \) is a linear spline with
knots at the observations \( \{ x_i \}_{i=1}^n \), when the solution space is expanded to include functions
with absolutely continuous first derivatives, and \( L_1 \) penalty is replaced with a total variation
on \( g' \), i.e. \( V(g') = \sup_{F} \sum |g'(s_i) - g'(s_{i-1})| \), and \( \mathbb{P} \) is the set of all partitions \( \{ s_1, s_2, ... \} \) over the
definition range.

Suppose \( \hat{g}(x) = a_i + b_i(x - x_i) \) for \( x \in [x_i, x_{i+1}) \), then

\[
\hat{g}(x_i) = a_i
\]

and \( b_i = (a_{i+1} - a_i)/h_i \)

where \( h_i = x_{i+1} - x_i \).

\[
(2.3) \quad \text{and} \quad (2.4) \quad \text{and} \quad (2.5)
\]
Thus,
\[ V(\hat{g}') = \sum_{i=1}^{n-1} |b_{i+1} - b_i| = \sum_{i=1}^{n-1}|(a_{i+2} - a_{i+1})/h_{i+1} - (a_{i+1} - a_i)/h_i|. \] (2.6)

Let \( a = (a_1, a_2, ..., a_n)^T \); then the problem (2.2) can be written as a linear programming problem:
\[ \min_{a \in \mathbb{R}^n} \sum_{i=1}^{n} \rho_r(y_i - a_i) + \lambda \sum_{j=1}^{n-1} |d'_j a|, \] (2.7)
where
\[ d'_j = (0, ..., 0, h_j^{-1}, -(h_j^{-1} + h_{j+1}^{-1}), h_{j+1}^{-1}, 0, ..., 0). \] (2.8)

### 2.3.2 Extension to Additive Model

Extension of smoothing spline to additive model is not well defined because of the problem could be different given different form of the penalty on the smoothness of a multivariate function, but under the assumption of additivity, the extension is straightforward. He and Shi [1996] proposed a bivariate quantile smoothing spline which solves
\[ \min_g \left[ \sum_i \rho_r\{y_i - g(x_{1i}, x_{2i})\} + R(g) \right], \] (2.9)
where \( R(g) = \lambda_1 \sum_i V_2\{ \partial g(x_{1i}, \cdot) / \partial x_2 \} + \lambda_2 \sum_i V_1\{ \partial g(x_{1i}, \cdot) / \partial x_1 \} \) and \( V_1(h) \) is the total variation of \( h(\cdot, x_2) \) along \( x_1 \) direction and \( V_2(h) \) is the total variation of \( h(x_{1i}, \cdot) \) along the \( x_2 \) direction.

Within the class of functions that are continuous, twice continuously differentiable on each observed subrectangle \([x_{1i}, x_{1,i+1}] \times [x_{2i}, x_{2,i+1}]\) and have bounded total variations of partial derivatives in \( x_1, x_2 \) respectively\(^1\), the minimizer to this problem is bilinear tensor spline.

Given the assumption of additivity, \( g = g_1(x_1) + g_2(x_2)\),
\[ R(g) = n\lambda_1 V_2(g'_2) + n\lambda_2 V_1(g'_1) = \lambda_1^* V(g'_2) + \lambda_2^* V(g'_1) \equiv \tilde{R}(g). \]

**Proposition 2.3.1** If \( g \) is an additive bivariate function under the condition, the solution that solves
\[ \min_g \left[ \sum_i \rho_r\{y_i - g(x_{1i}, x_{2i})\} + \tilde{R}(g) \right] \]
is also an additive function, and each component is linear spline with knots at the observations \( x_{1i} \) and \( x_{2i} \) (\( i = 1, 2, ..., n \)) respectively.

**Proof** Given the values of \( g \) at observations \((x_{1i}, x_{2i}), i = 1, 2, ..., n\), according to the proof of
\(^1\)The same conditions as in He et al. [1998].
Theorem 1 in He et al. [1998], there exists a bilinear tensor product spline $g^*$ that interpolates
the same values of $g$ at $(x_{1i}, x_{2i})$ and minimizes $\tilde{R}(g)$.

Within each subrectangle $(a, b) \times (c, d)$, since $g_{12}^* = c^*$ is a constant,

\[
\int \int |g_{12}^*| = \int \int g_{12}^* \\
= |g^*(a, c) + g^*(b, d) - g^*(b, c) - g^*(a, d)| \\
= |(g_1(a) + g_2(c)) + (g_1(b) + g_2(d)) - (g_1(b) + g_2(c)) - (g_1(a) + g_2(d))| \\
= 0.
\]

Therefore, $g_{12}^* \equiv 0$, and $g^*$ is an additive bilinear tensor product spline where each component
is a linear spline. Then what remains to solve is to search over a 2n-grid that minimizes the
loss sum, which is an attainable solution.

Therefore, the estimation of additive bivariate smoothing spline is to satisfy a global crite-
ron,

\[
\min_{a^{(1)}, a^{(2)}} T \sum_{i=1}^{T} \rho_r \left( y_i - (a_i^{(1)} + a_i^{(2)}) \right) + \lambda_1 \sum_{j=1}^{n-1} |d_{xj}' a^{(1)}| + \lambda_2 \sum_{j=1}^{n-1} |d_{xj}' a^{(2)}|,
\]

which is an extension form of the criterion (1.6).
Chapter 3

Local Linear Additive Quantile Regression

3.1 Introduction

Since local linear regression is natural and interpretable, so it has become very popular since locally weighted scatterplot smoothing (LOWESS) was introduced. In quantile regression, local linear regression has also gained much attention lately. Yu and Lu [2004] discussed an additive quantile regression model:

\[ q_i = \beta_0 + \sum_{j=1}^{v} g_j(x_{ij}), \]

where \( \{g_j(\cdot)\}_{j=1}^{v} \) are univariate nonparametric functions. They used kernel weighting local linear regression on each component and backfitting algorithm in the estimation. Alternative methods to estimate this model are also discussed. Lu et al. [2004] further discussed the asymptotic properties of the estimates under \( \alpha \)-mixing conditions. Gooijer and Zerom [2003] applied a marginal integration approach, which first estimates a high-dimensional quantile function by inverting the conditional distribution and then projects them to each component. Horowitz and Lee [2005] use a two-stage approach that consist of the series estimation in the first step and a local polynomial fitting in the second. Kong et al. [2010] generally discussed the local polynomial estimates of M-Regression and their application to additive model under \( \alpha \)-mixing conditions.
3.2 Local Linear Regression with Kernel Weighting

3.2.1 Introduction

In multivariate local linear regression with kernel weighting, given a kernel density function \( K_h : \mathbb{R}^v \rightarrow \mathbb{R} \), the estimation at observation \( \tilde{x} \) is to optimize a global criterion over \( \{a_t\}_{t=1,2,...,v} \) and \( \{b_t\}_{t=1,2,...,v} \),

\[
\sum_{i=1}^{n} \rho_{\tau} \left\{ Y_i - \beta_0 - \sum_{t=1}^{v} (a_t + b_t(X_{it} - x_t)) \right\} K_h(X_i - \tilde{x}) \tag{3.1}
\]

where \( \tilde{x} = (x_1, x_2, \ldots, x_v) \).

Kong et al. [2010] proposed to apply marginal integration on each direction to the estimates in order to obtain the estimates of additive components. The procedure is equivalent to finding the additive model to approximate the estimates. However, the estimates may suffer from “curse-of-dimensionality”. The additive model approximation may not be very helpful in dimension reduction because the approximations is dependent on the quality of estimates in the first step.

In order to avoid the problem incurred by high dimensions, Yu and Lu [2004] developed a recursive procedure to estimate additive local linear regression model. Instead of aiming at a global criterion, the procedure is to minimize

\[
\sum_{i=1}^{n} \rho_{\tau} \left( y_i - \beta_0 - \sum_{j=1}^{v} l_j(x_{ij}) \right) \tag{3.2}
\]

and for all \( s = 1, 2, \ldots n \) and \( t = 1, 2, \ldots v \),

\[
\sum_{i=1}^{n} \rho_{\tau} \left( y_i - \beta_0 - \sum_{j \neq t} l_j(x_{ij}) - a_{st} - b_{st}(x_{it} - x_{st}) \right) K \left( \frac{x_{it} - x_{st}}{h_t} \right) \tag{3.3}
\]

where \( K(\cdot) \) is a univariate kernel density function and \( h_t \) is the bandwidth for \( t \)-th component \( l_t(x) \). The estimation for each component is then \( \hat{l}_j(x_{ij}) = \hat{a}_{ij} \).

For each \( x_{st} \), (3.3) is equivalent to a univariate quantile regression problem, weighted so that the observations far from \( x_{st} \) have little impact on the local estimation.
3.3 Back-fitting Algorithm

3.3.1 Introduction

Obtaining a solution for the optimization problem (3.2) and (3.3) in one step is difficult. However, the back-fitting algorithm provides a feasible procedure to find a solution, by regressing one component at a time iteratively, until convergence criteria are satisfied. Following are the detailed steps:

1. Initial estimation of $\hat{\beta}_0$:

   $$\hat{\beta}_0 = \text{Argmin}_{\beta_0} \sum_{i=1}^{n} \rho_\tau(y_i - \beta_0);$$

   which is equivalent to finding the $\tau$th sample quantile of $\{y_i\}$. Set

   $$\hat{l}_j(x_{ij}) = 0$$

   for all $i$ and $j$.

2. Let $t = 1$, and $s$ to run from 1 to $n$,

   $$(\hat{a}_{st}, \hat{b}_{st}) = \text{Argmin}_{a,b} \sum_{i=1}^{n} \rho_\tau \left( y_i - \beta_0 - \sum_{j \neq t} \hat{l}_j(x_{ij}) - a_{st} - b_{st}(x_{it} - x_{st}) \right) K \left( \frac{x_{it} - x_{st}}{h_t} \right)$$

   Then update $\hat{l}_t(x_{it}) = \hat{a}_{it}$ for $i = 1, 2, ..., n$.

3. Update $\hat{\beta}_0$ with the $\tau$th sample quantile of $\left\{y_i - \sum_{j=1}^{v} \hat{l}_j(x_{ij})\right\}$

4. Continue with $t = 2, 3, ..., v$ to repeat steps 2 and 3.

5. Cycle step 2 to 4 until estimations $\{\hat{\beta}_0, \hat{l}_j(x_{ij})\}$ converge for all $i$ and $j$.

3.3.2 Monte Carlo Simulation

Because of the iterative process, showing the theoretical properties of backfitting algorithms is difficult. A Monte Carlo simulation was implemented to explore some properties of this backfitting algorithm.

1. Ready extension to correlated covariates:

   Current studies have shown that back-fitting algorithm is successful in additive quantile regression estimation with independent data, but there is little discussion about the performance of backfitting algorithm when the covariates are highly correlated.
Simulation settings: Explanatory variable \( \{x_{1i}\}_{i=1}^{n}(n=200, 500, 1000) \) is from a uniform distribution \( U(-3, 3) \) and \( x_2 \) is linearly correlated to \( x_1 \):

\[
x_{2i} = 0.6x_{1i} + \epsilon_i,
\]

where \( \epsilon_i \) i.i.d \( N(0, 1) \). However, quantile function of \( y \) is only dependent on \( x_1 \):

\[
y_i = 2x_{1i} + \sqrt{\frac{10 - x_{1i}^2}{3}}u_i,
\]

where \( u_i \) i.i.d \( N(0, 1) \). Therefore, the \( \tau \)-th-quantile function can be written down explicitly:

\[
q_\tau(x_1) = 2x_1 + \sqrt{\frac{10 - x_1^2}{3}}F_u^{-1}(\tau).
\]

In the simulations, both variables will be fitted into the model with \( x_1 \) as the first component and \( x_2 \) as the second to obtain the 75th and 95th quantile estimates. Each sample size setting has 300 replicates.

Figure (3.1a) shows the simulation data with the corresponding 25th, 50th and 75th quantile in the red and green dashes from low to high. Figure (3.1b) is the scatter plot of \( y \) vs \( x_2 \). Because both \( x_2 \) and \( y \) are highly correlated with \( x_1 \), it is not surprising to see an obvious correlation between \( x_2 \) and \( y \). Figure (3.1c) and figure (3.1d) are respectively the estimated 75th quantile functions of the first and second component in 30 simulation replicates. Because local linear regression is subject to boundary effects, the component estimates are very volatile at the both ends. Figure (3.1e) and figure (3.1f) are the average estimated component functions vs. the true functions (dashed lines). The estimates are very close to the true functions on the graphs. To measure the difference between \( \hat{l}_i \) and \( l_i \), integrated squared error,

\[
\int (\hat{l}_i(x_i) - l_i(x_i))^2dF(x_i), \quad i = 1, 2
\]

is estimated and summarized in the following Table (3.1), where the estimated ISE goes to 0 as the sample size grows larger.

2. Independence of estimation order: Repeat the same simulation setting as above, but fit \( x_2 \) as the first component and \( x_1 \) as the second component. The results show that the estimations are the same as that of study 1. Although back-fitting algorithm takes an iterative procedure and fits the nuisance variable \( x_2 \) first, the final solution still converges to the true model.
Table 3.1: Average Estimated ISE of Component Functions Across 300 Simulation Replicates

<table>
<thead>
<tr>
<th>n=200</th>
<th>n=500</th>
<th>n=1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{l}_1 )</td>
<td>75th</td>
<td>0.051</td>
</tr>
<tr>
<td></td>
<td>95th</td>
<td>0.110</td>
</tr>
<tr>
<td>( \hat{l}_2 )</td>
<td>75th</td>
<td>0.025</td>
</tr>
<tr>
<td></td>
<td>95th</td>
<td>0.049</td>
</tr>
</tbody>
</table>

3.3.3 Problem with Convergence

Throughout the implementation, the estimation is not always convergent with the backfitting algorithm. In terms of computation, it means convergence criteria are not met (i.e. the differences of the estimations from the last two steps cannot go down below any given threshold). To consider the whole process of the backfitting algorithm, one can note that the procedure does not have a global optimization target, but aims to optimize a different target function (3.3) in each step, in the hope that optimizing in each step will lead to optimizing the global target function (3.1). Therefore, the convergence is not guaranteed. In the simulation settings above, the estimations do not always converge. Even though the halfing step procedure was implemented, the convergence issue is still not solved.

One way to avoid this problem can be to first obtain the estimates with spline smoothing methods and use them as initial estimate in the backfitting algorithm to update estimates with local linear regression. In this way, the backfitting algorithm is only implemented once and there is no need to proceed recursively until convergence, because the initial estimates are already consistent estimates.
Figure 3.1: Plots from Simulation Study 1
Chapter 4

Recursive Quantile Regression Model

4.1 Model Motivation

Recursive models are very popular in time series because they can incorporate long-term dependence with a moderate number of parameters, for example, the GARCH model. Nowadays, much literature is devoted to studying models derived from GARCH. Among them, Taylor [1986] and Schwert [1990] proposed TS-GARCH model, which mitigates the influence of large values on the standard deviation by modeling with absolute lagged values instead of quadratic values:

\[ \sigma_t = \alpha_0 + \sum_j \alpha_j |r_{t-j}| + \sum_i \theta_i \sigma_{t-i}, \]

where the underlying assumption is that the data distributions are in the same family that can be fully described by mean and standard deviation. To avoid the distribution assumption, one can substitute \( \sigma_t \) with the \( \tau \)-th-quantile \( q_\tau(t) \). (Denoted as \( q_t \) as a simplified notation in the following parts.)

\[ q_t = \beta_0 + \sum_j \beta_j |r_{t-j}| + \sum_i \theta_i q_{t-i}. \]

This model above is actually equivalent to model (1.5) where \( l_j(\cdot) \) is a linear regression spline with a single knot at 0. Therefore, the model discussed in this work is a more generalized form of TS-GARCH, allowing \( l_j \) to be a regression spline with more than one knot, or a smoothing spline. Notice that as mentioned in Section (1.3), this work is also motivated as a generalized form of CAViaR, which was first developed from the perspective of quantile regression. Therefore, our model integrates the classical time series models and the quantile regression techniques.
In this chapter, the estimation procedure of a recursive nonparametric quantile regression will be discussed in detail. As the spline smoothing has the advantage of computing efficiency and alternative linear expressions, we are able to transform the original problem into a solvable linear programming problem. Recursive model with regression spline and smoothing spline will be separately described in the following.

4.2 Recursive model

4.2.1 Estimation with Regression Spline

First consider the simplest case. Suppose the quantile is an additive function of one lagged quantile value and a function of one lagged observation value, that is, when \( u = 1 \) and \( v = 1 \) in model (1.5):

\[
q_t = \theta q_{t-1} + l(r_{t-1}),
\]

where \( t = 2, ..., T \), and parameter \( \beta_0 \) is included in function \( l(\cdot) \).

If the nonparametric component \( l(\cdot) \) can be approximated by a regression spline with \( K \) basis functions,

\[
l(x) = \sum_{j=1}^{K} \beta_j B_j(x).
\]

Then, substitute \( q_{t-1} \) with the basis functions recursively, the equation becomes

\[
q_t = l(r_{t-1}) + \theta q_{t-1} = \sum_{j=1}^{K} \beta_j B_j(r_{t-1}) + \theta q_{t-1} = \sum_{j=1}^{K} \beta_j B_j(r_{t-1}) + \theta \left( \sum_{j=1}^{K} \beta_j B_j(r_{t-2}) + \theta q_{t-2} \right) = \sum_{j=1}^{K} \beta_j (B_j(r_{t-1}) + \theta B_j(r_{t-2})) + \theta^2 q_{t-2},
\]

and after repeating the substitution \( (t - 2) \) times,

\[
q_t = \sum_{j=1}^{K} \beta_j \left( \sum_{i=1}^{t-1} \theta^{i-1} B_j(r_{t-i}) \right) + \theta^{t-1} q_1. \tag{4.1}
\]

Now, for each fixed \( \theta \), the model can be viewed as a linear quantile regression model, where \( \beta' = (\beta_1, \beta_2, ..., \beta_K, q_1) \) and the new design points are preserved basis function values. Denote
\( q_t \) as \( Q_t(\beta, \theta) \), the estimation minimizes

\[
\sum_{t=1}^{T} \rho_{\tau}(y_t - Q_t(\beta, \theta)).
\] (4.2)

The estimation of the recursive model is two-staged:

1. Given a grid of \( \theta \)s within \((-1, 1)\) that minimizes (4.2), for each \( \theta \), estimate the corresponding parameter \( \beta \).

2. The final estimation is the \((\hat{\beta}, \hat{\theta})\) that minimizes (4.2).

Usually a two-stage optimization is in concern of reaching at whether the global or local minimization. However, this estimation process does not suffer from this issue. The second stage is minimizing a univariate function of \( \theta \) within a constraint range. If the first stage always derives a global minimum at each given \( \theta \), the global minimum of (4.2) can be achieved by a grid search over \( \theta \). Note that the first stage is just a classical linear quantile regression, which means that the estimation is a global minimizer of (4.2) at each \( \theta \), so this two-stage estimation will achieve the global minimum with a grid search over \( \theta \). As shown in the later Chapter, in real data applications, the second stage univariate function is usually unimodal, so in terms of computational implementation, the \texttt{R} function \texttt{optimize} is used instead of brutal grid search algorithm to improve computing efficiency. Estimation with regression spline does not require too much computing effort because it is just a combination of linear quantile regression with transformed design matrix and a minimization of a univariate function.

### 4.2.2 Estimation with Smoothing Spline

The estimation with smoothing spline, compared to regression spline, is a little more complicated and requires much more computing capacity in a single run. Denote the stochastic process \( \{q_t\} \) as \( Q(x|F_{t-1}) \), \( F_{t-1} = \{q_{t-1}, q_{t-2}, ..., r_{t-1}, r_{t-2}, ...\} \), then

\[
Q(x|F_{t-1}) = l(x) + \theta Q(r_{t-1}|F_{t-2}).
\]

Therefore,

\[
\frac{dQ(x|F_{t-1})}{dx} = \frac{dl(x)}{dx}
\]

is constant \( \forall t \). That is, the smoothness of \( Q(x|F_{t-1}) \) only depends on \( l(x) \) over time. Now the smoothing spline problem becomes to minimize

\[
\sum_{t=1}^{T} \rho_{\tau}\{r_t - Q(r_{t-1}|F_{t-1})\} + \lambda V(l').
\] (4.3)
Proposition 4.2.1 The minimizer of problem (4.3) is a linear spline with knots at the observations \( \{ x_i \}_{i=1}^n \) for each fixed \( \theta \) where \( g \) is a function with absolutely continuous first derivatives.

Proof If \( Q(r_t|F_{t-1}) = q_t^\star \) is fixed, then \( l(r_t) = Q(r_t|F_{t-1}) - \theta Q(r_{t-1}|F_{t-2}) = q_t^\star - \theta q_{t-1}^\star \) is also fixed. Therefore, we can first obtain the form of \( \hat{l} \) that minimizes the roughness penalty \( \lambda V(l') \), then what remains is just an interpolation problem to minimize the sum of two parts.

By the same arguments in Koenker et al. [1994], it is straightforward to show the curve that minimizes \( V(l') \) is a piecewise linear interpolator with knots at the observations \( r_t \), and thus it can also minimize (4.3).

Define an index function \( i_t = \text{Argmin}_l \{ r_l : r_l > r_t \} \), \( (t = 1, ..., T - 1) \), i.e. \( r_{i_t} \) is the smallest value greater than \( r_t \). Without loss of generality, assume that

\[
l(x) = a_t + b_t(x - r_t), \ x \in [r_t, r_{i_t})
\]

Then,

\[
q_2 = Q(r_1|F_1) = l(r_1) + \theta q_1 = a_1 + \theta q_1
\]

\[
l'(x) |_{x=r_1} = b_1
\]

\[
q_3 = Q(r_2|F_2) = l(r_2) + \theta Q(r_1|F_1) = l(r_2) + \theta (a_1 + \theta q_1) = a_2 + \theta a_1 + \theta^2 q_1
\]

\[
l'(x) |_{x=r_2} = b_2
\]

\[
:\quad
\]

To summarize,

\[
q_t = Q(r_{t-1}|F_{t-1}) = \sum_{i=1}^{t-1} \theta^{t-1-i} a_i + \theta^{t-1} q_1, \quad (4.4)
\]

\[
l'(x) |_{x=r_{t-1}} = b_{t-1}, \quad (4.5)
\]

where \( b_t = \frac{a_{i_t} - a_t}{x_{i_t} - x_t} \), and \( t = 2, ..., T \).

Note the difference of the definition of \( a_t \) and \( b_t \) in equation (4.6) and in equation (2.4) The algorithm described in section (2.3.1) by default sorts the data before fitting a smoothing spline. If the order of observations is not meaningful, sorting the data would not make a difference in the quantile estimation. However, it is not true in this model because the quantile is a weighted sum of past quantile values. Therefore, some modifications are required in the algorithm.
Define a set of index

\( \{s_t, t = 1, \ldots, T - 1\} \), s.t. \( r_{s_1} \leq r_{s_2} \leq \ldots \leq r_{s_{T-1}} \),

which means \( s_1 \) is the time when the return value is the smallest, and \( s_2 \) is the time when the return is second smallest, and so on, \( s_{T-1} \) is the time when the return is the largest. Assume \( r_t \) values are distinct, then for each \( r_{s_t} \),

\[ \exists m, \text{ s.t. } r_m = r_{s_t}, \text{ and } r_{tm} = r_{s_{t+1}} \]

Therefore, the model can be estimated following the similar procedures as usual quantile smoothing splines, except that the parameters need to be re-arranged as

\[ a' = (q_1, a_{s_1}, a_{s_2}, \ldots a_{s_{T-1}}), \]

then the total variation part can be written similarly as the second part of equation (2.7) as

\[ \lambda \sum_{j=1}^{T-2} |d_j a|, \]

where \( d_j \) is the same as (2.8) and \( h_t = r_{s_{t+1}} - r_{s_t}, (t = 1, \ldots, T - 1) \). The estimations can be obtained by using \texttt{rqss} function in \texttt{quantreg} package of \texttt{R} but need some modification in the functions \texttt{rqss} and \texttt{qss} to adjust for the changes in the underlying design matrix. The modified function codes are attached in the appendix.

One issue raised in this method is that it requires intensive computing capacity due to the change of design matrix structure. In the original smoothing spline problem, with \( N \) observations, the dimension of the design matrix is \((2N - 2) \times N\), but in most cells the value is just zero and the \texttt{R} package \texttt{quantreg} is able to use a sparse matrix storage method in order to reduce the memory usage for computation. However, in the recursive model, the quantile function is no longer equal to a single \( a_t \) value but a summation of all previous values \( \sum_{i=1}^{t-1} \theta^{t-1-i} a_i + \theta^{t-1} q_1 \). When \( |\theta| \) is close to 1, the decaying rate of \( \theta^{t-1-i} \) is relatively slow when \( i \) grows, thus, the design matrix is no longer sparse and the memory requirement is almost \( \frac{1}{2} N^2 \) instead of \( N \). Therefore, the algorithm can go over the memory limit very quickly when the number of observations grows. In approximation, 1000 observation requires around 30G in memory capacity. In order to alleviate this problem, one method is to reparameterize \( \{a_t\} \) into \( \{q_t\} \), so that the \( \theta^{t-1-i} \) terms become zeros in the design matrix. This modification saves the memory dramatically. For the same 1000 observation case, the requirement in memory is only 400M.
The modified function of \texttt{rqss} and related codes are attached in the appendix.

4.2.3 Extension to Higher Order

The case when \( u > 1 \) or \( v > 1 \) is not discussed in detail in this work, but the estimation can be extended from the simple case \( u = 1 \) and \( v = 1 \).

- When \( u = 1 \) and \( v > 1 \),
  \[
  q_t = \theta q_{t-1} + \sum_{d=1}^{v} I_d(r_{t-d}),
  \]
  the extension is straightforward.

1. Regression Spline: Each link component \( I_d \) can be estimated by a regression spline with \( K^d \) basis functions, then similar to equation 4.1,
  \[
  q_t = \sum_{d=1}^{v} \sum_{j=1}^{K^d} \beta_{j,d} \left( \sum_{i=1}^{t-1} \theta^{i-1} B_{j,d}(r_{t+1-i-d}) \right) + \theta^{t-1} q_1,
  \]
  and for each fixed \( \theta \), estimation can be derived by a linear quantile regression.

2. Smoothing Spline: Each link component \( I_d \) is a piecewise linear function, then the estimation is equivalent to additive quantile smoothing spline for each fixed \( \theta \).

- When \( u > 1 \) and \( v = 1 \),
  \[
  q_t = \sum_{s=1}^{u} \theta_s q_{t-s} + I(r_{t-1}),
  \]
  the estimation form remains the same as the case \( u = 1 \) and \( v = 1 \). As an example, following is a brief description of the estimation when \( u = 2 \) and \( v = 2 \).

1. Regression Spline

   In analogue to equation 4.1, suppose that the estimation has a general form
   \[
   q_t = \sum_{j=1}^{K} \beta_j X_{t,j} + W_{1,t} q_1 + W_{2,t} q_2.
   \] (4.7)

   To substitute this equation in the model
   \[
   q_t = \theta_1 q_{t-1} + \theta_2 q_{t-2} + \sum_{j=1}^{K} \beta_j B_j(r_{t-1}),
   \]
so that
\[ q_t = \theta_1 \left( \sum_{j=1}^{K} \beta_j X_{t-1,j} + W_{1,t-1} q_1 + W_{2,t-1} q_2 \right) \\
+ \theta_2 \left( \sum_{j=1}^{K} \beta_j X_{t-2,j} + W_{1,t-2} q_1 + W_{2,t-2} q_2 \right) + \sum_{j=1}^{K} \beta_j B_j (r_{t-1}). \]

By equating the coefficients of \( q_1, q_2 \) and \( \{ \beta_j \}_{j=1,...,K} \) to the general equation form,

\[ X_{t,j} = \theta_1 X_{t-1,j} + \theta_2 X_{t-2,j} + B_j (r_{t-1}), \]
\[ W_{1,t} = \theta_1 W_{1,t-1} + \theta_2 W_{1,t-2}, \]
\[ W_{2,t} = \theta_1 W_{2,t-1} + \theta_2 W_{2,t-2} \quad \text{for } t > 2, \]

and

\[ X_{1,j} = 0, X_{2,j} = 0, \]
\[ W_{1,1} = 1, W_{1,2} = 0, \]
\[ W_{2,1} = 0, W_{2,2} = 1. \]

Because the estimation with regression spline is in the form (4.7), it can also be estimated by linear quantile regression given fixed \( (\theta_1, \theta_2) \), where the design matrices need more calculations than in the simple case. Now the dimension of the search space in the second stage of the estimation is increased.

2. Smoothing Spline

In analogue to equation 4.6, suppose that the estimation has a general form

\[ q_t = \sum_{j=2}^{t-1} X_{t,j} a_j + W_{1,t} q_1 + W_{2,t} q_2. \quad (4.8) \]

To substitute this equation in the model

\[ q_t = \theta_1 q_{t-1} + \theta_2 q_{t-2} + a_{t-1} \]

so that
\[ q_t = \theta_1 \left( \sum_{j=2}^{t-2} X_{t-1,j} + Wa_j + W_{1,t-1}q_1 + W_{2,t-1}q_2 \right) \\
+ \theta_2 \left( \sum_{j=2}^{t-3} X_{t-2,j} + Wa_j + W_{1,t-2}q_1 + W_{2,t-2}q_2 \right) + a_{t-1}. \]

By equating the coefficients of \( q_1 \), \( q_2 \) and \( \{a_j\}_{t=2,\ldots,t-1} \) to the general form, for \( t > 2 \),

\[
W_{1,t} = \theta_1 W_{1,t-1} + \theta_2 W_{1,t-2}, \\
W_{2,t} = \theta_1 W_{2,t-1} + \theta_2 W_{2,t-2}
\]

and

\[
X_{t,t-1} = 1 \\
X_{t,t-2} = \theta_1 X_{t-1,t-2} = \theta_1 \\
X_{t,i} = \theta_1 X_{t-1,i} + \theta_2 X_{t-2,i} \text{ for } 3 \leq i \leq t-3,
\]

and the initial values are

\[
W_{1,1} = 1, W_{1,2} = 0, \\
W_{2,1} = 0, W_{2,2} = 1.
\]

Again, the estimation is in the form 4.8, that is, the link function can be estimated by linear smoothing spline at given \((\theta_1, \theta_2)\), but the design matrix needs more computations than the simple case.

In summary, the estimations of the recursive model at higher order actually have similar form to that in the simplest case, so the computational procedure described in the previous sections can also be applied to higher order cases. In addition, the model with \( u = 1 \) and \( v = 1 \) is often found to be sufficient in practice, so this work will mainly focus on the simple case.

### 4.3 Simulation Study

As discussed in Section 4.2.1, in theory, the linear regression spline with a single knot at 0 share the same model as CAViaR of Asymmetric Slope. Both of the methods obtain the estimations by minimizing the same criterion (1.6), the only difference is the computational strategy. This section will continue the discussion by comparing the estimation performance of these two methods.
methods in a Monte Carlo simulation study.

We consider a quantile function

\[ q_t = 0.9q_{t-1} + l(r_{t-1}), \]

where \( l(x) = [0.09 - 0.02xI(x < 0) + 0.01xI(x >= 0)]Q_\epsilon(\tau) \), and \( \epsilon \) is from a skewed-t distribution \( (\mu = 0, \sigma = 2, \nu = 6.35, \xi = .92) \). \( (Q_\epsilon(0.05) = -3.29, Q_\epsilon(0.01) = -5.38) \) So to rewrite it in the form of

\[ q_t = \beta_1 + \beta_2 q_{t-1} + \beta_3 (r_{t-1})^+ + \beta_4 (r_{t-1})^-, \]

for \( \tau = .05 \),

\[ \beta_1 = -0.30, \beta_2 = 0.9, \beta_3 = -0.033, \beta_4 = -0.066, \]

for \( \tau = .01 \),

\[ \beta_1 = -0.48, \beta_2 = 0.9, \beta_3 = -0.054, \beta_4 = -0.108 \]

The observations \( \{r_t\} \) can be generated recursively as follows

\[ r_1 = \epsilon_1, \]
\[ q_1 = Q_\epsilon(\tau), \]

for \( t \geq 2 \), and \( k = 1, 2 \)

\[ r_t = \{[0.9q_{t-1} + l(r_{t-1})] / Q_\epsilon(\tau)\} \epsilon_t \]
\[ q_t = 0.9k_{t-1} + l(r_{t-1}), \]

where \( \{\epsilon_t\} \) are independently generated from the skewed-t distribution.

In order to study how the estimation performance is influence by the sample size, data are generated with three different size \( (n = 250, 500, 1000) \), and for each size we have 300 replicates. Quantiles are estimated at both 1% and 5% level by two methods, CAViaR with Asymmetric Slope and Linear Spline Regression with knot fixed at 0. The goodness of fit is measured by the Average Absolute Deviation (AAD) of estimation from the true quantile values,

\[ AAD(\hat{q}\{\text{method}\}) = \frac{1}{T} \sum_{t=1}^{T} |\hat{q}_t\{\text{method}\} - q_t\{\text{method}\}|, \]

(4.9)

where \( T = 1000 \) in this simulation.

The results are summarized in the following Table (4.1) and Table (4.2). Both methods have very similar performance in the estimation. Both have higher AAD in estimating 1% quantile than 5% quantile. The estimation with \( n = 500 \) has a big improvement over estimations with
$n = 250$, and the improvement is milder from $n = 500$ to $n = 1000$. When in the case $n = 500$ and $n = 1000$, the estimation of $\beta_3$ and $\beta_4$ is reasonably approximating to the true value. However, there is no implication that the estimation of $\beta_1$ and $\beta_2$ can be improved when $n$ increases. Another point we need to note is that CAViaR estimation requires an input of the initial value of the parameters and its estimation can be very different when given different intial value.
Table 4.1: Comparison of Estimation Performance between CAViaR and Linear Regression Spline at 5% VaR

<table>
<thead>
<tr>
<th></th>
<th>AAD</th>
<th>$\hat{\beta}_1 - \beta_1$</th>
<th>$\hat{\beta}_2 - \beta_2$</th>
<th>$\hat{\beta}_3 - \beta_3$</th>
<th>$\hat{\beta}_4 - \beta_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>se</td>
<td>mean</td>
<td>se</td>
<td>mean</td>
</tr>
<tr>
<td>n=250</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAViaR</td>
<td>0.678</td>
<td>0.017</td>
<td>-0.406</td>
<td>0.046</td>
<td>-0.092</td>
</tr>
<tr>
<td>Linear Spline</td>
<td>0.689</td>
<td>0.015</td>
<td>-0.550</td>
<td>0.039</td>
<td>-0.131</td>
</tr>
<tr>
<td>n=500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAViaR</td>
<td>0.460</td>
<td>0.010</td>
<td>-0.414</td>
<td>0.047</td>
<td>-0.104</td>
</tr>
<tr>
<td>Linear Spline</td>
<td>0.456</td>
<td>0.009</td>
<td>-0.520</td>
<td>0.037</td>
<td>-0.133</td>
</tr>
<tr>
<td>n=1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAViaR</td>
<td>0.315</td>
<td>0.007</td>
<td>-0.260</td>
<td>0.043</td>
<td>-0.066</td>
</tr>
<tr>
<td>Linear Spline</td>
<td>0.320</td>
<td>0.007</td>
<td>-0.450</td>
<td>0.036</td>
<td>-0.118</td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of Estimation Performance between CAViaR and Linear Regression Spline at 1% VaR

<table>
<thead>
<tr>
<th></th>
<th>AAD</th>
<th>$\hat{\beta}_1 - \beta_1$</th>
<th>$\hat{\beta}_2 - \beta_2$</th>
<th>$\hat{\beta}_3 - \beta_3$</th>
<th>$\hat{\beta}_4 - \beta_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>se</td>
<td>mean</td>
<td>se</td>
<td>mean</td>
</tr>
<tr>
<td>n=250</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAViaR</td>
<td>1.418</td>
<td>0.037</td>
<td>-0.413</td>
<td>0.068</td>
<td>-0.061</td>
</tr>
<tr>
<td>Linear Spline</td>
<td>1.443</td>
<td>0.036</td>
<td>-0.807</td>
<td>0.065</td>
<td>-0.126</td>
</tr>
<tr>
<td>n=500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAViaR</td>
<td>0.996</td>
<td>0.024</td>
<td>-0.532</td>
<td>0.067</td>
<td>-0.090</td>
</tr>
<tr>
<td>Linear Spline</td>
<td>1.021</td>
<td>0.022</td>
<td>-0.943</td>
<td>0.066</td>
<td>-0.154</td>
</tr>
<tr>
<td>n=1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAViaR</td>
<td>0.735</td>
<td>0.018</td>
<td>-0.437</td>
<td>0.066</td>
<td>-0.071</td>
</tr>
<tr>
<td>Linear Spline</td>
<td>0.748</td>
<td>0.017</td>
<td>-0.695</td>
<td>0.062</td>
<td>-0.111</td>
</tr>
</tbody>
</table>
Chapter 5

Selection of Smoothing Parameter

5.1 Introduction

Although a nonparametric estimation method can be free from assumptions on the data distribution or functional form, the goodness of the fit is still affected by the smoothing parameter, or tuning parameter. In regression spline, the smoothing parameters are the number of knots \( K \) and their locations, which determine the structural change in the shape of the fitted curve; with smoothing parameter, the smoothing parameter is \( \lambda \), which controls the smoothness of the fitted curve, or the dimension of the effective model. While in local linear models with kernel weighting, the smoothing parameter is \( h \), the window width that implicitly controls the number of observations to be included in the local linear fitting. In a word, the smoothing parameters determine the model dimension in nonparametric regressions.

Selection of smoothing parameters is a very critical step in nonparametric regressions. Although it can be sometimes selected subjectively according to the past studies or experiences, an automatic procedure of selecting smoothing parameters driven by data is still highly desirable. Then the question is what makes a good choice of the smoothing parameter. One could possibly allow the nonparametric regression model large enough so that the estimates will interpolate every observation: when \( K = N - 2 \), \( \lambda = 0 \) or \( h \) is small enough. Although this type of model has very good fit in the training data, they are too specified for the sample data and usually have poor performance in the prediction. After all, the purpose of fitting a nonparametric model is to discover the associations in the current data, and make further prediction or inference when new observations come in, so a good nonparametric model should have good prediction power. While this power can be evaluated by the expected prediction error:

\[
E_{\rho}(y - \hat{g}(x)),
\]

we define it as “Gold Criterion” (GC) because it directly measures the ultimate goal. In practice,
this value is not achievable because the true underlying distribution is unknown, but it can be approximated by Monte Carlo method as sampling an additional test data set (really large enough data set) which are distributed identically and independent from the training data. Then the GC of \( \hat{g} \) can be estimated by

\[
\sum_{i=1}^{N_{test}} \rho_{\tau}(y_{test} - \hat{g}(x_{test})) / N_{test}.
\] (5.1)

Though this value is usually not attainable, it can be used here as a reference to compare between different smoothing parameters selection methods. Another way to approximate the expected prediction error is cross-validation (CV), because it is a measure of the average prediction power when one observation is crossed out from the training data. However, this procedure is very time-consuming. In order to attenuate this issue, many criteria have been proposed to approximate CV, say AIC, BIC, ACV and GCV.

Since nonparametric method has attracted more and more interest in real application, many studies have been devoted into discussing the automatic selection of smoothing parameters. Compared to the mean regression settings, there is rather limited literature about smoothing parameter selection in quantile regression settings. Koenker et al. [1994] first suggested the SIC (Scharwz Information Criterion) to be used to select optimal \( \lambda \) in quantile smoothing spline and was commonly used by many following studies. He and Shi [1994] also suggested using SIC in regression B-spline estimation.

\[
\text{SIC} = \log\left( \frac{1}{N} \sum_i \rho_{\tau}(y_i - \hat{g}(x_i)) \right) + \frac{1}{2} R \log(N) / N, \tag{5.2}
\]

where \( R \) is the model dimension. Cai and Xu [2009] later proposed to select a bandwidth for the kernel smoothing estimation with an improved AIC (Akaike Information Criterion) (Hurvich et al. [1998]):

\[
\text{AIC} = \log\left( \frac{1}{N} \sum_i \rho_{\tau}(y_i - \hat{g}(x_i)) \right) + \frac{2(R + 1)}{N - (R + 2)}. \tag{5.3}
\]

The behavior of AIC and SIC is better studied for model selection in mean regression models. However, there is a lack of theoretical study on the properties of using the AIC or SIC in quantile regression. Both the criteria are very commonly seen in model selection. Both of them have a likelihood-like term to measure the goodness-of-fit, and they are usually different by the other term that accounts for the model dimension. In past study, the AIC will tend to select more parameters than SIC and lead to a more wiggly estimation, while SIC puts a stronger penalty on the model dimension and thus generates a smoother estimator. Machado [1993] showed that SIC criterion asymptotically selects the optimal model with probability 1 when the number of
observations goes to infinity from a limited number of candidate pool in M-estimators. Although
there is a gap between the SIC criterion in this paper and the one used by Koenker et al. [1994],
the goodness-of-fit part of the latter one is actually an analogue of the likelihood when the
data distribution is double exponential. In addition, because of the ease of computation and
previous success in numerous empirical analysis, SIC is still the most commonly used in model
selection of quantile models. A recent development, Yuan [2006], introduced the estimation of
GACV (Generalized Approximate Cross-Validation) for smoothing quantile. In that work, they
have compared the GACV with AIC and SIC in real data application and found that GACV
and SIC have very similar results in the model selection. The only problem with the GACV is
the intuitive explanation of this value when the data is time series type. Therefore, we suggest
using SIC, by Koenker et al. [1994] and He and Shi [1994] as the criterion to select the optimal
smoothing parameters.

5.2 Knot Selection in Regression Spline

5.2.1 Introduction

In past studies, the number of knots and knot locations have been recognized to greatly influence
the estimation and prediction of regression splines, because misspecified knots is equivalent to
misspecified curve structures and will lead to a biased estimation. Because deleting or adding
one knot leads to a different model (especially in B-spline regression, the Basis functions are all
influenced by the addition or deletion one knot), given $K$ knots, the knot selection procedure is
equivalent to model selection from $2^K$ candidate models, which is a very challenging problem
even when $K$ is a moderate value. Among many possible procedures that help avoid to directly
select from a pool of $2^K$ candidate models, there are two most popular methods for knot
selection:

1. Stepwise Selection

   First specify a maximum number $M$ (usually $M = 9$), then choose the optimal model from
   a pool of candidate models with $\{0,1,2,...,M\}$ number of internal knots that are uniformly
distributed in percentage or Euclidean distance. For example, a model with 9 internal
   knots uniformly distributed in percentage has the knots located at the 10%, 20%, ..., 90%
   quantiles. Then starting from this selected model, cycle the knot deletion and addition
   steps until the target value cannot be improved any further. If the target value is SIC,
   then the procedure will cycle until SIC cannot be reduced any further.

   (a) Step 1: Stepwise knot deletion. Deleting the knots sequentially that leads to the
   largest improvement in the target value (e.g. largest reduction in SIC), and repeat
the procedure until SIC does not decrease if removing any knot.

(b) Step 2: Stepwise knot addition. Consider the midpoint of two adjacent knots (in terms of percentile of Euclidean distance) as potential knots and add the one that improves the target value best. Repeat the procedure until no more knots can be added.

2. Penalized Shrinkage

Given truncated polynomial basis, the estimated function is

\[ \hat{g}(x) = \sum_{i=0}^{p} \alpha_i x^i + \sum_{k=1}^{K} \alpha_{pk} (x - \xi_k)^p_+ \]

A small absolute value of \( \alpha_{pk} \) corresponds to a small impact of \( k \)th knot in \( p \)the order. Penalizing on the parameters \( \alpha_{pk} \) is equivalent to penalizing on the smoothness of the fitted function, therefore, an idea to develop an optimal criterion is as follows,

\[ \rho_r(y_i - \hat{g}_\lambda(x_i)) + \gamma \sum_{k=1}^{K} \alpha_{pk}^2, \quad (5.4) \]

then \( \gamma \) controls the extent of smoothness. The selection of knots location is now reduced to the selection of \( \gamma \). Because this procedure is not instantly applied to B-spline basis, in the following discussion, we will continue to discuss using the Stepwise Selection for knot selection.

5.2.2 Selection in Recursive Model

As discussed in Chapter (4), the recursive model with the link function as a regression spline can be re-written as a modified regression spline. Therefore, the smoothing parameter selection methods in traditional regression spline can also be applied to the model selection of recursive model.

In terms of model dimension, the recursive model indeed increases the number of parameters. Specifically, the recursive model has additional lagged quantile variables to the non-recursive one, and therefore requires more parameters estimations. In calculating the effective model dimension, \( R \), we need to count in all estimated parameters. Except for the estimated parameters from regression spline, \( \theta \) and \( q_1 \) are also estimated in the model as parameters. For example, the model dimension for non-recursive linear regression spline regression is \( K + 2 \), where \( K \) is the number of internal knots. If added one lagged quantile variable to the above model, the model dimension is now \( K + 2 + 2 \).
In fact, SIC can also be used to identify the significant lagged variables in the model, because the selection of number of lagged variables is also a procedure of model selection.

5.2.3 Simulation Study

Current studies on smoothing parameter selections for nonparametric quantile regression mainly focus on quantile levels such as the median and quartiles, and the behavior of the criteria were examined given the predictor variables are uniformly or normally distributed. However, in our setting, the predictor and responses are time series data and our interest is to apply it to financial data. In fact, the distribution of financial data are usually asymmetric and fat-tailed, as shown in the scatter plot of (6.3). Obviously, we are facing the challenge that data at both ends of the range is very sparse and a lot of data points are clustered close to the median. Therefore, in this section, a simulation study will be carried out to evaluate the current smoothing parameter selection criteria under a setting close to the real-world data, say S&P 500.

The simulation data are generated as follows:

- We consider the quantile processes as

\[ q_t^k = 0.9 q_{t-1}^k + l_k(r_{t-1}), \]

and \( l_1 \) and \( l_2 \) are two different link functions:

\[ l_1(x) = [0.09 - 0.02xI(x < 0) + 0.01xI(x >= 0)] Q_{\xi}(\tau), \]
\[ l_2(x) = [0.28 - 0.18 \cos(\frac{\pi}{10} x)] Q_{\xi}(\tau), \]

so we have two different processes in this simulation study.

Accordingly, the observations \( \{r_t\} \) can be generated recursively as follows

\[ r_1 = \epsilon_1, \]
\[ q_1^k = Q_{\xi}(\tau), \]

for \( t \geq 2 \), and \( k = 1, 2 \)

\[ r_t = \{[0.9 q_{t-1}^k + l_k(r_{t-1})] / Q_{\xi}(\tau)\} \epsilon_t \]
\[ q_t^k = 0.9 q_{t-1}^k + l_k(r_{t-1}), \]

where \( \{\epsilon_t\} \), are independently generated from a skewed-t distribution (\( \mu = 0, \sigma = 2, \nu = 6.35, \xi = .92 \)).
In the graph (5.1) gives the scatterplot of a sample data and the function curves of 1\%(green dotted) and 5\%(red dashed) quantiles. Notice that \( l_1(x) \) is a piecewise linear function with the structure change point at 0 and the function is not symmetric about 0, and \( l_2(x) \) has more curvature than \( l_1(x) \) and is symmetric about 0. Therefore, our simulation result will cover both linear and nonlinear type of nonparametric functions and the comparisons will not be in favor of a particular model that is in nature closer to the underlying function. For example, the linear spline model may work better on estimating \( q_1 \) than natural cubic spline.

- For each process, generate 300 data sets with 1500 observations in each. The first 1000 observations are then used for training, called training data, while the remaining 500 observations are test data.

- With each training data, quantiles are estimated at both 1\% and 5\% level. Each fit is called a replicate, so this simulation has 300 replicates for each process. The test data is then used for choosing the best smoothing parameters by the Gold Criterion.

- For each replicate, use the Gold Criterion (GC), SIC and AIC separately to choose the number of knots and knot locations, and fit 3 different models corresponding to the selected knots by different criteria.

- The goodness of fit is measured by the Average Absolute Deviation (AAD) of estimation from the true quantile values,

\[
AAD(\hat{q}^k\{\text{knots}\}) = \frac{1}{T} \sum_{t=1}^{T} |\hat{q}^k_t\{\text{knots}\} - q^k_t\{\text{knots}\}|, \quad (5.5)
\]

where \( k = 1, 2 \) and \( T = 1000 \) in this simulation.

All the simulation results are summarized in the following tables (5.1)-(5.2). Using the knots selected by GC gives the smallest AAD among the three methods, which means that the estimated quantiles are closest to the truth on average. At the 5\% level, SIC works very similarly as GC and tends to choose fewer knots than GC, but at 1\% level, SIC works slightly worse than GC. In all cases, SIC works consistently better than AIC. For all methods, the AAD of estimation at 1\% level are higher than at 5\% level. To estimate 1\% quantile is harder because we have higher variability in the 1\% sample quantiles. From the results, we can conclude that in the recursive quantile regression settings like in this work, SIC is a better selection criterion than AIC, but the differences may be diminished when estimating extreme quantiles, i.e. 0.5\% or 1\% quantiles.
Figure 5.1: The Scatter Plot of Simulated Data by $l_1$ and $l_2$ and Link Functions
Table 5.1: Knot Selections Results w/ Different Criteria at 5% Level

<table>
<thead>
<tr>
<th>Process by $l_1(x)$</th>
<th>mean(AAD)</th>
<th>se(AAD)</th>
<th>Avg # of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC</td>
<td>0.394</td>
<td>0.009</td>
<td>3.24</td>
</tr>
<tr>
<td>SIC</td>
<td>0.459</td>
<td>0.012</td>
<td>2.66</td>
</tr>
<tr>
<td>AIC</td>
<td>0.556</td>
<td>0.012</td>
<td>5.16</td>
</tr>
<tr>
<td>GC</td>
<td>0.339</td>
<td>0.007</td>
<td>1.97</td>
</tr>
<tr>
<td>SIC</td>
<td>0.394</td>
<td>0.009</td>
<td>1.58</td>
</tr>
<tr>
<td>AIC</td>
<td>0.424</td>
<td>0.009</td>
<td>2.73</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process by $l_2(x)$</th>
<th>mean(AAD)</th>
<th>se(AAD)</th>
<th>Avg # of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC</td>
<td>0.644</td>
<td>0.013</td>
<td>3.23</td>
</tr>
<tr>
<td>SIC</td>
<td>0.707</td>
<td>0.017</td>
<td>2.76</td>
</tr>
<tr>
<td>AIC</td>
<td>0.841</td>
<td>0.016</td>
<td>4.85</td>
</tr>
<tr>
<td>GC</td>
<td>0.631</td>
<td>0.011</td>
<td>2.31</td>
</tr>
<tr>
<td>SIC</td>
<td>0.640</td>
<td>0.012</td>
<td>2.40</td>
</tr>
<tr>
<td>AIC</td>
<td>0.674</td>
<td>0.012</td>
<td>3.19</td>
</tr>
</tbody>
</table>

*The standard errors of all pairwise mean AAD differences within the same estimation method have range 0.008-0.013 for $l_1$, and 0.012-0.016 for $l_2$.*
Table 5.2: Knot Selections Results w/ Different Criteria at 1% Level $^a$

<table>
<thead>
<tr>
<th>Process by $l_1(x)$</th>
<th>mean(AAD)</th>
<th>se(AAD)</th>
<th>Avg # of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC</td>
<td>0.778</td>
<td>0.018</td>
<td>1.70</td>
</tr>
<tr>
<td>SIC</td>
<td>1.122</td>
<td>0.021</td>
<td>4.54</td>
</tr>
<tr>
<td>AIC</td>
<td>1.659</td>
<td>0.020</td>
<td>5.38</td>
</tr>
<tr>
<td>GC</td>
<td>0.767</td>
<td>0.019</td>
<td>1.53</td>
</tr>
<tr>
<td>SIC</td>
<td>1.040</td>
<td>0.019</td>
<td>4.28</td>
</tr>
<tr>
<td>AIC</td>
<td>1.098</td>
<td>0.019</td>
<td>4.86</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process by $l_2(x)$</th>
<th>mean(AAD)</th>
<th>se(AAD)</th>
<th>Avg # of Knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC</td>
<td>1.365</td>
<td>0.023</td>
<td>1.86</td>
</tr>
<tr>
<td>SIC</td>
<td>1.752</td>
<td>0.031</td>
<td>4.23</td>
</tr>
<tr>
<td>AIC</td>
<td>1.832</td>
<td>0.040</td>
<td>4.91</td>
</tr>
<tr>
<td>GC</td>
<td>1.370</td>
<td>0.026</td>
<td>1.87</td>
</tr>
<tr>
<td>SIC</td>
<td>1.606</td>
<td>0.029</td>
<td>4.18</td>
</tr>
<tr>
<td>AIC</td>
<td>1.697</td>
<td>0.029</td>
<td>4.99</td>
</tr>
</tbody>
</table>

$^a$The standard errors of all pairwise mean AAD differences within the same estimation method have range 0.012-0.021 for $l_1$ and 0.016-0.032 for $l_2$. All pairwise difference are significant.
5.3 Selection of $\lambda$ in Smoothing Spline

5.3.1 Introduction

The selection of $\lambda$ in smoothing spline is a simpler procedure than the knots selections procedure for regression spline because it only involves an optimization of one parameter in $(0, +\infty)$. Even in additive smoothing spline models, the optimization is to search over $(0, +\infty)_1 \times (0, +\infty)_2 \times \ldots \times (0, +\infty)_d$. In mean smoothing splines, GCV (Generalized Cross Validation) is often used as the criterion to search for the optimal $\lambda$ because of its computational efficiency. However, in quantile regression, GCV is not motivated. Though later Yuan [2006] proposed GACV (Generalized Approximate Cross-Validation) for quantile regression, the approximation is not working very well when estimating extreme quantiles. Instead, SIC (Schwarz Information Criterion) proposed in Koenker et al. [1994] was widely applied in many VaR estimations. Its definition is given in equation (5.2) while the model dimension $R = p_\lambda$ is the number of interpolated data points, also called active knots. He et al. [1998] extended the usage of SIC to bivariate smoothing splines by assuming equal smoothing parameter for each component, $\lambda = \lambda_1 = \lambda_2$.

The function of SIC is a step function of $\lambda$. It is desirable to examine the plot of $SIC$ vs. $\lambda$ in order to avoid selecting too small or too large value of $\lambda$ that minimizes $SIC$. If that happens, the second minimizer can be considered.

5.3.2 Selection in Recursive Model

Similarly to the recursive quantile regression spline model, the recursive quantile smoothing spline model can also be re-written as a modified modified smoothing spline, and the smoothing parameter selection methods in traditional smoothing spline can be naturally applied to the recursive model. In terms of model dimension, we should consider not only the model dimension in smoothing spline, but also the additional parameters $\theta$ and $q_1$, so $R = $ number of interpolated data points + 2.

5.3.3 Simulation Study

Because of the lack of study in evaluating the smoothing parameter selection methods in recursive VaR settings, we are motivated to run a simulation to validate its performance in this particular situation.

The data generation of the simulation study is the same as described in Section (5.2.3). The quantiles are estimated at both 5% and 1% levels using the selected $\lambda$ by GC, SIC and AIC respectively. The goodness of fit of the estimations is measured as AAD, similarly to equation...
(5.5),
\[
AAD(\hat{q}^k(\lambda)) = \frac{1}{T} \sum_{t=1}^{T} |\hat{q}^k_t(\lambda) - q^k_t(\lambda)|/T,
\]
where \(k = 1, 2\) and \(T = 1000\). The results are summarized in the following table (5.3) - (5.4). At both 5\% and 1\% level, GC selects largest \(\lambda\) on average, which penalizes the smoothness part the most, and also has the smallest mean AAD among all three methods. SIC works slightly worse than GC but consistently slightly better than AIC. Not surprisingly, for all criterion methods, AAD for the process generated by \(l_2(x)\) is higher than \(l_1(x)\), because \(l_1(x)\) has a piecewise linear link function, which is the same form as smoothing spline estimations. Similar to the simulation results in section (5.2.3), the mean AAD at 1\% level is higher than at 5\% level. From the results, we may conclude that the estimation using \(\lambda\) selected by SIC is better than AIC, and the estimation performance is strongly affected by the quantile level, that is, estimating 1\% quantile is harder and should require more data than estimating 5\% quantile.

Among the estimations of regression splines and smoothing splines using the smoothing parameters selected by GC, smoothing spline method has consistently lowest AAD on average in all scenarios. However, by using the the SIC or AIC method to select the smoothing parameter, we do not observe the same result, so no single estimation method can beat the others at all times by SIC or AIC.
Table 5.3: Lambda Selection Results w/ Different Criteria at 5% Level $^a$

<table>
<thead>
<tr>
<th></th>
<th>mean(AAD)</th>
<th>se(AAD)</th>
<th>mean($\lambda$)</th>
<th>avg # of active knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process by $l_1(x)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GC</td>
<td>0.341</td>
<td>0.008</td>
<td>5.81</td>
<td>3.82</td>
</tr>
<tr>
<td>SIC</td>
<td>0.416</td>
<td>0.012</td>
<td>4.55</td>
<td>4.19</td>
</tr>
<tr>
<td>AIC</td>
<td>0.515</td>
<td>0.018</td>
<td>3.83</td>
<td>10.73</td>
</tr>
<tr>
<td>Process by $l_2(x)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GC</td>
<td>0.592</td>
<td>0.009</td>
<td>6.01</td>
<td>5.04</td>
</tr>
<tr>
<td>SIC</td>
<td>0.722</td>
<td>0.015</td>
<td>4.55</td>
<td>2.88</td>
</tr>
<tr>
<td>AIC</td>
<td>0.973</td>
<td>0.030</td>
<td>3.35</td>
<td>16.30</td>
</tr>
</tbody>
</table>

$^a$The MAD of all pairwise mean AAD differences range 0.011-0.018 for $l_1$ and 0.014-0.031 for $l_2$. All pairwise difference are significant.

Table 5.4: Lambda Selection Results w/ Different Criteria at 1% Level $^a$

<table>
<thead>
<tr>
<th></th>
<th>mean(AAD)</th>
<th>se(AAD)</th>
<th>mean($\lambda$)</th>
<th>avg # of active knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process by $l_1(x)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GC</td>
<td>0.687</td>
<td>0.015</td>
<td>5.82</td>
<td>1.38</td>
</tr>
<tr>
<td>SIC</td>
<td>0.990</td>
<td>0.029</td>
<td>3.63</td>
<td>4.50</td>
</tr>
<tr>
<td>AIC</td>
<td>1.185</td>
<td>0.036</td>
<td>3.18</td>
<td>9.28</td>
</tr>
<tr>
<td>Process by $l_2(x)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GC</td>
<td>1.257</td>
<td>0.021</td>
<td>5.49</td>
<td>2.60</td>
</tr>
<tr>
<td>SIC</td>
<td>1.742</td>
<td>0.045</td>
<td>3.47</td>
<td>5.24</td>
</tr>
<tr>
<td>AIC</td>
<td>1.982</td>
<td>0.056</td>
<td>2.93</td>
<td>12.44</td>
</tr>
</tbody>
</table>

$^a$The MAD of all pairwise mean AAD differences have range 0.028-0.037 for $l_1$ and 0.042-0.056 for $l_2$. All pairwise difference are significant.
Chapter 6

Model Evaluation

6.1 Introduction

Many VaR models have already been proposed and applied to measure financial risk in many cases, and a question has come to one’s attention that whether a model correctly covers the risk at the expected level. In practice, risk managers do need to check regularly if their current VaR models perform well enough and compare them to alternative models for any possible improvements. The procedure used to evaluate the VaR models is usually backtesting, that is, to predict the VaRs of some past days assuming the actual observations are unknown, and then compare the predictions against the real data. Backtesting VaR is to evaluate the occurrence of exceptions, which is defined as loss exceeding the predicted VaR. In assumption of a sufficient model, the occurrence of exceptions should be at the frequency level as expected and the occurrence should be independent. To test these two properties, several procedures have been developed and their definitions are as follows:

1. Unconditional Coverage

   Unconditional coverage test, proposed by Kupiec [1995], is simple and straightforward to test whether the frequency of exceptions is consistent with the expectation. The event of exception can be expressed as a indicator function:

   \[
   I_t = \begin{cases} 
   1 & r_t \leq -VaR_t \\
   0 & r_t > -VaR_t 
   \end{cases},
   \]

   where \(VaR_t = -q_t\). Therefore, given that the model is correct for \(\tau\)-th quantile (VaR at \(\tau\) confidence level), the exceptions are independent Bernoulli variables, then the number of exceptions, \(N\), should follow Binomial distribution (\(T, p = \tau\)). The likelihood ratio
test can be constructed under $H_0 : p = \tau$ as follows:

$$LR_{uc} = -2\ln \left( \frac{\tau^N(1-\tau)^{T-N}}{\hat{p}^N(1-\hat{p})^{T-N}} \right) \sim \chi^2(1),$$

where $\hat{p} = N/T$, called as nominal coverage, is the maximum likelihood estimate of $p$ under alternative.

2. Independence and Conditional Coverage

Instead of focusing on the frequency of exceptions only, Christoffersen [1998] suggested a test on independence of occurrence because the exceptions should be independent of the previous day if the exception occurred or not. Therefore, the test is based on the likelihood with transition probabilities, that is,

$$L(\Pi; I_1, I_2, ..., I_T) = \pi_{00} \pi_{01} \pi_{10} \pi_{11}, \hspace{1cm} (6.1)$$

where

$$\Pi = \begin{bmatrix} \pi_{00} & \pi_{01} \\ \pi_{10} & \pi_{11} \end{bmatrix},$$

and $\pi_{ij} = P(I_t = j|I_{t-1} = i)$. The maximum likelihood estimates of the parameters are

$$\hat{\Pi} = \begin{bmatrix} \frac{n_{00}}{n_{00}+n_{01}} & \frac{n_{01}}{n_{00}+n_{01}} \\ \frac{n_{10}}{n_{10}+n_{11}} & \frac{n_{11}}{n_{10}+n_{11}} \end{bmatrix}.$$

By claiming that testing $E[I_t|I_{t-1}, I_{t-2}, ..., I_1] = \pi$ is equivalent to testing independence, they were able to construct an LR-ratio test with the null hypothesis that $\pi_{01} = \pi_0$ and $\pi_{11} = \pi_0$. The maximum likelihood under $H_0$ is

$$\hat{\Pi}_0 = \begin{bmatrix} 1 - \hat{\pi}_0 & \hat{\pi}_0 \\ 1 - \hat{\pi}_0 & \hat{\pi}_0 \end{bmatrix},$$

where $\hat{\pi}_0 = (n_{01} + n_{11})/(n_{00} + n_{10} + n_{01} + n_{11})$, then

$$LR_{ind} = -2\log[L(\hat{\Pi}_0; I_1, I_2, ..., I_T)/L(\hat{\Pi}; I_1, I_2, ..., I_T)] \sim \chi^2(1). \hspace{1cm} (6.2)$$

To jointly test independence and correct coverage, a LR test can be used as follows:

$$LR_{cc} = LR_{uc} + LR_{ind} \sim \chi^2(2).$$

Later, Christoffersen and Pelletier [2004] further considered the time between two excep-
tions and proposed a test based on duration, because ideally the exceptions should be independently and uniformly distributed over time and any clustering of exceptions is an indication that the model is not efficient enough.

3. Dynamic Quantile

As the aforementioned tests are totally model free, Engle and Manganelli [2004] proposed a new test that fits a linear model on the hit function, which defined as $\text{Hit}_t = I_t - \tau$, but will be able to test on the independence of the exceptions with other possible influential factors, e.g. lagged quantile estimates. They consider the model

$$\text{Hit}_t = Z_t' \delta + u_t, \quad u_t = \begin{cases} -\tau & \text{with prob. } 1 - \tau \\ 1 - \tau & \text{with prob. } \tau \end{cases},$$

where $Z_t$ consist of the potential correlated factors. In order to test unconditional coverage, one needs to include an intercept in $Z_t$; to test independence of exceptions on the past information, one can choose to include $\{I_{t-1}, I_{t-2}, \ldots\}$ or $\{\hat{q}_{t-1}, \hat{q}_{t-2}, \ldots\}$. (Berkowitz et al. [2007] found that the DQ test using the lagged VaR estimates (lagged $\hat{q}$) and lagged $I$ variables has good testing power.) Then the parameters $\delta$s are jointly tested against 0.

Out-of-sample dynamic quantile (DQ) test has the form

$$\text{DQ}_{OOS} = \frac{\text{Hit}' Z[Z'Z]^{-1} Z' \text{Hit}}{N_{OOS} \cdot \tau(1 - \tau)} \sim \chi^2(r),$$

(6.3)

where $r$ is the number of factors in the model and $N_{OOS}$ is the number of of out-of-sample observations.

While VaR backtesting has considered the occurrence of exceptions, the actual loss beyond the VaR estimates at exceptions are not tested, which is a very important metric in measuring risk. However, the loss beyond VaR is in fact related to another risk measure, expected shortfall. As mentioned in the beginning of Chapter 1, this work only focuses on modeling the Value-at-Risk because the expected shortfall can be also estimated once VaR can be estimated. Therefore, in this work, we will use only the backtest on VaRs introduced above in the following section to evaluate the several existing VaR models and the nonparametric recursive VaR model proposed by us.

6.2 Empirical Data

This section will apply the aforementioned additive nonparametric quantile regression techniques to predict the 1% VaR and 5% VaR with real data from S&P 500 and IBM stock Price
from 12/01/2004 to 11/15/2010. Figure (6.1) are their trend graphs of the daily close market value during that period, and figure (6.2) are their daily log returns in percentage according to the Close Price vs. time. This period is particularly of interest because it includes a relatively more tranquil time in 2005 and 2006, and more volatile period from later half of 2007 to first half of 2008, followed by a super dynamic period from later 2008 to later 2009. In the study, a window frame of 1000 observations is used for fitting models and predicting 1-step ahead 1% VaR and 5% VaR. As moving the window along the timeline, we fit a new model, allowing the model estimates can vary over time. In this study, we focus on examining the one-step ahead predictions of 500 days from 11/20/2008 to 11/15/2010, which is the period to the right of the vertical green line on figure (6.2). (Note that this model can also be used for n-days ahead prediction, if regressing on the lag-n observation.) On this figure, we can tell the difference between these two data sets: the volatility of S&P500 during the tranquil period (2005-2007) is lower than IBM stock price, but when entering into late 2008, their volatilities are very close. That is to say, in the scatter plots of the log returns vs. the 1-lag log returns (see figure (6.3)), the scatter plot for S&P500 has denser clusters of data points in the middle of the plot than IBM’s, and the data points at the both ends are more sparse. We are interested in investigating whether the change in the data distribution would have effects in the performance of the VaR models.
Figure 6.1: Daily Close Price for S&P 500 and IBM Stock Price 12/01/2004 - 11/15/2010
Figure 6.2: Daily Log Returns on Close Price for S&P 500 and IBM Stock
Figure 6.3: Log Return of Today vs. Previous Day
We chose 3 other very popular VaR models as competitors to our models: ARMA-GARCH with skewed t-distributed errors, CaViaR model with Asymmetric Slope and Indirect GARCH. ARMA-GARCH is a very classical recursive time series model that has been extensively studied and proved to be successful in estimating time series means, so its efficiency in estimating the quantiles is of our interest. The CaViaR type models with different link function types should be the major competitors to our models, because the current types are mostly parametric, and one of purposes of this work is to find out whether the flexibility of nonparametric formation can add benefits in the quantile estimation. On the other side, for our nonparametric quantile regression model, we have the estimations from the models where the lagged-one log return and lagged-one quantile value are included as explanatory variables. The smoothing parameter selection procedures follow the same steps as described in Chapter (5). The testing metrics are derived as the same procedures as in the previous section. Note that DQ test includes the intercept, hit variable and the estimated VaR in the design matrix for regression.

Following are the tables presenting the p-values from the backtesting procedures (Unconditional Coverage, Independence, Conditional Coverage, Dynamic Quantile). Table (6.1) includes the results for backtesting the predictions on 5% VaR, and Table (6.2) on 1% VaR.

At the 5% level, all three nonparametric methods have competitive estimated exception rate around or exactly at the expected level 0.05. The parametric models have higher exception rate on S&P 500 data and the lower rate on IBM data. In the independence test, the regression splines, i.e. linear regression spline and natural cubic regression spline, have lower p-values compared to the parametric models, but in terms of conditional coverage, which jointly tests the unconditional coverage and independence, the 5% VaR estimates by regression splines on IBM stock price are comparable to the parametric methods. And both linear and natural cubic regression splines do not have impressive p-values in passing the DQ test. After all, the smoothing spline estimation passes all the tests with highest or second highest p-values among all methods on both S&P500 and IBM stock data.

In theory, the nonparametric model linear regression splines should be a more general class of models that includes the asymmetric slope. Note that the asymmetric slope is a special linear regression spline with a single knot at 0. By searching for the optimal model within the class, we expect to find a more appropriate one. Unfortunately, in the real data application, such improvement is not obvious at least according to these two data example. One of the advantages of the nonparametric method is that one does not need to make assumption on the form of the link function. The shape of the function can be automatically generated after selecting the optimal smoothing parameter. The plots of link functions for predicting 5% VaR at 6 different time points are presented in figure (6.12) - figure (6.15). All the plots confirm that the 5% VaR are mostly affected by loss in the previous day. Some plots suggest that the structural change of the link function may not be always located at 0.
At the 1% level, the coverage rate of nonparametric regression splines is a little far off the expected level 0.01, while the parametric models are around 0.01. In the independence test, the regression spline methods have comparable p-values as the parametric ones. However, the linear regression spline does not pass the DQ test on IBM data, which is mainly because of the poor uncoverage rate in exceptions, 0.018. Referring to the estimated link functions in figure (6.16) - figure (6.18), we can see that the estimation is more wiggly than in 5% VaR case in linear regression spline. Its poor performance in unconditional coverage test may be due to the lack of penalty in smoothing, that is, SIC is not strong enough in penalizing the smoothness.

In summary, the parametric methods may have different performance according to different data, while nonparametric methods, in advantage of being adaptive, are more consistent on the performance. With these two real data samples, the parametric CAViaR models have proved to be sufficient in predicting VaR, but the concerns still remain as the performance may be poor with some other data samples. We may be interested to see how the nonparametric method performs in the cases when the parametric ones fails.

Table 6.1: Estimated Exception Rate and P-values of Backtesting on Predicting 5% VaR

<table>
<thead>
<tr>
<th></th>
<th>τ</th>
<th>UC</th>
<th>Ind</th>
<th>CC</th>
<th>DQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>S&amp;P 500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARMA(1,1)-GARCH(1,1)</td>
<td>0.064</td>
<td>0.168</td>
<td>0.413</td>
<td>0.276</td>
<td>0.440</td>
</tr>
<tr>
<td>Asymmetric Slope</td>
<td>0.056</td>
<td>0.546</td>
<td>0.607</td>
<td>0.730</td>
<td>0.590</td>
</tr>
<tr>
<td>Indirect GARCH(1,1)</td>
<td>0.054</td>
<td>0.685</td>
<td>0.671</td>
<td>0.842</td>
<td>0.837</td>
</tr>
<tr>
<td>Linear Regression Spline</td>
<td>0.046</td>
<td>0.678</td>
<td>0.136</td>
<td>0.302</td>
<td>0.328</td>
</tr>
<tr>
<td>Natural Cubic Regression Spline</td>
<td>0.050</td>
<td>1.000</td>
<td>0.104</td>
<td>0.267</td>
<td>0.386</td>
</tr>
<tr>
<td>Smoothing Spline(p=1)</td>
<td>0.050</td>
<td>1.000</td>
<td>0.806</td>
<td>0.970</td>
<td>0.603</td>
</tr>
<tr>
<td>IBM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GARCH(1,1)</td>
<td>0.034</td>
<td>0.082</td>
<td>0.273</td>
<td>0.121</td>
<td>0.227</td>
</tr>
<tr>
<td>Asymmetric Slope</td>
<td>0.040</td>
<td>0.288</td>
<td>0.196</td>
<td>0.247</td>
<td>0.626</td>
</tr>
<tr>
<td>Indirect GARCH(1,1)</td>
<td>0.036</td>
<td>0.131</td>
<td>0.246</td>
<td>0.163</td>
<td>0.443</td>
</tr>
<tr>
<td>Linear Regression Spline</td>
<td>0.052</td>
<td>0.838</td>
<td>0.091</td>
<td>0.234</td>
<td>0.201</td>
</tr>
<tr>
<td>Natural Cubic Regression Spline</td>
<td>0.046</td>
<td>0.678</td>
<td>0.136</td>
<td>0.302</td>
<td>0.477</td>
</tr>
<tr>
<td>Smoothing Spline(p=1)</td>
<td>0.048</td>
<td>0.836</td>
<td>0.878</td>
<td>0.967</td>
<td>0.843</td>
</tr>
</tbody>
</table>
Table 6.2: Estimated Exception Rate and P-values of Backtesting on Predicting 1% VaR

<table>
<thead>
<tr>
<th></th>
<th>( \hat{\tau} )</th>
<th>UC</th>
<th>Ind</th>
<th>CC</th>
<th>DQ</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S&amp;P 500</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARMA(1,1)-GARCH(1,1)</td>
<td>0.008</td>
<td>0.641</td>
<td>0.799</td>
<td>0.869</td>
<td>0.566</td>
</tr>
<tr>
<td>Asymmetric Slope</td>
<td>0.010</td>
<td>1.000</td>
<td>0.750</td>
<td>0.951</td>
<td>0.909</td>
</tr>
<tr>
<td>Indirect GARCH(1,1)</td>
<td>0.010</td>
<td>1.000</td>
<td>0.750</td>
<td>0.951</td>
<td>0.666</td>
</tr>
<tr>
<td>Linear Regression Spline</td>
<td>0.016</td>
<td>0.215</td>
<td>0.610</td>
<td>0.407</td>
<td>0.238</td>
</tr>
<tr>
<td>Natural Cubic Regression Spline</td>
<td>0.014</td>
<td>0.397</td>
<td>0.655</td>
<td>0.632</td>
<td>0.382</td>
</tr>
<tr>
<td>Smoothing Spline(p=1)</td>
<td>0.008</td>
<td>0.641</td>
<td>0.799</td>
<td>0.869</td>
<td>0.800</td>
</tr>
<tr>
<td><strong>IBM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GARCH(1,1)</td>
<td>0.008</td>
<td>0.641</td>
<td>0.799</td>
<td>0.869</td>
<td>0.620</td>
</tr>
<tr>
<td>Asymmetric Slope</td>
<td>0.016</td>
<td>0.215</td>
<td>0.610</td>
<td>0.407</td>
<td>0.417</td>
</tr>
<tr>
<td>Indirect GARCH(1,1)</td>
<td>0.014</td>
<td>0.397</td>
<td>0.655</td>
<td>0.632</td>
<td>0.761</td>
</tr>
<tr>
<td>Linear Regression Spline</td>
<td>0.018</td>
<td>0.106</td>
<td>0.565</td>
<td>0.230</td>
<td>0.004</td>
</tr>
<tr>
<td>Natural Cubic Regression Spline</td>
<td>0.014</td>
<td>0.397</td>
<td>0.655</td>
<td>0.632</td>
<td>0.760</td>
</tr>
<tr>
<td>Smoothing Spline(p=1)</td>
<td>0.014</td>
<td>0.397</td>
<td>0.655</td>
<td>0.632</td>
<td>0.454</td>
</tr>
</tbody>
</table>

In the following graphs, figure (6.4) - (6.7) are the one-step ahead 5% VaR and 1% VaR predictions from 11/20/2008 to 11/15/2010 on S&P 500, while figure (6.8) - (6.11) are the predictions on IBM stock. The circles represent the actual observations for each day, and the red and blue lines are the 1-step ahead predictions, while the vertical bars indicate the time when the exceptions occurred.
Figure 6.4: One-step Ahead Predictions of 5% VaR(quantile) on S&P 500 Daily Log Returns
Figure 6.5: One-step Ahead Predictions of 5% VaR(quantile) on S&P 500 Daily Log Returns
Figure 6.6: One-step Ahead Predictions of 1% VaR(quantile) on S&P 500 Daily Log Returns
Figure 6.7: One-step Ahead Predictions of 1% VaR(quantile) on S&P 500 Daily Log Returns
Figure 6.8: One-step Ahead Predictions of 5% VaR(quantile) on IBM Stock Daily Log Returns
Figure 6.9: One-step Ahead Predictions of 5% VaR(quantile) on IBM Stock Daily Log Returns
Figure 6.10: One-step Ahead Predictions of 1% VaR(quantile) on IBM Stock Daily Log Returns
Figure 6.11: One-step Ahead Predictions of 1% VaR(quantile) on IBM Stock Daily Log Returns
Figure 6.12: Estimated Link Functions by Asymmetric Slope at 5% VaR on S&P 500 Data
Figure 6.13: Estimated Link Functions by Linear Regression Spline at 5% VaR on S&P 500 Data
Figure 6.14: Estimated Link Functions by Natural Cubic Spline at 5% VaR on S&P 500 Data
Figure 6.15: Estimated Link Functions by Smoothing Spline at 5% VaR on S&P 500 Data
Figure 6.16: Estimated Link Functions by Asymmetric Slope at 1% VaR on S&P 500 Data
Figure 6.17: Estimated Link Functions by Linear Regression Spline at 1% VaR on S&P 500 Data
Figure 6.18: Estimated Link Functions by Natural Cubic Regression Spline at 1% VaR on S&P 500 Data
Chapter 7

Summary

7.1 Results

In summary, a model extended from CAViaR and TS-GARCH has been discussed for quantile estimations, i.e. VaR estimation. A selection of nonparametric methods, regression spline, smoothing spline and local linear regression are reviewed and discussed in the estimation and their extension to additive model in multivariate regression.

Local linear regression with kernel weighting has the most flexibility in function estimation because each data points are fitted locally, but the calculation effort is more demanding and there is no obvious way to adopt it in estimating a recursive model. In addition, the back-fitting algorithm used in the additive model has no guarantee to estimation convergence.

Splines in definition force constraints on the global structure of the fitted curve, so they may not cover as wide function classes as local linear regression but the computation is efficient and consistent. With the advantage that their estimation problems can be transformed into solving linear programming problems, they can be used to estimate the recursive type of models.

A Monte-Carlo simulation study is implemented to evaluate the criteria for selecting the best smoothing parameters in the recursive nonparametric models. The results suggest that SIC is consistently better than improved-AIC in our scenario: extreme VaR (5% and 1% level) estimation on data with fat-tailed distribution. At 5% level, SIC can work almost as well as GC.

The real data examples have demonstrated that our proposed recursive model estimations do at least as well as the existing VaR models. In addition to that, the nonparametric model provides a favor in exploring the underlying link function from a larger family than the existing models.
7.2 Future Work

As mentioned at the beginning of the work, there are several risk measures, among which Expected Shortfall is one coherent measure. Expected Shortfall includes information that Value-at-Risk does not, but the derivation of conditional Expected Shortfall from VaR is applicable by using the following correlation equation.

\[ ES_\tau(t|I_{t-1}) = VaR_\tau(t|I_{t-1}) + \frac{1}{\tau} E[(r_t + VaR_\tau(t) + |I_{t-1})], \]

where the expectation part can be estimated by mean regression of \( r_t - \hat{VaR}_\tau(t) \) on past observations.

In risk management, the magnitude of exceptions is also a very important measure, because when loss exceeds the estimated VaR, the magnitude can be very large and it is desirable to control the difference between the observations and the estimated VaR. While the magnitude cannot be used to evaluate the VaR estimates, it can be used to evaluate ES estimates. Lopez [1998] proposed one method to incorporate the information of frequency of exceptions and their magnitude by the summation of the values of the following variable over time,

\[ C_t = \begin{cases} 1 + (y_t - q_t)^2 & \text{if } y_t < q_t \\ 0 & \text{o.w.} \end{cases} \]

where the errors are penalized by a squared function.

Like other quantile estimations, our method also may have the occasions when the estimated quantiles would cross over, although the issue is not very severe in our setting, because in the real financial data application, we observed that the lagged quantile explains the majority of the trend of the current quantile function. One may still want to address for the cross-over issue.

Within a window of 1000 observations, we expect to see around only 10 exceptions. In nonparametric regression, this is actually a relatively small number and could be one of the reasons that leads to an under smoothing link function estimation. However, increasing the window length may not be appropriate in the financial data as the quantile function may have been changing over a long run.
REFERENCES


Appendix A

Codes of Modified rqss Functions

#This code modifies the quantile smoothing spline functions
#in quantreg package to estimate the recursive quantile
#regression model when u=1 and v=1.

library(quantreg)

# Given a time series data \{r_t\}, y is the current obs
# and x is the lag-1 day obs

make_newy = function(x, y){ #make "y" a new "y" to be used as the response
sx=sort(x)
loc=which(sx[-length(sx)]==sx[-1])
add_len=length(loc)
c(y, rep(0, add_len))
}

## myqss1 generates the design matrix for rqss fit, to be used in myrqss
myqss1<-function (x, constraint = "N", lambda = 1, dummies = dummies,
theta = theta, ndum = 0, w = rep(1, length(x)))
{
xun <- unique(x[order(x)])
h <- diff(xun)
NH <- length(h)


nx <- length(x)
p <- nh + 1
B <- new("matrix.csr", ra = c(rbind(-1/h, 1/h)),
ja = as.integer(c(rbind(1:nh, 2:(nh + 1))))),
ia = as.integer(2 * (1:(nh + 1)) - 1),
dimension = as.integer(c(nh, nh + 1)))
makeD <- function(p) {
new("matrix.csr", ra = c(rbind(rep(-1, (p - 1)), rep(1,(p - 1))))),
ja = as.integer(c(rbind(1:(p - 1), 2:p))),
ia = as.integer(2 * (1:p) - 1),
dimension = as.integer(c(p - 1, p)))
}
D <- makeD(nh)
A <- D %*% B
# up to this point, the design matrix is the same as in the original rqss
if (length(xun)<nx) { # this part adjusts the design matrix for
sx = sort(x) #identical values in x
num_same = function(u){sum(sx==u)}
um = sapply(xun,num_same)
loc_same = c(which(num>=2),0)
Anew = as.matrix(A[,1])
for (k in 1:ncol(A)) {
Anew = cbind(Anew,
matrix(rep(as.matrix(A[,k]/num[k]),num[k]), ncol=num[k]))
}
A=Anew[,,-1]
}

ox = order(x) #this part does the column manipulation in the
rankx = 1 #design matrix for estimating recursive regression
rankx[ox] = 1:nx
A <- A[,rankx]
A <- cbind( A[,-nx]-theta*A[,,-1] , A[,nx])
A <- as.matrix.csr(A)
design = diag(1,nrow=nx)

if (length(xun)<nx) {
loc = which(sx[-length(sx)]==sx[-1])
add = matrix(0, nrow=length(loc), ncol=nx)
for (i in 1:length(loc)){
  add[i,loc[i]] = 1
  add[i,loc[i]+1] = -1
}
add <- add[,rankx]
if (length(loc)==1) {
  add <- c( add[-nx]-theta*add[-1] , add[nx])
} else {
  add <- cbind( add[-nx]-theta*add[-1] , add[,nx])
}
design = rbind(as.matrix(design), add)
}
F<-as.matrix.csr(design)

switch(constraint, V = {
  R <- A
  r <- rep(0, nrow(R))
}, C = {
  R <- -A
  r <- rep(0, nrow(R))
}, I = {
  R <- makeD(p)
  r <- rep(0, p - 1)
}, D = {
  R <- -makeD(p)
  r <- rep(0, p - 1)
}, VI = {
  R <- makeD(p)
  R <- rbind(R, A)
  r <- rep(0, nrow(R))
}, VD = {
  R <- -makeD(p)
  R <- rbind(R, A)
  r <- rep(0, nrow(R))

\[
R <- \text{makeD}(p)
\]
\[
R <- \text{rbind}(R, -A)
\]
\[
r <- \text{rep}(0, \text{nrow}(R))
\]
\[
R <- -\text{makeD}(p)
\]
\[
R <- \text{rbind}(R, -A)
\]
\[
r <- \text{rep}(0, \text{nrow}(R))
\]
\[
R = \text{NULL}
\]
\[
r = \text{NULL}
\]
\[
\text{list}(x = \text{list}(x = xun), F = F, \lambda = \lambda, A = A, R = R[, -1], r = r)
\]

\[
\text{qss} <- \text{function}(x, \text{constraint} = "N", \lambda = 1, \text{ndum} = 0, \text{dummies} = \text{NULL}, \\
\theta, w = \text{rep}(1, \text{length}(x)))
\]
\[
\{
\text{if (is.matrix}(x) \text{)} \{ \\
\text{if (ncol}(x) == 2) \text{ }
\text{qss} <= \text{qss2}(x, \text{constraint} = \text{constraint}, \text{dummies} = \text{dummies}, \\
\lambda = \lambda, \text{ndum} = \text{ndum}, w = w) \\
\text{else if (ncol}(x) == 1) \\
x <= \text{as.vector}(x) \\
\text{else stop("qss objects must have dimension 1 or 2")}
\}
\text{if (is.vector}(x) \text{)} \{
\text{qss} <= \text{myqss1}(x, \text{constraint} = \text{constraint}, \lambda = \lambda, \theta = \theta, \\
\text{dummies} = \text{dummies}, \text{ndum} = \text{ndum}, w = w)
\text{qss}
\}
\}

#only changed the part of rqss fitting on qss terms without restrictions
\[
\text{myrqss} <- \text{function}(\text{formula}, \tau = 0.5, \text{data} = \text{parent.frame}(), \text{weights}, \\
\text{na.action}, \text{method} = "fn", \text{contrasts} = \text{NULL}, \ldots)
\]
\[
\{
\]
call <- match.call()

m <- match.call(expand = FALSE)

temp <- c("", "formula", "data", "weights", "na.action")
m <- m[match(temp, names(m), nomatch = 0)]
m[[1]] <- as.name("model.frame")
special <- "qss"

Terms <- if (missing(data))
terms(formula, special)
else terms(formula, special, data = data)

qssterms <- attr(Terms, "specials")$qss
dropx <- NULL

if (length(qssterms)) {
tmpc <- untangle.specials(Terms, "qss")
ord <- attr(Terms, "order")[tmpc$terms]
if (any(ord > 1))
stop("qss can not be used in an interaction")
dropx <- tmpc$terms
if (length(dropx))
Terms <- Terms[-dropx]
attr(Terms, "specials") <- tmpc$vars
qssnames <- unlist(lapply(parse(text = tmpc$vars),
function(x) deparse(x[[2]])))
}

m$formula <- Terms

m <- eval(m, parent.frame())

weights <- model.extract(m, weights)

process <- (tau < 0 || tau > 1)

Y <- model.extract(m, "response")

X <- model.matrix(Terms, m, contrasts)
p <- ncol(X)

if (length(qssterms) > 0) {
F <- as.matrix.csr(X)

qss <- lapply(tmpc$vars, function(u) eval(parse(text = u),

78
data, enclos = pf))
mqss <- length(qss)
ncA <- rep(0, mqss + 1)
nrA <- rep(0, mqss + 1)
nrR <- rep(0, mqss + 1)
for (i in 1:mqss) {
  F <- cbind(F, qss[[i]]$F)
  ncA[i + 1] <- ncol(qss[[i]]$A)
  nrA[i + 1] <- nrow(qss[[i]]$A)
  nrR[i + 1] <- ifelse(is.null(nrow(qss[[i]]$R)), 0, nrow(qss[[i]]$R))
}
F = F[, -1]
A <- as.matrix.csr(0, sum(nrA), sum(ncA))
if (sum(nrR) > 0) {
  R <- as.matrix.csr(0, sum(nrR), sum(ncA))
  nrR <- cumsum(nrR)
}
ncA <- cumsum(ncA)
nrA <- cumsum(nrA)
lambdas <- rep(0, mqss)
for (i in 1:mqss) {
  lambdas[i] <- qss[[i]]$lambda
  Arows <- (1 + nrA[i]):nrA[i + 1]
  Acols <- (1 + ncA[i]):ncA[i + 1]
  A[Arows, Acols] <- qss[[i]]$lambda * qss[[i]]$A
  if (nrR[i] < nrR[i + 1])
    R[(1 + nrR[i]):nrR[i + 1], (1 + ncA[i]):ncA[i + 1]] <- qss[[i]]$R
}
if (nrR[mqss + 1] > 0) {
  R <- cbind(as.matrix.csr(0, nrR[mqss + 1], p), R)
  r <- rep(0, nrR[mqss + 1])
}
else {
  R <- NULL
  r <- NULL
}
X <- rbind(F, A)
Y <- c(Y, rep(0, nrow(A)))
rhs <- t(rbind((1 - tau) * F, 0.5 * A)) %*% rep(1, nrow(X))
XpX <- t(X) %*% X
nnzdmax <- XpX@ia[length(XpX@ia)] - 1
nsubmax <- max(nnzdmax, floor(1000 + exp(-1.6) * nnzdmax^1.2))
nnzlmax <- floor(2e+05 - 2.8 * nnzdmax + 7e-04 * nnzdmax^2)
tmpmax <- floor(1e+05 + exp(-12.1) * nnzdmax^2.35)
fit <- if (length(r) > 0)
  rqss.fit(X, Y, tau = tau, rhs = rhs, method = "sfnc",
  R = R, r = r, nsubmax = nsubmax, nnzlmax = nnzlmax,
  tmpmax = tmpmax)
else rqss.fit(X, Y, tau = tau, rhs = rhs, method = "sfn",
  nnzlmax = nnzlmax, nsubmax = nsubmax, tmpmax = tmpmax)
for (i in 1:mqss) {
  ML <- p + ncA[i]
  MU <- p + ncA[i + 1] - 1
  qss[[i]] <- list(xyz = cbind(qss[[i]]$x$x, qss[[i]]$x$y),
                  coef = fit$coef[ML:MU],
                  dummies = qss[[i]]$dummies,
                  X = as.matrix(F))
  names(qss) <- qssnames
  fit$qss <- qss
}
else {
  fit <- if (length(weights))
    rq.wfit(X, Y, tau = tau, weights, method, ...)
  else rq.fit(X, Y, tau = tau, method, ...)
}
fit$terms <- Terms
fit$formula <- formula
fit$tau <- tau
if (length(qss$terms)) {
  fit$lambdas <- lambdas
  fit$nrA <- nrA
} else fit$lambda <- fit$nrA <- NA
attr(fit, "na.message") <- attr(m, "na.message")
fit
}