Nonparametric Conditional Density Estimation Using Piecewise-Linear Solution Path of Kernel Quantile Regression

Ichiro Takeuchi
Division of Computer Science, Graduate School of Engineering, Mie University

Kaname Nomura
Ricoh Software Inc.

Takafumi Kanamori
Department of Computer Science and Mathematical Informatics,
Graduate School of Information Science, Nagoya University

October 23, 2007

Abstract

The goal of regression analysis is to describe the stochastic relationship between a vector of inputs $\boldsymbol{x}$ and a scalar output y. This can be achieved by estimating the entire conditional density $p(y|\boldsymbol{x})$. In this paper we present a new approach for nonparametric conditional density estimation. We develop a piecewise-linear path-following method for kernel-based quantile regression. It enables us to estimate the cumulative distribution function (CDF) of conditional density $p(y|\boldsymbol{x})$ in piecewise-linear form. After smoothing the estimated piecewise-linear CDF, we obtain nonparametric conditional density estimate $\hat{p}(y|\boldsymbol{x})$ for all $\boldsymbol{x}$ in the input domain. Theoretical analyses and numerical experiments are presented for showing the effectiveness of the approach.

1 Introduction

The goal of regression analysis is to describe a statistical relationship between a vector of inputs $\boldsymbol{x}$ and a scalar output y. The stochastic dependency of y on $\boldsymbol{x}$ can be fully described by modeling the conditional density $p(y|\boldsymbol{x})$. In this paper, we are concerned with the problem of estimating conditional density from a set of input-output training pairs. We denote training set as $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathcal{X}$ is a vector of inputs from domain $\mathcal{X}$ and $y_i \in \mathbb{R}$ is a scalar output.

Figure 1 shows an example of conditional density estimation for a data set on bone mineral density (BMD) in adolescent.[1] In this example, the relative change in spinal BMD is studied as a function of age, and measurements of 485 adolescents $\{(x_i, y_i)\}_{i=1}^{485}$ are displayed as a scatter plot. Using the methodology introduced in this paper we estimated the densities of the relative change in spinal BMD for each of 12, 18, and 24 years old adolescents. These estimated conditional densities are superimposed in the scatter plot (their scales are adjusted for display). Conditional density estimation is an effective tool for exploratory data analysis. It provides deeper insight into the underlying mechanism of the data than conventional regression analysis does.

In many regression analyses, strong assumptions are usually imposed for the structure of conditional densities. The regression model with the simplest structure is represented as

$$y_i = \mu(x) + \varepsilon_i, \quad E(\varepsilon_i) = 0, \quad i = 1, ..., n, \quad \{\varepsilon_i\}_{i=1}^n \text{ i.i.d.}, \tag{1}$$

where $\mu : \mathcal{X} \to \mathbb{R}$ is a deterministic function and $\{\varepsilon_i\}_{i=1}^n$ are independently and identically distributed zero mean random variables. The model (1) is sometimes called *location-shift model* because only the location of the conditional densities depends on the inputs. The location function μ in (1) can be estimated via

[1] The data set is analyzed in Hastie et al. (2001) and available from its web site.

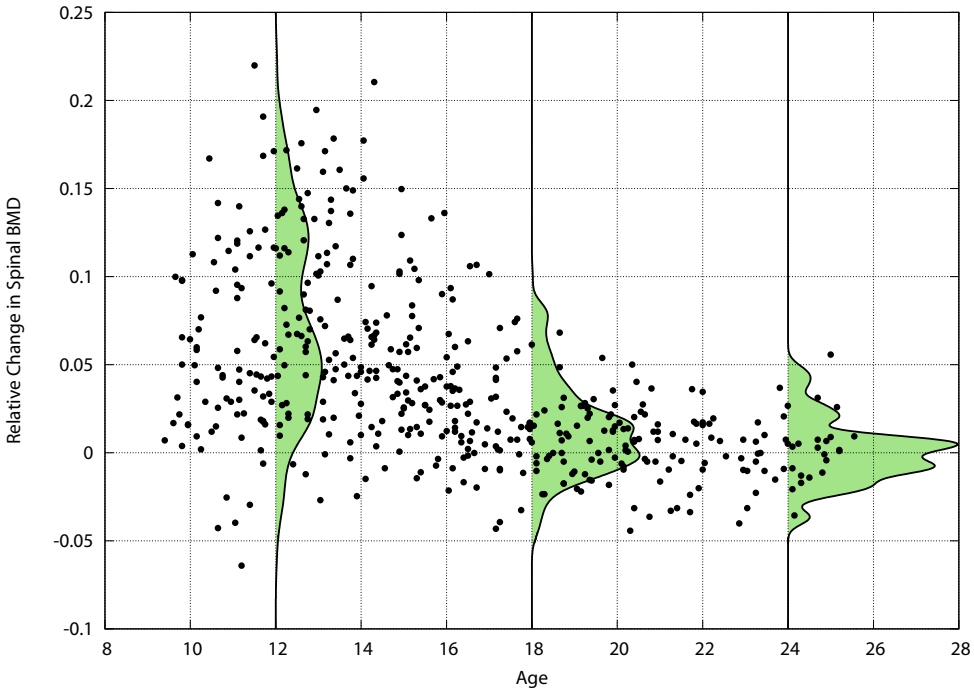


Figure 1: An example of conditional density estimation.

least-square fit etc. The density of ε can be easily estimated using the residuals from the fit. In many cases, it is often assumed that ε is normally distributed.

There is a more general model in which not only the location but also the scale of conditional density depends on the inputs. It is called *location-scale model* or *heteroscedastic model*, and represented as

$$y_i = \mu(x) + \sigma(x) \cdot \varepsilon_i, \quad E(\varepsilon_i) = 0, \quad i = 1, ..., n, \quad \{\varepsilon_i\}_{i=1}^n \text{ i.i.d.}, \tag{2}$$

where $\sigma : \mathcal{X} \to \mathbb{R}^+$ is a positive-value function. In this model we need to estimate scale function σ as well as location function μ. If parametric representation of ε is provided, maximum likelihood method can be used to estimate μ and σ. In the literature of neural networks, the model (2) is sometimes called *input-dependent-variance model*. Nix and Weigend (1994) and Williams (1996) implemented location-scale model using multi-layer perceptron (MLP) with an assumption that ε is normally distributed. Bishop (1994) proposed a more general and useful approach called *mixture density networks* (MDN). MDN describes conditional density by mixture of parametric distributions. All the involving parameters are estimated as the outputs of a MLP using maximum likelihood criterion.

Several conditional density estimators have been studied in the framework of nonparametric smoothing (Hyndman et al., 1996; Fan et al., 1996; Hall et al., 1999). In these methods, conditional densities are estimated using *weighted* version of traditional kernel density estimation. Roughly speaking, when we estimate $p(y|\boldsymbol{x}_0)$ for a certain $\boldsymbol{x}_0 \in \mathcal{X}$, the "weight" of y_i is determined based on the distance between $\boldsymbol{x}_0$ and $\boldsymbol{x}_i$. If the distance is large, small weight is assigned to y_i and vice-versa. These approaches are applicable only to one or two-dimensional input problems because of the *curse of dimensionality*.

In this paper we propose a different approach for conditional density estimation. The building block of our approach is quantile regression (Koenker and Bassett, 1978). Quantile regression is a tool to estimate the quantiles of conditional density $p(y|\boldsymbol{x})$ as functions of $\boldsymbol{x}$. For $0 < \tau < 1$, the τ-th conditional quantile function $f_\tau : \mathcal{X} \to \mathbb{R}$ is estimated by the following minimization problem

$$\hat{f}_\tau = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \psi_\tau(y_i - f(\boldsymbol{x}_i)), \quad \psi_\tau(r) = \begin{cases} (1-\tau)|r|, & \text{if } r \leq 0, \\ \tau|r|, & \text{if } 0 < r. \end{cases} \tag{3}$$

where $\mathcal{F}$ is a certain class of functions. In Takeuchi et al. (2006) and Li et al. (2007), kernel-based quantile regression (KQR) was presented. KQR inherited many desirable properties of kernel-machines. The solution is (almost) unique and guaranteed to be globally (not locally) optimal because it is formulated as convex (quadratic) optimization problem. It can be applied to high-dimensional or structured input problem with the use of kernel-trick. Figure 2 shows some conditional quantile functions estimated by KQR for BMD example.

Figure 2: An example of kernel quantile regression (KQR).

A family of conditional quantile functions f_τ at all the continuum of $\tau \in (0,1)$ provide a full description of the conditional density $p(y|\boldsymbol{x})$. Our main idea introduced in this paper is to develop a *piecewise-linear path-following* method (Hastie et al., 2004) for KQR in order to estimate the solution path of conditional quantile functions for all $\tau \in (0,1)$. This enables us to estimate the cumulative distribution function (CDF) of conditional density $p(y|\boldsymbol{x})$ in piecewise-linear form. After smoothing the CDF, we obtain nonparametric conditional density estimate $\hat{p}(y|\boldsymbol{x})$ as illustrated in Figure 1.

Our conditional density estimator has several advantages. Compared with the works by Nix and Weigend (1994) and Williams (1996), our estimator is more flexible because it does not assume any parametric form of the conditional density distributions. The normal mixture representation by MDN (Bishop, 1994) is a useful approach, but it is still in the framework of parametric density estimation. The works by Hyndman et al. (1996); Fan et al. (1996); Hall et al. (1999) are applicable only to one or two input dimensional problems, but ours can be straightforwardly applied to high-dimensional or structured input problems with the use of kernel-trick.

The paper is organized as follows. In section 2 we briefly review KQR and summarize the optimality conditions. In section 3 we describe the piecewise-linear path-following algorithm for KQR. In section 4 we discuss how to estimate conditional density from the piecewise-linear solution path. In addition, we provide some theoretical statements on our conditional density estimator. In section 5 we empirically examine the performances of our conditional density estimator. We close in section 6 with some concluding remarks and future works.

2 Kernel Quantile Regression (KQR)

In this section, we summarize the kernel-based quantile regression (KQR) studied in Takeuchi et al. (2006). They studied a nonparametric model for conditional quantile function by considering a linear model in a Reproducing Kernel Hilbert Space (RKHS). The model with quantile order $\tau \in (0,1)$ is written as

$$f_\tau(\boldsymbol{x}) = \boldsymbol{\beta}_\tau^\top \Phi(\boldsymbol{x}) + \beta_{\tau 0},$$

where Φ is the feature map implicitly defined by a kernel K, $\boldsymbol{\beta}_\tau$ is a vector of slope parameters, and $\beta_{\tau 0}$ is a scalar intercept parameter for the linear model in the RKHS. The superscript $^\top$ denotes the transpose of vectors or matrices. The quantile regression estimator (3) for f_τ with a L_2 regularizer yields

$$\min_{\boldsymbol{\beta}_\tau, \beta_{\tau 0}} \sum_{i=1}^{n} \psi_\tau(y_i - (\boldsymbol{\beta}_\tau^\top \Phi(\boldsymbol{x}_i) + \beta_{\tau 0})) + \frac{\lambda}{2} \boldsymbol{\beta}_\tau^\top \boldsymbol{\beta}_\tau, \tag{4}$$

where $\lambda \in \mathbb{R}^+$ is a positive scalar parameter to control the degree of regularization.

The problem (4) is rewritten as the following constrained minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\beta}_\tau, \beta_{\tau 0}, \{\xi_{\tau i}^+\}_{i=1}^n, \{\xi_{\tau i}^-\}_{i=1}^n} \quad & \sum_{i=1}^{n} (\tau \xi_{\tau i}^+ + (1-\tau)\xi_{\tau i}^-) + \frac{1}{2}\lambda \boldsymbol{\beta}_\tau^T \boldsymbol{\beta}_\tau \\ \text{s.t.} \quad & y_i - \boldsymbol{\beta}_\tau^T \Phi(\boldsymbol{x}_i) - \beta_{\tau 0} - \xi_{\tau i}^+ + \xi_{\tau i}^- = 0, \;\; \xi_{\tau i}^+ \geq 0, \;\; \xi_{\tau i}^- \geq 0, \;\; \forall i, \end{aligned} \tag{5}$$

where $\{\xi_{\tau i}^+\}_{i=1}^n, \{\xi_{\tau i}^-\}_{i=1}^n$ are slack variables. The Lagrange primal function of (5) is represented as

$$\begin{aligned} L_\tau &= \sum_{i=1}^n \{\tau\xi_{\tau i}^+ + (1-\tau)\xi_{\tau i}^-\} + \frac{1}{2}\lambda\boldsymbol{\beta}_\tau^T\boldsymbol{\beta} \\ &+ \sum_{i=1}^n \alpha_{\tau i}\{y_i - \boldsymbol{\beta}_\tau^T\Phi(\boldsymbol{x}_i) - \beta_{\tau 0} - \xi_{\tau i}^+ + \xi_{\tau i}^-\} - \sum_{i=1}^n (\eta_{\tau i}^+\xi_{\tau i}^+ + \eta_{\tau i}^-\xi_{\tau i}^-), \end{aligned}$$

where $\{\alpha_{\tau i} \in \mathbb{R}\}_{i=1}^n$, $\{\eta_{\tau i}^+ \in \mathbb{R}^+\}_{i=1}^n$ and, $\{\eta_{\tau i}^- \in \mathbb{R}^+\}_{i=1}^n$ are the Lagrange multipliers. Letting the derivatives of L_τ w.r.t. primal variables $\boldsymbol{\beta}_\tau$, $\beta_{\tau 0}$, $\{\xi_{\tau i}^+\}_{i=1}^n$ and $\{\xi_{\tau i}^-\}_{i=1}^n$ be zero, we have

$$\boldsymbol{\beta}_\tau = \frac{1}{\lambda}\sum_{i=1}^n \alpha_{\tau i}\Phi(\boldsymbol{x}_i), \;\; \sum_{i=1}^n \alpha_{\tau i} = 0, \;\; \eta_{\tau i}^+ = \tau - \alpha_{\tau i}, \; \eta_{\tau i}^- = (1-\tau) + \alpha_{\tau i}, \;\; \forall i.$$

In addition, optimal solution needs to satisfy

$$\xi_{\tau i}^+\eta_{\tau i}^+ = \xi_{\tau i}^+\{\tau - \alpha_{\tau i}\} = 0, \;\; \forall i, \;\; \xi_{\tau i}^-\eta_{\tau i}^- = \xi_{\tau i}^-\{(1-\tau) + \alpha_{\tau i}\} = 0, \;\; \forall i.$$

These optimality conditions are summarized as

$$\begin{aligned} y_i > f_\tau(\boldsymbol{x}_i) &\implies \alpha_{\tau i} = \tau, && \text{(6a)} \\ y_i = f_\tau(\boldsymbol{x}_i) &\implies \tau - 1 \le \alpha_{\tau i} \le \tau, && \text{(6b)} \\ y_i < f_\tau(\boldsymbol{x}_i) &\implies \alpha_{\tau i} = \tau - 1, && \text{(6c)} \\ \sum_{i=1}^n \alpha_{\tau i} &= 0. && \text{(6d)} \end{aligned}$$

If there are non-bounded $\alpha_{\tau i} \in (\tau - 1, \tau)$ at optimality, the intercept $\beta_{\tau 0}$ is determined using any one of these by

$$\beta_{\tau 0} = y_i - \frac{1}{\lambda}\sum_{j=1}^n \alpha_{\tau j}K_{ij}, \;\; \{i : \tau - 1 < \alpha_{\tau i} < \tau\},$$

otherwise, we can only specify the range of $\beta_{\tau 0}$ as

$$\max_{\{i:\alpha_{\tau i}=\tau-1\}}\{y_i - \frac{1}{\lambda}\sum_{j=1}^n \alpha_{\tau j}K_{ij}\} \le \beta_{\tau 0} \le \min_{\{i:\alpha_{\tau i}=\tau\}}\{y_i - \frac{1}{\lambda}\sum_{j=1}^n \alpha_{\tau j}K_{ij}\}. \quad (7)$$

The τ-th conditional quantile model is represented by

$$f_\tau(\boldsymbol{x}) = \frac{1}{\lambda}\{\sum_{i=1}^n \alpha_{\tau i}K(\boldsymbol{x}, \boldsymbol{x}_i) + \alpha_{\tau 0}\}, \quad (8)$$

where $\alpha_{\tau 0} = \lambda\beta_{\tau 0}$ is introduced for notational simplicity.

3 Kernel Quantile Regression Path

In recent machine learning literature, piecewise-linear path-following algorithms of several types of kernel machines have been investigated. Hastie et al. (2004) used piecewise-linear path-following algorithms to efficiently compute the solutions of support vector machine for various regularization parameters. Bach et al. (2006) applied piecewise-linear path-following algorithms to obtain ROC curves for binary classification problem. Recently, Li et al. (2007) developed an algorithm for the solution path with respect to the regularization parameter λ for kernel quantile regression using the same approach as Hastie et al. (2004).

In this paper, we use piecewise-linear path-following algorithm to obtain a continuum of the estimates of conditional quantiles in all $\tau \in (0,1)$.[2] The model of τ-th conditional quantile function (8) is characterized by $n+1$ parameters $\{\alpha_{\tau i}\}_{i=0}^n$. We investigate how these parameters change with $\tau \in (0,1)$, and show that they are described as piecewise-linear functions of τ. Let us define the following sets:

$$\begin{aligned} \mathcal{I}_\tau^+ &\stackrel{\text{def}}{=} \{i \mid y_i > f_\tau(\boldsymbol{x}_i), \alpha_{\tau i} = \tau\}, \\ \mathcal{I}_\tau^0 &\stackrel{\text{def}}{=} \{i \mid y_i = f_\tau(\boldsymbol{x}_i), \alpha_{\tau i} \in [\tau - 1, \tau]\}, \\ \mathcal{I}_\tau^- &\stackrel{\text{def}}{=} \{i \mid y_i < f_\tau(\boldsymbol{x}_i), \alpha_{\tau i} = \tau - 1\}. \end{aligned} \tag{9}$$

From the optimality conditions (6), each training data point is a member of any one of these sets. Let us define *event point* as the point in $[0,1]$ at which any one of the training points moves from one of the above three sets to another, and call such movement as *event*. Our algorithm starts with $\tau = 0$ and increases it toward 1. As τ increases, the algorithm keeps track all the events. Let the number of events as L and denote the sequence of event points as $0 = \tau_0 \le \tau_1 \le \tau_2 \le ... \le \tau_\ell \le ... \le \tau_L = 1$. It will be shown that the parameters $\{\alpha_{\tau i}\}_{i=0}^n$ are linear in τ between any two event points $[\tau_\ell, \tau_{\ell+1}]$, $l = 0, 1, ..., L-1$.

3.1 Linearity of the solution path between two event points

First, we consider the solution path of KQR between two event points τ_l and τ_{l+1}, $l = 0, 1, ..., L-1$. Here, we omit the subscript τ to denote three sets in (9) because the members of these three sets are invariant between two event points. Let us denote the size of $\mathcal{I}^0$ as n^0 and let the first n^0 elements be the members of $\mathcal{I}^0$ without loss of generality. Also, let K^0 be the $n^0 \times n^0$ matrix with (i,j)-th entry K_{ij} for i and j in $\mathcal{I}^0$, and $\mathbf{1}_{n^0}$ be n^0 vector of 1. In this subsection, we assume $\mathcal{I}^0$ is not empty (i.e. $n^0 > 0$). We deal with the case of empty $\mathcal{I}^0$ in 3.3.

In the interval $\tau_\ell \le \tau \le \tau_{\ell+1}$, for each $j \in \mathcal{I}^0$,

$$\begin{aligned} f_\tau(\boldsymbol{x}_j) - f_{\tau_l}(\boldsymbol{x}_j) &= \frac{1}{\lambda}\{\sum_{i=1}^n \alpha_{\tau i} K_{ij} + \alpha_{\tau 0}\} - \frac{1}{\lambda}\{\sum_{i=1}^n \alpha_{\tau_l i} K_{ij} + \alpha_{\tau_l 0}\} \\ &= \frac{1}{\lambda}\{\sum_{i=1}^n (\alpha_{\tau i} - \alpha_{\tau_l i}) K_{ij} + (\alpha_{\tau 0} - \alpha_{\tau_l 0})\} \\ &= \frac{1}{\lambda}\{\sum_{i \in \mathcal{I}^0} (\alpha_{\tau i} - \alpha_{\tau_l i}) K_{ij} + \sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} (\alpha_{\tau i} - \alpha_{\tau_l i}) K_{ij} + (\alpha_{\tau 0} - \alpha_{\tau_l 0})\} \\ &= \frac{1}{\lambda}\{\sum_{i \in \mathcal{I}^0} (\alpha_{\tau i} - \alpha_{\tau_l i}) K_{ij} + (\tau - \tau_l) \sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} K_{ij} + (\alpha_{\tau 0} - \alpha_{\tau_l 0})\} \\ &= 0, \end{aligned}$$

where the fourth equality holds because $\alpha_{\tau i} = \tau, \alpha_{\tau_\ell i} = \tau_\ell, \forall i \in \mathcal{I}^+$ and $\alpha_{\tau i} = \tau - 1, \alpha_{\tau_\ell i} = \tau_\ell - 1, \forall i \in \mathcal{I}^-$ from (6a) and (6c), respectively. The last equality holds simply because $f_\tau(\boldsymbol{x}_j) = f_{\tau_l}(\boldsymbol{x}_j) = y_j, \forall j \in \mathcal{I}^0$ from (6b). Also from (6d)

$$\begin{aligned} \sum_{i=1}^n \alpha_{\tau i} - \sum_{i=1}^n \alpha_{\tau_l i} &= \sum_{i \in \mathcal{I}^0} (\alpha_{\tau i} - \alpha_{\tau_l i}) + \sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} (\alpha_{\tau i} - \alpha_{\tau_l i}) \\ &= \sum_{i \in \mathcal{I}^0} (\alpha_{\tau i} - \alpha_{\tau_l i}) + \sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} (\tau - \tau_l) \\ &= \sum_{i \in \mathcal{I}^0} (\alpha_{\tau i} - \alpha_{\tau_l i}) + (n - n^0)(\tau - \tau_l) = 0. \end{aligned}$$

[2] A preliminary version of the piecewise-linear path-following algorithm described in 3.1 and 3.2 was presented at International Joint Conference on Neural Network in July, 2006.

We can combine these n^0+1 equations as follows:

$$\begin{bmatrix} 0 & \mathbf{1}_{n^0}^\top \\ \mathbf{1}_{n^0} & K^0 \end{bmatrix} \begin{bmatrix} \alpha_{\tau 0} - \alpha_{\tau_l 0} \\ \alpha_{\tau 1} - \alpha_{\tau_l 1} \\ \alpha_{\tau 2} - \alpha_{\tau_l 2} \\ \vdots \\ \alpha_{\tau n^0} - \alpha_{\tau_l n^0} \end{bmatrix} = \begin{bmatrix} n^0 - n \\ -\sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} K_{i1} \\ -\sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} K_{i2} \\ \vdots \\ -\sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} K_{in^0} \end{bmatrix} (\tau - \tau_l). \tag{10}$$

Solving the linear system of equations, we can write

$$\alpha_{\tau j} - \alpha_{\tau_l j} = c_{\ell j}(\tau - \tau_l),\ j = 0,\ j \in \mathcal{I}^0,$$

where

$$\begin{bmatrix} c_{\ell 0} \\ c_{\ell 1} \\ c_{\ell 2} \\ \vdots \\ c_{\ell n^0} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{1}_{n^0}^\top \\ \mathbf{1}_{n^0} & K^0 \end{bmatrix}^{-1} \begin{bmatrix} n^0 - n \\ -\sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} K_{i1} \\ -\sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} K_{i2} \\ \vdots \\ -\sum_{i \in \mathcal{I}^+ \cup \mathcal{I}^-} K_{in^0} \end{bmatrix}.$$

Also for $j \in \mathcal{I}^+ \cup \mathcal{I}^-$,

$$\alpha_{\tau j} - \alpha_{\tau_l j} = \tau - \tau_l,\ j \in \mathcal{I}^+ \cup \mathcal{I}^-,$$

and we define $c_{\ell j} = 1$ for all $j \in \mathcal{I}^+$ and $\mathcal{I}^-$. From these results, we can conclude that the KQR parameters $\{\alpha_{\tau i}\}_{i=0}^n$ are linear in τ between any two event points.

3.2 How to find events

The next event point τ_{l+1} is found when one of the followings happens:

`OUT+` : One of the training points moves from $\mathcal{I}_\tau^0$ to $\mathcal{I}_\tau^+$;

`OUT-` : One of the training points moves from $\mathcal{I}_\tau^0$ to $\mathcal{I}_\tau^-$;

`IN` : One of the training points moves from $\mathcal{I}_\tau^+$ or $\mathcal{I}_\tau^-$ to $\mathcal{I}_\tau^0$.

From the optimality conditions (6a) and (6c), `OUT+` event and `OUT-` event happen when

$$\begin{aligned} \alpha_{\tau i} &= \alpha_{\tau_\ell i} + c_{\ell i}(\tau - \tau_\ell) = \tau, \text{ for some } i \in \mathcal{I}_\tau^0 \\ \alpha_{\tau i} &= \alpha_{\tau_\ell i} + c_{\ell i}(\tau - \tau_\ell) = \tau - 1, \text{ for some } i \in \mathcal{I}_\tau^0, \end{aligned}$$

respectively. From the optimality condition (6b), `IN` event is found when

$$\begin{aligned} f_{\tau_\ell}(\boldsymbol{x}_i) &= \frac{1}{\lambda}\{\sum_{j=1}^n \alpha_{\tau j} K_{ij} + \alpha_{\tau 0}\} \\ &= \frac{1}{\lambda}\{\sum_{j=1}^n [\alpha_{\tau_\ell j} + c_{\ell j}(\tau - \tau_\ell)] K_{ij} + [\alpha_{\tau_{\ell 0}} + c_{\ell 0}(\tau - \tau_\ell)]\} = y_i, \end{aligned}$$

for some $i \in \mathcal{I}_\tau^+ \cup \mathcal{I}_\tau^-$.

Among these, the smallest $\tau \geq \tau_\ell$ is chosen as the next event point, i.e.

$$\tau_{\ell+1} = \min_{i \in \mathcal{I}_{\tau_\ell}^0,\ j \in \mathcal{I}_{\tau_\ell}^+ \cup \mathcal{I}_{\tau_\ell}^-}^{\geq \tau_\ell} \left\{ \frac{\alpha_{\tau_\ell i} - c_{\ell i}\tau_\ell}{1 - c_{\ell i}},\ \frac{\alpha_{\tau_\ell i} - c_{\ell i}\tau_\ell + 1}{1 - c_{\ell i}},\ \tau_\ell + \frac{\lambda\{y_j - f_{\tau_\ell}(\boldsymbol{x}_j)\}}{\sum_{k=1}^n c_{\ell k} K_{jk} + c_{\ell 0}} \right\}, \tag{11}$$

where we used $\min_i^{\geq a}\{z_i\}$ as a simplified notation of $\min_i\{z_i | z_i \geq a\}$.

3.3 Empty $\mathcal{I}_\tau^0$ and the boundary conditions

We need to develop a way to deal with the empty $\mathcal{I}_\tau^0$ situations, and establish the initial and terminal conditions of the KQR solution path. The initial ($\tau = 0$) and the terminal ($\tau = 1$) states of the KQR solution path can be considered as special cases of the empty $\mathcal{I}_\tau^0$ situations. Thus, we first examine the situations where $\mathcal{I}_\tau^0$ is empty, and use the obtained result to describe the initial and the terminal conditions.

The following lemma shows $\mathcal{I}_\tau^0$ can be empty at most $n+1$ isolated points in $\tau \in [0,1]$.

Lemma 1. *$\mathcal{I}_\tau^0$ is empty only if $\tau \in \{0, \frac{1}{n}, \frac{2}{n}, ..., \frac{n-1}{n}, 1\}$.*

The proof is given in A.1. When $\mathcal{I}_\tau^0$ is empty, the derivation of the piecewise-linear solution path in the previous subsection fails. Suppose that $\mathcal{I}_{\tau^*}^0$ is empty at $\tau^* \in \{0, \frac{1}{n}, \frac{2}{n}, ..., \frac{n-1}{n}, 1\}$. Then, from (7),

$$\max_{i \in \mathcal{I}_{\tau^*}^-} (\lambda y_i - \sum_{j=1}^n \alpha_{\tau^* j} K_{ij}) \le \alpha_{\tau^* 0} \le \min_{i \in \mathcal{I}_{\tau^*}^+} (\lambda y_i - \sum_{j=1}^n \alpha_{\tau^* j} K_{ij}).$$

From Lemma 1, for $0 < \varepsilon < \frac{1}{n}$, $\mathcal{I}_{\tau^*-\varepsilon}^0$ and $\mathcal{I}_{\tau^*+\varepsilon}^0$ are not empty. It indicates that the following two events occur at $\tau = \tau^*$. In the first event one last element of $\mathcal{I}_{\tau^*}^0$ moves out and $\mathcal{I}_{\tau^*}^0$ becomes empty. In the second event one new element moves in $\mathcal{I}_{\tau^*}^0$. Denote l^* be the index of the first event. The next lemma specifies which element moves out from $\mathcal{I}_{\tau^*}^0$ at l^* and which element moves in $\mathcal{I}_{\tau^*}^0$ at l^*+1-th event.

Lemma 2. *When $\mathcal{I}_{\tau^*}^0$ is empty, if the sample size $n > 1$,*

$$i_{max}^- = \arg\max_{i \in \mathcal{I}_{\tau^*}^-} \{\lambda y_i - \sum_{j=1}^n \alpha_{\tau^* j} K_{ij}\}, \tag{12}$$

moves out from $\mathcal{I}_{\tau^*}^0$ at l^*-th event and

$$i_{min}^+ = \arg\min_{i \in \mathcal{I}_{\tau^*}^+} \{\lambda y_i - \sum_{j=1}^n \alpha_{\tau^* j} K_{ij}\}. \tag{13}$$

moves in $\mathcal{I}_{\tau^*}^0$ at l^*+1-th event, where l^* denotes the index of the event that $\mathcal{I}_{\tau^*}^0$ becomes empty.

The proof is given in A.2.From Lemma 2, we can develop the algorithm for empty $\mathcal{I}_\tau^0$ cases. If it happens that $\mathcal{I}_\tau^0$ becomes empty at l^*-th event we set the next event point

$$\begin{aligned}
&\tau_{l^*+1} = \tau_{l^*}, \\
&\alpha_{\tau_{l^*+1} i} = \alpha_{\tau_{l^*} i} \text{ for all } i \in \{1, ..., n\}, \\
&\alpha_{\tau_{l^*+1} 0} = \min_{i \in \mathcal{I}_{\tau_{l^*}}^+} \{\lambda y_i - \sum_{j=1}^n \alpha_{\tau_{l^*} j} K_{ij}\}.
\end{aligned}$$

Interestingly, the non-uniqueness of the solution at τ with empty $\mathcal{I}_\tau^0$ corresponds to the non-uniqueness of the ordinary sample quantiles in unconditional settings.[3] If all of our training inputs $\{\boldsymbol{x}_i\}_{i=1}^n$ take same values, our conditional quantile estimator reduces to ordinary *un*conditional sample quantiles. In this case, the empty $\mathcal{I}_\tau^0$ situation examined in this subsection occurs exactly $n-1$ times in $\tau \in (0,1)$ at $\tau = \frac{1}{n}, \frac{2}{n}, ..., \frac{n-1}{n}$.

Using this result, we can describe the initial and the terminal stages of the algorithm. It is easy to note that, when $\tau = 0$,

$$\alpha_{\tau 1} = ... = \alpha_{\tau n} = 0, \ \alpha_{\tau 0} < \min\{y_1, ..., y_n\}$$

[3]For example, consider a univariate sample $Y_1, Y_2, ..., Y_n$ and denote the r-th smallest, $1 \le r \le n$, of the Y's as $Y_{(r)}$. Then, for n odd, the sample median is uniquely defined as $Y_{(n+1)/2}$, but for n even, it is not uniquely defined, i.e., any number between $Y_{(n/2)}$ and $Y_{(n/2+1)}$ is the sample median.

satisfy the optimality conditions (6a) $\sim$ (6d). In this case, all the training data points $\{1, ..., n\}$ are in $\mathcal{I}_\tau^+$, the special case of empty $\mathcal{I}_{\tau_0}^0$. Using the result of the second part of Lemma 2, we can initialize the algorithm at $\tau = 0$ with

$$\alpha_{\tau 1} = ... = \alpha_{\tau n} = 0,\ \alpha_{\tau 0} = \min_{i \in \mathcal{I}_\tau^+} \{\lambda y_i - \sum_{j=1}^{n} \alpha_{\tau j} K_{ij}\} = \lambda \min_{i=1,...,n} \{y_i\},$$

where $\mathcal{I}_\tau^0$ contains only $\arg\min_i \{y_i\}$ and $\mathcal{I}_\tau^+$ contains the rest of the training data points. Similarly, the algorithm is terminated at $\tau = 1$ with

$$\alpha_{\tau 1} = ... = \alpha_{\tau n} = 1,\ \alpha_{\tau 0} = \max_{i \in \mathcal{I}_\tau^-} \{\lambda y_i - \sum_{j=1}^{n} \alpha_{\tau j} K_{ij}\} = \lambda \max_{i=1,...,n} \{y_i\},$$

where $\mathcal{I}_\tau^0$ contains only $\arg\max_i \{y_i\}$ and $\mathcal{I}_\tau^-$ contains the rest of the training data points.

3.4 KQR-path algorithm

The piecewise-linear path-following algorithm for KQR (KQR-path algorithm) is summarized as follows:

Input. Provide training data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, kernel function K, and regularization parameter λ.

1. Set $l = 0$, $\tau_l = 0$, $\alpha_{\tau_\ell i} = 0$ for $i = 1, ..., n$, and $\alpha_{\tau_\ell 0} = \lambda \min_{i=1,...,n} \{y_i\}$. Let $\arg\min_{i=1,...,n} \{y_i\}$ be the member of $\mathcal{I}_{\tau_\ell}^0$ and the rest of the training data points be the member of $\mathcal{I}_{\tau_\ell}^+$. Go to **2**.

2. If $\mathcal{I}_{\tau_\ell}^0$ is empty, go to **3**. Otherwise, go to **4**.

3. Set $\tau_{\ell+1} = \tau_\ell$, $\alpha_{\tau_{\ell+1}\, i} = \alpha_{\tau_\ell i}$ for $i = 1, ..., n$, and $\alpha_{\tau_{\ell+1}\, 0} = \min_{i \in \mathcal{I}_{\tau_\ell}^+} \{\lambda y_i - \sum_{j=1}^n \alpha_{\tau_\ell j} K_{ij}\}$. Go to **5**.

4. Solve the linear system of equations (10) to obtain the piecewise-linear functions of $\{\alpha_{\tau i}\}_{i=0}^n$ for $\tau_\ell < \tau < \tau_{\ell+1}$. Find the next event point $\tau_{\ell+1}$ using (11). Set $\alpha_{\tau_{\ell+1} i}, i = 0, 1, ..., n$ using the obtained piecewise-linear functions. Go to **5**.

5. Update the three sets in (9). Set $\ell = \ell + 1$. If $\mathcal{I}_{\tau_\ell}^0$ contains only $\arg\max_{i=1,...,n} \{y_i\}$ and $\mathcal{I}_{\tau_\ell}^-$ contains the rest of training data points, go to **6**. Otherwise, go back to **2**.

6. Set $L = \ell + 1$, $\tau_L = 1$, $\alpha_{\tau_L i} = 1$ for $i = 1, ..., n$, $\alpha_{\tau_L 0} = \lambda \max_{i=1,...,n} \{y_i\}$, and terminate the algorithm.

Output. Obtain a sequences of event points: $\tau_0, \tau_1, ..., \tau_L$, and a sequences of parameter vectors: $\boldsymbol{\alpha}_{\tau_0}, \boldsymbol{\alpha}_{\tau_1}, ..., \boldsymbol{\alpha}_{\tau_L}$, where $\boldsymbol{\alpha}_{\tau_\ell} = \begin{bmatrix} \alpha_{\tau_\ell 0} & \alpha_{\tau_\ell 1} & \dots & \alpha_{\tau_\ell n} \end{bmatrix}^\top$.

The discussion on the computational complexity of the regularization-path algorithm in Hastie et al. (2004) similarly applies to the KQR-path algorithm. At each event, the main computational cost is solving the linear system of equations (10). With the use of rank-one update of the factorizations of the left-hand-side matrix in (10), it involves $\mathcal{O}(|\mathcal{I}_{\tau_\ell}^0|^2)$ complexities. For finding the next event in (11), we need to compute $\sum_{j=1}^n \alpha_j K_{ij}$ for all $i \in \mathcal{I}_{\tau_\ell}^+ \cup \mathcal{I}_{\tau_\ell}^-$. Using the fact that $\alpha_j = \tau_\ell$ for $j \in \mathcal{I}_{\tau_\ell}^+$ and $\alpha_j = \tau_\ell - 1$ for $j \in \mathcal{I}_{\tau_\ell}^-$, we can efficiently update $\sum_{j \in \mathcal{I}_{\tau_\ell}^+} \alpha_j K_{ij}$ and $\sum_{j \in \mathcal{I}_{\tau_\ell}^-} \alpha_j K_{ij}$. Thus, we actually need to compute $\sum_{j \in \mathcal{I}_{\tau_\ell}^0} \alpha_j K_{ij}$ for all $i \in \mathcal{I}_{\tau_\ell}^+ \cup \mathcal{I}_{\tau_\ell}^-$, and it requires $\mathcal{O}(|\mathcal{I}_{\tau_\ell}^0|(|\mathcal{I}_{\tau_\ell}^+| + |\mathcal{I}_{\tau_\ell}^-|))$ computations. If we denote the maximum size of $\mathcal{I}_{\tau_\ell}^0, \ell = 0, 1, ..., L$ as m, then the entire algorithm has $\mathcal{O}(m^2 L + mnL)$ complexity. As conjectured in other path-following studies of kernel machines (Hastie et al., 2004; Bach et al., 2006), our empirical results also indicated that the number of events L is $\mathcal{O}(n)$. With this conjecture, the computational complexity of the KQR-path algorithm is $\mathcal{O}(m^2 n + mn^2)$, which is similar to that of solving KQR for a single fixed τ.

At any `IN` event of the KQR-path algorithm, if the left-hand-side matrix of (10) becomes singular, the solution is not unique. In this case, we need to specify the subspace of optimal solutions and develop a way to continue the path-following. This can occur if and only if the kernel matrix K is not strictly

positive definite in the subspace $\{\boldsymbol{z} \in \mathbb{R}^n \mid \mathbf{1}_n^\top \boldsymbol{z} = 0,\ z_j = 0,\ \forall j \in \mathcal{I}_{\tau_\ell}^+ \cup \mathcal{I}_{\tau_\ell}^-\}$. In the case of Gaussian kernel, this degeneracy can happen if and only if there are two or more identical training examples (tied both in $\boldsymbol{x}$ and y). Such identical training examples can be detected in pre-processing stage, and we can easily cope with the situation. For example, if training examples i_1 and i_2 are identical, we can impose a constraint that the corresponding parameters α_{i_1} and α_{i_2} must take same values. This additional constraint can enable us to avoid the degeneracy situation. In the case of other kernels, degeneracies can occur (although rarely in practice) even if there are no identical training examples. Our current implementation can cope with degeneracies caused by identical training examples. But it can fail in the case of other type of degeneracies.[4] We note that this degeneracy problem on the path-following or related active set algorithms for kernel machines still remains to be an open problem in the literature. At the best of our knowledge, any existing studies, including Cauwenberghs and Poggio (2001); Hastie et al. (2004); Bach et al. (2006); Laskov et al. (2006); Li et al. (2007), do not provide satisfactory answer to this problem.

4 KQR-path and conditional probability density

The entire solution path of kernel quantile regression (KQR) in $\tau \in [0, 1]$ provides nonparametric descriptions of conditional probability densities $p(y|\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathcal{X}$.

4.1 Conditional density estimation from KQR-path

The following theorem states that KQR-path algorithm describes the cumulative distribution function (CDF) of conditional density $p(y|\boldsymbol{x})$ in piecewise-linear form.

Theorem 1. *Given an arbitrary input $\boldsymbol{x}_0 \in \mathcal{X}$, the CDF of the distribution $p(y|\boldsymbol{x} = \boldsymbol{x}_0)$ is represented as*

$$\hat{F}_{y|\boldsymbol{x}=\boldsymbol{x}_0}(\tau) = \frac{\tau_{\ell+1} - \tau}{\tau_{\ell+1} - \tau_\ell} \hat{F}_{y|\boldsymbol{x}=\boldsymbol{x}_0}(\tau_\ell) + \frac{\tau - \tau_\ell}{\tau_{\ell+1} - \tau_\ell} \hat{F}_{y|\boldsymbol{x}=\boldsymbol{x}_0}(\tau_{\ell+1}),\ \textit{for } \tau_\ell < \tau < \tau_{\ell+1},$$

where $\hat{F}_{y|\boldsymbol{x}=\boldsymbol{x}_0}(\cdot)$ is the estimated CDF of $p(y|\boldsymbol{x} = \boldsymbol{x}_0)$ by KQR-path.

The proof is given in A.3. Figure 3 shows examples of estimated piecewise-linear CDFs of $p(y|x = 12)$, $p(y|x = 18)$, and $p(y|x = 24)$ in BMD example. Unfortunately, the estimated CDF is not necessarily monotonically increasing function. This undesirable property corresponds to notorious *quantile crossing* problem. Quantile crossing refers to the situations that two or more estimated conditional quantile functions can possibly cross or overlap. This can occur because conditional quantile functions at different τs are individually estimated in quantile regression. See He (1997) and Takeuchi et al. (2006) for details on quantile crossing problem.

To obtain smooth conditional density estimates, we need to correct quantile crossing and smooth the piecewise-linear CDF. There are many possible approaches for these purposes. For example, we can apply constrained smoothing to estimated piecewise-linear quantile functions or CDFs. There are many smoothing techniques that guarantees the resulting smoothed functions to be monotonically increasing (for example, Ramsay, 1998; Mammen et al., 2001; Dette et al., 2006). In this paper, we use another simple approach. First, n samples are uniformly sampled from piecewise-linear CDFs. Then unconditional density estimation technique (see, for example, Silverman, 1986; Wand and Jones, 1994) is applied to those n samples to obtain smoothed conditional density estimate. For the experiments described in the next section, we used kernel density estimation with *Silverman's heuristics* (Silverman, 1986). Any other unconditional density estimator, such as plug-in approaches (Wand and Jones, 1994) can also be used. The conditional density plots in Figure 1 were obtained by applying the above mentioned approach to piecewise-linear CDFs in Figure 3.

[4] In all the experiments performed, we used only Gaussian kernel. Thus, we never encountered degenerate cases in this paper.

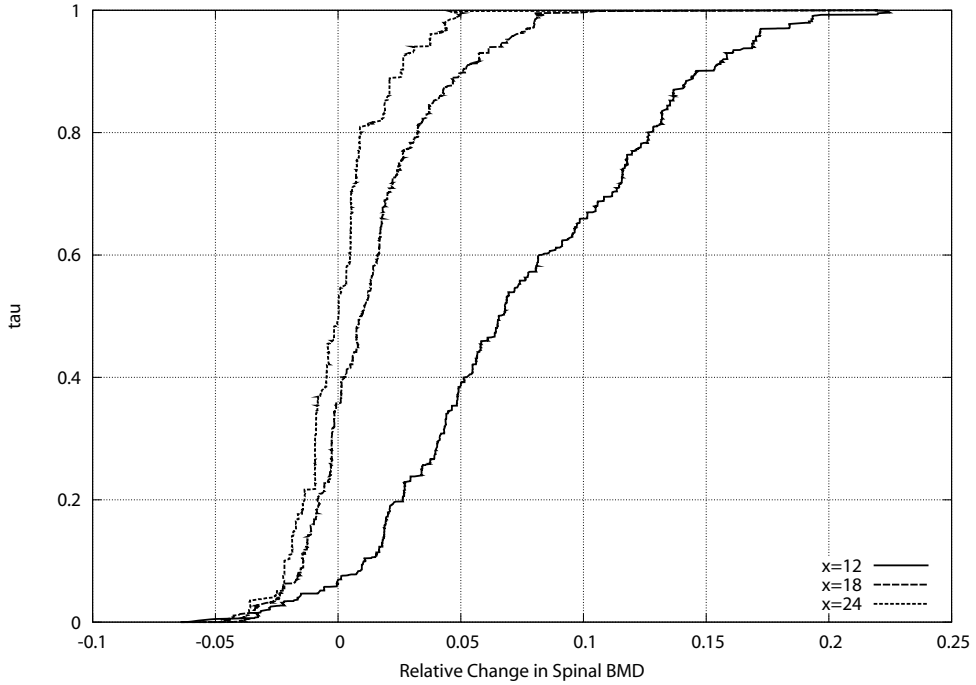


Figure 3: Estimated piecewise-linear cumulative distribution function in BMD example

4.2 Uniform bound of KQR-path estimate

We prove uniform convergence property of KQR-path estimator. Based on ψ_τ we define the expected quantile risk as

$$R_\tau[f] := E[\psi_\tau(Y - f(X))],$$

where $f \in \mathcal{F}$ is represented as $f = g + \beta$ with $g \in \mathcal{H}$ and $\beta \in \mathbb{R}$, where $\mathcal{H}$ is the reproducing kernel Hilbert space with the kernel function K. In estimation of KQR-path, we resort to minimize the empirical risk plus a regularizer. That is, the estimator is a minimizer of the empirical average of the loss function,

$$\phi(z; f, \tau) = \psi_\tau(y - f(x)) + \frac{\lambda_n}{2}\|f\|^2_{\mathcal{H}},$$

where $z = (x, y) \in \mathcal{X} \times \mathbb{R}$ and $\|\cdot\|_{\mathcal{H}}$ is the norm on $\mathcal{H}$. The regularization parameter λ_n depending on sample size n goes to zero monotonically as n goes to infinity. Let $R_{\tau,\mathrm{reg}}[f] = E[\phi(Z; f, \tau)]$, and $\widehat{R}_{\tau,\mathrm{reg}}[f]$ be the empirical counterpart. We define $\mathcal{G}$ as a class of functions mapping from $\mathcal{X} \times \mathbb{R}$ to $\mathbb{R}$ satisfying

$$\mathcal{G} \ \subset \ \{\phi(\cdot\,; f, \tau) : \mathcal{X} \times \mathbb{R} \to \mathbb{R} \mid f = g + \beta,\ g \in \mathcal{H},\ \beta \in \mathbb{R},\ \tau \in [0, 1]\}\,.$$

We suppose that $\phi(\cdot\,; f, \tau) \in \mathcal{G}$ for any $\tau \in [0, 1]$ if there exists $\tau' \in [0, 1]$ such as $\phi(\cdot\,; f, \tau') \in \mathcal{G}$. For given τ, let f_τ be a minimizer of $R_\tau[f]$ subject to $\phi(\cdot\,; f, \tau) \in \mathcal{G}$, and $\hat{f}_\tau$ be the estimator minimizing $\widehat{R}_{\tau,\mathrm{reg}}[f]$ subject to the same constraint. We evaluate the probability of the event,

$$\sup_{\tau \in [0,1]} R_\tau[\hat{f}_\tau] - R_\tau[f_\tau] > \varepsilon,$$

in the infinite sample limit.

We denote the covering number of $\mathcal{G}$ as $\mathcal{N}_\infty(\varepsilon, \mathcal{G}, n)$. Roughly speaking, $\mathcal{N}_\infty(\varepsilon, \mathcal{G}, n)$ is the minimal number to cover the set $\mathcal{G}$ by spheres with radius ε, when the distance between two functions are measured only on n sample points. The subscript of $\mathcal{N}_\infty$ denotes that ∞-norm is applied. See Zhang's papers (Zhang, 2004, 2002) for the detailed definition of the covering number and its upper bound. The rate of uniform convergence of the function class $\mathcal{G}$ is given by the following theorem.

Theorem 2 (Pollard (1984)). *Suppose all functions in $\mathcal{G}$ are uniformly bounded such as $|\phi| < L$. For any $\varepsilon > 0$ and any n such as $n > 8L^2/\varepsilon^2$, the inequality*

$$P\left\{\sup_{\phi\in\mathcal{G}}\left|\frac{1}{n}\sum_{i=1}^{n}\phi(Z_i) - E[\phi(Z)]\right| > \varepsilon\right\} \le 8\mathcal{N}_\infty(\varepsilon,\mathcal{G},n)\exp\left\{-\frac{n\varepsilon^2}{128L^2}\right\}$$

holds.

We apply Pollard's uniform convergence theorem to investigate the statistical property of KQR-path estimate.

Theorem 3. *Suppose that $\sup_{x\in\mathcal{X}} K(x,x) < \infty$, and that for any function $f = g + \beta \in \mathcal{F}$ for $g \in \mathcal{H}$ and $\beta \in \mathbb{R}$ satisfying $\phi(\cdot; f, \tau) \in \mathcal{G}$, the inequalities $\|g\|_{\mathcal{H}} \le \bar{A} := \max\{A,\, A/\sup_{x\in\mathcal{X}} K(x,x)\}$ and $|\beta| \le B$ hold. The random variable Y is assumed to be bounded such as $|Y| < M$ in probability one. Let L be a constant such that $|\phi| < L$ holds for any $\phi \in \mathcal{G}$. Note that L is represented by $\bar{A}, B, M$, and λ_1. Then, for any $\varepsilon > 0$ and any n such as $\max\left\{\lambda_n\bar{A}^2,\ \frac{2^{7/2}L}{\sqrt{n}}\right\} < \varepsilon$, we have*

$$P\Big\{\sup_{\tau\in[0,1]}(R_\tau[\hat{f}_\tau] - R_\tau[f_\tau]) > \varepsilon\Big\} \ \le\ 8\,\mathcal{N}_\infty(\varepsilon/2^5,\mathcal{G},n)\exp\left\{-\frac{n\varepsilon^2}{2^{11}L^2}\right\}. \tag{14}$$

The proof is given in A.4. Zhang (Zhang, 2004, 2002) has given an upper bound of the covering number $\mathcal{N}_\infty(\varepsilon,\mathcal{G},n)$. Applying Zhang's results, we find that the probability (14) goes to zero as n goes to infinity.

5 Numerical Experiments

To investigate and illustrate the conditional density estimation using kernel quantile regression path (KQR-path) method, we performed numerical experiments.

Various performance measures have been studied for density estimation. Most common one in the literature of (unconditional) density estimation is *mean integrated square error*. For a certain input $\boldsymbol{x}_0 \in \mathcal{X}$, if g^* and $\hat{g}$ are true and estimated conditional probability for given $\boldsymbol{x}_0$, respectively, mean integrated square error is defined as

$$\mathrm{MISE}_{y|\boldsymbol{x}_0}(\hat{g}|\boldsymbol{x}_0) = E_{\mathrm{data}}\{\int(\hat{g}(y|\boldsymbol{x}_0) - g^*(y|\boldsymbol{x}_0))^2 dy\},$$

where $E_{\mathrm{data}}\{\cdots\}$ denotes the expectation under the sample distribution.

In our KQR-path algorithm we minimize $\int_0^1\sum_{i=1}^n\psi_\tau\{y_i - f_\tau(\boldsymbol{x}_i)\}d\tau$. Let $\hat{G}(y|\boldsymbol{x}_0) = \int_{-\infty}^{y}\hat{g}(z|\boldsymbol{x}_0)dz$ be the cumulative distribution function (CDF) of $\hat{g}$. Since it has been shown in quantile regression literature that minimizing quantile loss function ψ is a promising approach for conditional quantile estimation,

$$E_{\mathrm{data}}\{\int_0^1 E_{Y|\boldsymbol{x}_0}\psi_\tau(Y - \hat{G}^{-1}(\tau|\boldsymbol{x}_0))d\tau\}, \tag{15}$$

may be one way to measure conditional density estimation performance.

We compare our KQR-path with mixture density network (MDN) (Bishop, 1994). In MDN a set of parameters for normal mixture densities are estimated by minimizing the negative log likelihood $-\sum_{i=1}^n\log\hat{g}(y_i|\boldsymbol{x}_i)$. It suggests us

$$E_{\mathrm{data}}\{-\int\log\hat{g}(y|\boldsymbol{x}_0)dy\} \tag{16}$$

may be another possible measure for conditional density estimation.

Unfortunately, neither (15) nor (16) is appropriate measure for nonparametric density estimation. The former (15) may be a sensible measure for conditional CDFs or quantile functions (QFs), but not for densities because densities can have totally different shapes even for similar CDFs or QFs. Negative log likelihood in (16) is known to be non-robust measure for nonparametric density estimation (e.g. Schuster

and Gregory, 1981; Silverman, 1986; Wand and Jones, 1994). For example, if it happens that $\hat{g}(y|\boldsymbol{x}_0) = 0$ for any y, (16) will be $+\infty$. Another notorious example of using likelihood for density estimation is actually found in mixture density estimation. In mixture density estimation with more than two components, (16) will attain minimum $(-\infty)$ if one of the components represents a spike density of infinite height at any one data point.

In 5.1 we provide quantitative evidences on the performance of our KQR-path conditional density estimator by comparing the MISEs with those of MDN for artificially generated data sets. In 5.2 we apply our conditional density estimator to real-world data sets and present qualitative discussions. For artificial data experiments we use MISE to quantify the conditional density estimation performances. For real data experiments we cannot evaluate MISE because MISE has unknown true density g^* in its definition. Note that there are no general way to estimate MISE from data. We cannot even use resampling techniques such as cross-validation because we need to know the density of y (not the y itself) in validation set in order to evaluate finite-sample analogue of MISE for validation set[5]. For the model selection (selection of hyper-parameters) of KQR-path and MDN we used cross-validation using finite-sample analogues of (15) and (16) as loss measure, respectively.

5.1 Artificial data

We generated some artificial data sets and compare the MISEs of KQR-path with those of MDN.

5.1.1 Data Sets

We considered only univariate input problems in order to visualize estimated conditional densities. Data set $\{(x_i, y_i)\}_{i=1}^n$ were generated as follows. The input $\{x_i\}_{i=1}^n$ were uniformly drawn from $[-1, 1]$. The output $\{y_i\}_{i=1}^n$ were computed as

$$y_i = \mu(x_i) + \sigma(x_i) \cdot \epsilon_i, \quad i = 1, ..., n, \quad \{\epsilon_i\}_{i=1}^n \text{ i.i.d..}$$

We considered two different sample sizes: $n =$ `100` or `200`. The location function μ was set to be $\mu(x) = \mathrm{sinc}(2\pi x)$ for all through the experiments in 5.1, where $\mathrm{sinc}(z) = (\sin z)/z$ if $z \neq 0$ and 1 otherwise. We considered homoscedastic (`homo`) and heteroscedastic (`hetero`) scale functions for σ. We used $\sigma(x) = 0.25$ for the former and $\sigma(x) = 0.125 \exp(1 - x)$ for the latter. To simulate a variety of densities, we used 8 normal mixture densities introduced in Marron and Wand (1992)[6]:

`D1 Gaussian`: $N(0, 1)$,

`D2 Skewed unimodal`: $\frac{1}{5}N(0, 1) + \frac{1}{5}N(\frac{1}{2}, (\frac{2}{3})^2) + \frac{3}{5}N(\frac{13}{15}, (\frac{5}{9})^2)$,

`D3 Strongly skewed`: $\sum_{k=0}^{7} \frac{1}{8}N(3\{(\frac{2}{3})^k - 1\}, (\frac{2}{3})^{2k})$,

`D4 Kurtotic unimodal`: $\frac{2}{3}N(0, 1) + \frac{1}{3}N(0, (\frac{1}{10})^2)$,

`D5 Outlier`: $\frac{1}{10}N(0, 1) + \frac{9}{10}N(0, (\frac{1}{10})^2)$,

`D6 Bimodal`: $\frac{1}{2}N(-1, (\frac{2}{3})^2) + \frac{1}{2}N(1, (\frac{2}{3})^2)$,

`D7 Separated bimodal`: $\frac{1}{2}N(-\frac{3}{2}, (\frac{1}{2})^2) + \frac{1}{2}N(\frac{3}{2}, (\frac{1}{2})^2)$,

`D8 Skewed bimodal`: $\frac{3}{4}N(0, 1) + \frac{1}{4}N(\frac{3}{2}, (\frac{1}{3})^2)$.

Thus, in total, we performed experiments in (2 sample sizes) $\times$ (2 scale functions) $\times$ (8 densities) = 32 conditions. Figures 4 and 5 show examples of the simulated data sets.

[5] In the literature of (unconditional) density estimation, considerable efforts have been devoted to compute approximate or asymptotic MISE for several density estimators. Computing asymptotic MISE of our KQR-path conditional density estimator is an important future works but lies outside the scope of this paper.

[6] We used the first 8 of the 15 normal mixture densities in Marron and Wand (1992).

5.1.2 Estimators

All through the experiments on KQR-path we used Gaussian kernel: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\{-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\gamma^2\}$. Thus, it has two hyper-parameters: the regularization parameter λ and the standard deviation of Gaussian kernel γ. We chose the pair of hyper-parameters from all possible 7×7 combinations of $\lambda \in \{0.1, 0.2, 0.5, 1, 2, 5, 10\}$ and $\gamma \in \{0.1, 0.2, 0.5, 1, 2, 5, 10\}$ using 10-fold cross-validation. We used finite-sample analogue of (15) as loss measure in the cross-validation. If $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_{\text{valid}}}$ is validation set, the validation error is given by $\sum_{i=1}^{n_{\text{valid}}} \int_0^1 \psi_\tau(y_i - f_\tau(\boldsymbol{x}_i))d\tau$. Note that we can easily compute the integral because both ψ and f_τ are piecewise-linear functions.

We controlled the capacity of MDN by number of hidden units and weight decay penalty. Thus, MDN also has a pair of hyper-parameters. Those pair of hyper-parameters were chosen from all possible 7×7 combinations of number of hidden units in $\{1, 2, 3, 4, 5, 7, 10\}$ and weight decay penalty coefficient in $\{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ using 10-fold cross-validation. Finite-sample analogue of (16) was used as loss measure in the cross-validation. The validation error is given by $-\sum_{i=1}^{n_{\text{valid}}} \log \hat{g}(y_i|\boldsymbol{x}_i)$, where $\hat{g}$ is the estimated conditional density by trained MDN. As discussed before, with the use of negative log-likelihood criterion, MDN tends to put spike densities at data points. One way to circumvent this phenomenon is to introduce a constraint that the variances of mixture components to be larger than a certain predefined value δ. All through the experiments on MDN in this paper, we employed this approach (we set $\delta = 10^{-4}$ in normalized scale). MDN was trained using Fletcher-Reeves conjugate gradient algorithm with Brent's line-search method (Press et al., 1992).

One of the important issues in mixture density estimation is how to determine the number of components. If one has some knowledge about the conditional densities, a simple way is to plot the densities with several number of components and choose the one that is most in accordance with one's prior ideas. This kind of subjective choice is not appropriate in exploratory nonparametric modeling framework, and we need to develop a method for data-dependent automatic selection. In our artificial data experiments, we examined three approaches. The first one is to use the *truth*. Since we generated data from normal mixture densities, we know the true number of components.Needless to say, this approach is not applicable to general situations, but it is informative to see the performances of MDN with optimal number of components. We call MDN with this approach as `MDN-OPT`. The second approach is to use *sufficiently many* components. Even when there are too many components, MDN has a potential to describe true mixture densities by assigning zero mixing proportions for redundant components. In our experiments we set the number of components to be 10 for this purpose. We call MDN with this second approach as `MDN-10`. The third approach is to use cross-validation. Note that this approach is extremely time-consuming because we now have three hyper-parameters for MDN. In our experiments we selected the number of components from $\{1, 2, ..., 10\}$. Thus, for each data set, (10 validation sets) $\times$ (7 candidates for number of hidden units) $\times$ (7 candidates for weight decay penalty coefficient) $\times$ (10 candidates for number of components) = 4,900 MDNs had to be trained. We call this time-consuming approach as `MDN-CV`.

5.1.3 Results

In each experimental condition we generated 10 data sets. The MISEs of `KQR-path` and each of three different MDNs, `MDN-OPT`, `MDN-10` and `MDN-CV`, were computed for each of 10 data sets. For numerical reasons, we approximated MISEs by mean square error at 50×500 meshed grid points in 2-dimensional input-output space $[-1, 1] \times [-3, 3]$. In each experimental condition MISEs of each 10 data sets were compared and the significance was evaluated using paired-sample t-test. Table 1 summarizes the results. The $\circ$ ($\bullet$) indicates the MISEs of KQR-path were smaller (larger) than those of MDN in average. Furthermore, $\circ\circ$, $\circ\circ\circ$ and $\circ\circ\circ\circ$ ($\bullet\bullet$, $\bullet\bullet\bullet$ and $\bullet\bullet\bullet\bullet$) indicate that the difference was significant at $p = 0.1$, 0.05 and 0.01, respectively.

The performances of `KQR-path` were comparable to those of `MDN-OPT`. Remember that the data sets were generated from normal mixture models and the parametric models for conditional distributions used in `MDN-OPT` were true ones. It indicates that our nonparametric conditional density estimator performed equally well to the true parametric counterpart. `KQR-path` were always better than `MDN-10` and the differences were almost always significant. It suggests that MDN performance is sensitive to the number of components. When the number of components were empirically estimated by cross-validation in

Table 1: Summary of the simulation results.

		KQR-path vs. MDN-OPT		KQR-path vs. MDN-10		KQR-path vs. MDN-CV	
		homo	hetero	homo	hetero	homo	hetero
$n = 100$	D1	●●●●	●●●●	○○○○	○○○○	○○○○	○○○
	D2	○	○	○○○○	○○○○	○○○	○○○○
	D3	○○○○	○○○	○○○○	○○○○	○○○○	○○○
	D4	○○○	○	○○○○	○○○○	○○○○	○○
	D5	○○○○	○○	○○○○	○○○○	○○○○	○○○
	D6	○	●●●	○○○○	○○○○	○○○○	○
	D7	○	●	○○○○	○○○○	○○○○	○○○○
	D8	○	●●●●	○○○○	○○○○	○○○○	○○
$n = 200$	D1	●●●●	●●●●	○○○	○○○	○	●
	D2	●	○	○○○○	○○○○	○	○
	D3	○○○○	○○○○	○○○○	○○○○	○○○○	○○○○
	D4	○○○	○○	○○○○	○	○○○○	○○○
	D5	○○○○	○○○○	○○	○○○○	○○○○	○○
	D6	●	●●●	○○○○	○○○○	○○○	●
	D7	●	●●●	○○○○	○○○○	○○○○	○
	D8	●●●●	●●●●	○○○○	○○○○	○○	○

MDN-CV, the performances were much improved. But KQR-path still worked better than MDN-CV in most experimental conditions and the differences were significant in many cases. The details of the results are given in appendix B. Figures 4 and 5 show examples of conditional density estimates by KQR-path.

5.2 Real data

Here we apply KQR-path and MDN to three real data sets. As discussed in the beginning of this section, there are no practical ways to evaluate the performances of density estimation without knowing the *true* density. The objective of this subsection is to illustrate the real-world applications and provide qualitative discussions on the properties of our conditional density estimator.

5.2.1 Bone mineral density (BMD) data set

In section 1, we provided an example on bone mineral density (BMD) data to illustrate conditional density estimation problems. We describe the detail here. The sample size is $n = 485$. We normalized both of the input and the output variables so that they have zero mean and unit variance.

In KQR-path estimator, we used Gaussian kernel. The pair of hyper-parameters were selected using 10-fold cross-validation in the same way as simulation studies in 5.1, and $\lambda = 0.1$ and $\gamma = 1.0$ were selected. The algorithm found $L = 1349$ event points. The average and maximum sizes of $\{\mathcal{I}^0_{\tau_\ell}\}_{\ell=1}^L$ were 5.59 and 9, respectively. Figure 6(a) shows the estimated conditional densities for $x = 12$, 18, and 24.

We also applied MDN to BMD data. The number of mixture components, number of hidden units, and weight decay penalty parameter were determined by 10-fold cross-validation in the same way as MDN-CV in 5.1. The selected number of components was 5, the number of hidden units was 4, and the weight decay penalty parameter was 1.0. We trained several MDNs with different initial parameters (weights), and slightly different but qualitatively similar results were obtained. Figure 6(b) shows an example of the estimated conditional densities by MDN.

As discussed above, it is difficult to assess the density estimation performances. At first sight, the densities estimated by KQR-path seems to be slightly over-smoothed and those by MDN look under-smoothed. For example, $\hat{p}(y|x = 12)$ by KQR-path clearly suggests bi-modality but the corresponding density by MDN is unimodal. It is important to know that whether or not these two modes are really present or are they just artifacts. An answer is available in the present example by separating the data set according to sex. Figure 7 shows the estimated densities for each of the male and female samples. In both of KQR-path (a) and MDN (b), estimated conditional densities of male and female sample for $x = 12$ were unimodal and their modes seem to have correspondence with the two modes in Figure 6(a). It indicates that the two modes suggested by KQR-path estimator are true important features of the conditional density $p(y|x = 12)$.

5.2.2 Geiser data set

Next, we applied KQR-path to *Geiser* data taken from Azzalini and Bowman (1990). This data is on the waiting time between the starts of successive eruptions and the duration of the subsequent eruption for the old faithful geyser in Yellowstone National Park. Figure 8 shows a scatter plot of the data. It is clear that the duration of the next eruption is relatively long if the waiting time was shorter than about 70 minutes. But, when the waiting time is longer than about 70 minutes, the duration of the next eruption seems to be a mixture of short and long durations. The sample size is $n = 299$. We normalized the input and the output variables so that they have zero mean and unit variance.

In KQR-path, we used Gaussian kernel. The pair of hyper-parameters were selected using 10-fold cross-validation in the same way as simulation studies in 5.1, and $\lambda = 0.2$ and $\gamma = 0.2$ were selected. The algorithm found $L = 788$ event points. The average and maximum sizes of $\{\mathcal{I}^0_{\tau_\ell}\}_{\ell=1}^L$ were 24.99 and 34, respectively. Figure 8(a) shows the estimated conditional densities of the output y given the input $x = 50$, 65, and 80.

In MDN, the number of components, number of hidden units, and weight decay penalty parameter were determined by 10-fold cross-validation in the same way as `MDN-CV` in 5.1. The selected number of components was 5, the number of hidden units was 5, and the weight decay penalty parameter was 1.0. We trained several MDNs with different initial parameters (weights), and slightly different but qualitatively similar results were obtained. Figure 8(b) shows an example of the estimated conditional densities by MDN.

We analyzed this data to examine how KQR-path and MDN perform when conditional density sharply changes from unimodal to bimodal. KQR-path estimated unimodal densities when $x = 50$ and 65 (ignoring minor zig-zags), and bimodal density when $x = 80$. MDN also successfully estimated bimodal density when $x = 80$ (ignoring spike density region at $y = 4$, which will be discussed later). But it falsely generated small modes around $y = 2$ also when $x = 50$ and 65. Spike densities at $y = 4$ in MDN estimates are not necessarily artifacts. When data was collected in Azzalini and Bowman (1990), some duration times were recorded as L(long), S(short) and M(medium) and later 4, 2, and 3 were numerically assigned, respectively. Thus, there are many points with $y = 4$ and we can interpret that MDN successfully found true high density region.

5.2.3 Boston housing data set

So far, we studied univariate input data set. Our conditional density estimator can be straightforwardly applied to high-dimensional input data set. To illustrate this, we apply KQR-path and MDN to well-known *Boston housing* data set. The sample size is $n = 506$ and the input dimension is 13. We normalized all the input and output variables so that they have zero mean and unit variance.

In KQR-path, we used Gaussian kernel. The pair of hyper-parameters were selected using 10-fold cross-validation in the same way as simulation studies in 5.1, and $\lambda = 0.2$ and $\gamma = 5.0$ were selected. The algorithm found $L = 1210$ event points. The average and maximum sizes of $\{\mathcal{I}^0_{\tau_\ell}\}_{\ell=1}^L$ were 41.78 and 58, respectively. Figure 9(a) shows the estimated conditional densities of the output y given the three input vectors $\boldsymbol{x}_{i_{(0.25)}}$, $\boldsymbol{x}_{i_{(0.50)}}$ and $\boldsymbol{x}_{i_{(0.75)}}$, where $i_{(0.25)}$, $i_{(0.50)}$ and $i_{(0.75)}$ are the indices of training data set such that their corresponding outputs $y_{i_{(0.25)}}$, $y_{i_{(0.50)}}$ and $y_{i_{(0.75)}}$ are 25, 50 and 75 percentiles of $\{y_1, ..., y_n\}$.

In MDN, the number of components, number of hidden units, and weight decay penalty parameter were determined by 10-fold cross-validation in the same way as `MDN-CV` in 5.1. The selected number of components was 2, the number of hidden units was 1, and the weight decay penalty parameter was 1.0. We trained several MDNs with different initial parameters (weights), and slightly different but qualitatively similar results were obtained. Figure 9(b) shows an example of the estimated conditional densities of the output y given the three input vectors $\boldsymbol{x}_{i_{(0.25)}}$, $\boldsymbol{x}_{i_{(0.50)}}$ and $\boldsymbol{x}_{i_{(0.75)}}$.

6 Conclusions

We have presented an algorithm to obtain the entire solution path for kernel-based quantile regression (KQR). The algorithm is based on the piecewise-linearity of the solution path. Unlike the previous study (Hastie et al., 2004; Li et al., 2007) whose goal is to obtain solution path with respect to the regularization parameter λ, our approach provides the KQR solution path with respect to the quantile order $\tau \in (0, 1)$. It enables us to estimate the cumulative distribution function (CDF) of conditional density in piecewise-linear form. After smoothing the CDF, we obtain nonparametric conditional density estimate. Conditional density estimation is an effective tool for exploratory data analysis. It provides deeper insight into the underlying mechanism of the data than conventional regression analysis does.

Relatively few studies have been reported on conditional density estimation for high-dimensional data. Some conditional density estimators have been studied in the framework of nonparametric smoothing, but they are applicable only to one or two-dimensional input data because of the curse of dimensionality. In machine learning literature, several efforts have been devoted to estimate input-dependent parameters of (parametric) conditional density as the outputs of multi-layer-perceptron (MLP). Among those approaches mixture density network (MDN) is especially useful. It describes conditional density by mixture of parametric distributions and all the involving parameters are estimated as the outputs of MLP.

Our conditional density estimator based on KQR-path has several advantages. It inherits many desirable properties of kernel-machines. The solution is (almost) unique and guaranteed to be globally (not locally) optimal because it is formulated as convex optimization problem. It can be applied to high-dimensional or structured input problems without much increasing computational complexity. In addition, numerical experiments in section 5 demonstrate the relative efficiency of KQR-path estimator. Total computational cost (including cross-validation for determining hyper-parameters) of KQR-path was much smaller than that of MDN.

Although both KQR-path and MDN (and other similar approaches) have a common goal, the criterions used for their model fittings are totally different. The former is trying to find the *true* CDF of conditional density by minimizing the integrated quantile error (15). On the other hand, the latter is trying to minimize the Kullback-Leibler distance between *true* and estimated conditional densities by minimizing the negative-log likelihood (16). As discussed in section 5, neither is perfectly appropriate measure for density estimation, and it is difficult to make clear-cut conclusion that one is better than the other.

Conditional density estimation has two different smoothing phases. The first phase is the smoothing in $\boldsymbol{x}$-direction. The *smoothing* here is used in the same sense as in nonparametric regression. The second phase is the smoothing in y-direction, in which the *smoothing* is used as in the context of **un**conditional density estimation. Actually, our main contribution in this paper is only in the former and one can use any other **un**conditional density estimator for the latter. Our KQR path algorithm is considered as an conditional CDF estimator with an ability to smooth in $\boldsymbol{x}$-direction. This *modularity* is helpful because we can control the degrees of smoothing independently in $\boldsymbol{x}$ and y-directions. On the other hand, MDN or other similar approaches perform smoothing both in $\boldsymbol{x}$ and y-directions simultaneously. Thus, their capacity control is more challenging and tends to have more hyper-parameters than our approach.

One of the limitations of KQR-path estimator is that it cannot be straightforwardly extended for high-dimensional **out**put problems. In other words, our approach can only be applied to univariate conditional density and not to multivariate conditional density. Nonparametric multivariate density estimation is challenging problem, but there are many important applications.

There remains to be several problems to improve the presented methodology. In terms of the algorithm, as discussed in section 3, we need to develop a way to deal with some degenerate situations.

Current piecewise-linear path-following algorithm in kernel machines including ours have a possibility to fail (although rarely in practice) in certain degenerate situations. To assess the estimation performances, it might be helpful if we could compute asymptotic mean integrated square error (MISE) of our estimator. In addition, application to practical problems, especially to data sets with high-dimensional inputs, is an important step for maturing our methodology.

A Proofs

A.1 Proof of Lemma 1

Proof. Let n_τ^+ and n_τ^- be the sizes of $\mathcal{I}_\tau^+$ and $\mathcal{I}_\tau^-$, respectively. If $\mathcal{I}_\tau^0$ is empty, the summation constraint (6d) is represented as

$$\sum_{i=1}^{n} \alpha_{\tau i} = \tau n_\tau^+ + (\tau - 1) n_\tau^- = \tau n - n_\tau^- = 0 \iff \tau = \frac{n_\tau^-}{n},$$

where $n_\tau^- \in \{0, 1, 2, ...n\}$. □

A.2 Proof of Lemma 2

Proof. We prove only the first part. For $\tau \in (\tau_{l^*-1}, \tau_{l^*})$, $\mathcal{I}_\tau^0$ contains only one element, say $\tilde{i}$. In this interval, (10) is represented as

$$\begin{bmatrix} 0 & 1 \\ 1 & K_{\tilde{i}\tilde{i}} \end{bmatrix} \begin{bmatrix} \alpha_{\tau 0} - \alpha_{\tau_{l^*-1} 0} \\ \alpha_{\tau \tilde{i}} - \alpha_{\tau_{l^*-1} \tilde{i}} \end{bmatrix} = \begin{bmatrix} 1 - n \\ -\sum_{i \neq \tilde{i}} K_{i\tilde{i}} \end{bmatrix} (\tau - \tau_{l^*-1}).$$

Solving this 2-by-2 linear system of equations,

$$\alpha_{\tau \tilde{i}} = \alpha_{\tau_{l^*-1} \tilde{i}} + (1 - n)(\tau - \tau_{l^*-1}).$$

It shows that $\alpha_{\tau \tilde{i}}$ is linearly decreasing function of τ in the interval $(\tau_{l^*-1}, \tau_{l^*})$ because the sample size $n > 1$. Remembering that $\tilde{i}$ moves into $\mathcal{I}_{\tau^*}^-$ if $\alpha_{\tau^* \tilde{i}} = 1 - \tau^*$ and moves into $\mathcal{I}_{\tau^*}^+$ if $\alpha_{\tau^* \tilde{i}} = \tau^*$, $\tilde{i}$ must move to $\mathcal{I}_{\tau^*}^-$, not to $\mathcal{I}_{\tau^*}^+$, at l^*-th event. Suppose that $\tilde{i}$ is any element in $\mathcal{I}_{\tau^*}^-$ other than i_{max}^-. Then the residual of i_{max}^- is

$$y_{i_{max}^-} - \frac{1}{\lambda}\{\sum_{j=1}^{n} \alpha_{\tau^* j} K_{i_{max}^- j}\} > y_{\tilde{i}} - \frac{1}{\lambda}\{\sum_{j=1}^{n} \alpha_{\tau^* j} K_{\tilde{i}j}\} = 0.$$

Since this contradicts the fact that $i_{max}^- \in \mathcal{I}_{\tau^*}^-$, we conclude that $\tilde{i} = i_{max}^-$. This proves the first part of lemma 2. The second part can be proved similarly. □

A.3 Proof of Theorem 1

Proof. From section 3, τ-th conditional quantile model f_τ is represented as

$$\begin{aligned}
f_\tau(\boldsymbol{x}_0) &= \frac{1}{\lambda}\{\sum_{i=1}^{n}\alpha_{\tau i}K(\boldsymbol{x}_0,\boldsymbol{x}_i)+\alpha_{\tau 0}\} \\
&= \frac{1}{\lambda}\{\sum_{i=1}^{n}[\alpha_{\tau_\ell i}+c_{\ell i}(\tau-\tau_\ell)]K(\boldsymbol{x}_0,\boldsymbol{x}_i)+[\alpha_{\tau_\ell 0}+c_{\ell 0}(\tau-\tau_\ell)]\} \\
&= \frac{1}{\lambda}\{\sum_{i=1}^{n}[\frac{\tau_{\ell+1}-\tau}{\tau_{\ell+1}-\tau_\ell}\alpha_{\tau_\ell i}+\frac{\tau-\tau_\ell}{\tau_{\ell+1}-\tau_\ell}\alpha_{\tau_{\ell+1}i}]K(\boldsymbol{x}_0,\boldsymbol{x}_i) \\
&\quad + [\frac{\tau_{\ell+1}-\tau}{\tau_{\ell+1}-\tau_\ell}\alpha_{\tau_\ell 0}+\frac{\tau-\tau_\ell}{\tau_{\ell+1}-\tau_\ell}\alpha_{\tau_{\ell+1}0}]\} \\
&= \frac{\tau_{\ell+1}-\tau}{\tau_{\ell+1}-\tau_\ell}\cdot\frac{1}{\lambda}\{\sum_{i=1}^{n}\alpha_{\tau_\ell i}K(\boldsymbol{x}_0,\boldsymbol{x}_i)+\alpha_{\tau_\ell 0}\} \\
&\quad + \frac{\tau-\tau_\ell}{\tau_{\ell+1}-\tau_\ell}\cdot\frac{1}{\lambda}\{\sum_{i=1}^{n}\alpha_{\tau_{\ell+1}i}K(\boldsymbol{x}_0,\boldsymbol{x}_i)+\alpha_{\tau_{\ell+1}0}\} \\
&= \frac{\tau_{\ell+1}-\tau}{\tau_{\ell+1}-\tau_\ell}\cdot f_{\tau_\ell}(\boldsymbol{x}_0)+\frac{\tau-\tau_\ell}{\tau_{\ell+1}-\tau_\ell}\cdot f_{\tau_{\ell+1}}(\boldsymbol{x}_0),
\end{aligned}$$

where we used $c_{\ell i}=(\alpha_{\tau_{\ell+1}i}-\alpha_{\tau_\ell i})/(\tau_{\ell+1}-\tau_\ell)$ in the third equality. It tells that the τ-the quantile of $p(y|\boldsymbol{x}=\boldsymbol{x}_0)$ is represented as linear interpolation of τ_ℓ-th and $\tau_{\ell+1}$-th quantiles of $p(y|\boldsymbol{x}=\boldsymbol{x}_0)$ with an ratio $\frac{\tau_{\ell+1}-\tau}{\tau_{\ell+1}-\tau_\ell}:\frac{\tau-\tau_\ell}{\tau_{\ell+1}-\tau_\ell}$. Noting that CDF is just the inverse of the quantile function, the proof is completed. □

A.4 Proof of Theorem 3

Proof. We use standard bounding trick that

$$\begin{aligned}
\sup_\tau(R_\tau[\hat{f}_\tau]-R_\tau[f_\tau]) &\le \sup_\tau(R_{\tau,\mathrm{reg}}[\hat{f}_\tau]-R_{\tau,\mathrm{reg}}[f_\tau])+\frac{\lambda_n\bar{A}^2}{2} \\
&\le 2\sup_{\tau,f}(\widehat{R}_{\tau,\mathrm{reg}}[f]-R_{\tau,\mathrm{reg}}[f])+\frac{\varepsilon}{2}.
\end{aligned}$$

Applying Theorem 2, we complete the proof. □

B Detailed Results on Simulation Study

Tables 2, 3 and 4 show the detailed results on the simulation study in 5.1. In these tables, the figures indicate the average and standard error of the mean integrated squared errors (MISE) for 10 data sets. We assessed the significances of the results using paired-sample t-test (`t`) and Wicoxon signed-rank test (`signrank`).

Table 2: Detailed simulation results (KQR-path vs. MDN-OPT)

	homo ($n = 100$)			
	MISE		Significance (p-value)	
	KQR-path	MDN-OPT	t	signrank
D1	0.0385 ± 0.0039	0.0176 ± 0.0037	0.0003	0.0039
D2	0.0545 ± 0.0085	0.0733 ± 0.0108	0.1478	0.1934
D3	0.1460 ± 0.0034	0.2581 ± 0.0140	0.0000	0.0020
D4	0.1566 ± 0.0108	0.2094 ± 0.0171	0.0257	0.0371
D5	0.4994 ± 0.0671	1.0334 ± 0.0849	0.0002	0.0020
D6	0.0342 ± 0.0029	0.0375 ± 0.0048	0.5410	0.6250
D7	0.0492 ± 0.0052	0.0595 ± 0.0107	0.3796	0.9219
D8	0.0404 ± 0.0034	0.0525 ± 0.0087	0.1865	0.1602
	hetero ($n = 100$)			
	MISE		Significance (p-value)	
	KQR-path	MDN-OPT	t	signrank
D1	0.0357 ± 0.0046	0.0154 ± 0.0039	0.0005	0.0020
D2	0.0546 ± 0.0083	0.0579 ± 0.0083	0.7170	0.6953
D3	0.1393 ± 0.0077	0.2922 ± 0.0608	0.0211	0.0020
D4	0.1445 ± 0.0110	0.1682 ± 0.0068	0.1304	0.1055
D5	0.4926 ± 0.0588	0.7201 ± 0.0829	0.0514	0.0273
D6	0.0379 ± 0.0034	0.0297 ± 0.0030	0.0492	0.0840
D7	0.0509 ± 0.0060	0.0397 ± 0.0048	0.1418	0.1309
D8	0.0385 ± 0.0037	0.0253 ± 0.0016	0.0071	0.0020
	homo ($n = 200$)			
	MISE		Significance (p-value)	
	KQR-path	MDN-OPT	t	signrank
D1	0.0309 ± 0.0035	0.0114 ± 0.0020	0.0007	0.0020
D2	0.0428 ± 0.0049	0.0329 ± 0.0045	0.1251	0.1602
D3	0.1284 ± 0.0066	0.1830 ± 0.0086	0.0024	0.0039
D4	0.1383 ± 0.0108	0.1673 ± 0.0066	0.0122	0.0371
D5	0.4403 ± 0.0546	0.8649 ± 0.0844	0.0025	0.0059
D6	0.0251 ± 0.0022	0.0219 ± 0.0010	0.2298	0.3223
D7	0.0338 ± 0.0026	0.0294 ± 0.0025	0.1654	0.2324
D8	0.0277 ± 0.0012	0.0245 ± 0.0012	0.0075	0.0137
	hetero ($n = 200$)			
	MISE		Significance (p-value)	
	KQR-path	MDN-OPT	t	signrank
D1	0.0320 ± 0.0043	0.0129 ± 0.0043	0.0068	0.0039
D2	0.0459 ± 0.0057	0.0486 ± 0.0079	0.7582	0.8457
D3	0.1250 ± 0.0092	0.1729 ± 0.0141	0.0078	0.0020
D4	0.1375 ± 0.0105	0.1511 ± 0.0073	0.0844	0.1309
D5	0.4341 ± 0.0602	0.7727 ± 0.0703	0.0019	0.0020
D6	0.0286 ± 0.0031	0.0201 ± 0.0010	0.0262	0.0371
D7	0.0375 ± 0.0038	0.0247 ± 0.0025	0.0124	0.0137
D8	0.0295 ± 0.0026	0.0222 ± 0.0024	0.0047	0.0039

Table 3: Detailed simulation results (`KQR-path` vs. `MDN-10`)

	`homo` ($n = 100$)			
	MISE		Significance (p-value)	
	`KQR-path`	`MDN-10`	`t`	`signrank`
`D1`	0.0385 ± 0.0039	0.1821 ± 0.0325	0.0023	0.0020
`D2`	0.0545 ± 0.0085	0.1825 ± 0.0271	0.0019	0.0020
`D3`	0.1460 ± 0.0034	0.2832 ± 0.0202	0.0002	0.0020
`D4`	0.1566 ± 0.0108	0.2448 ± 0.0187	0.0010	0.0039
`D5`	0.4994 ± 0.0671	0.9187 ± 0.0946	0.0017	0.0020
`D6`	0.0342 ± 0.0029	0.1305 ± 0.0200	0.0012	0.0020
`D7`	0.0492 ± 0.0052	0.1500 ± 0.0225	0.0008	0.0020
`D8`	0.0404 ± 0.0034	0.1584 ± 0.0289	0.0029	0.0039
	`hetero` ($n = 100$)			
	MISE		Significance (p-value)	
	`KQR-path`	`MDN-10`	`t`	`signrank`
`D1`	0.0357 ± 0.0046	0.1441 ± 0.0192	0.0003	0.0020
`D2`	0.0546 ± 0.0083	0.1905 ± 0.0236	0.0002	0.0020
`D3`	0.1393 ± 0.0077	0.3087 ± 0.0351	0.0009	0.0020
`D4`	0.1445 ± 0.0110	0.3247 ± 0.0421	0.0025	0.0020
`D5`	0.4926 ± 0.0588	0.8179 ± 0.0961	0.0100	0.0137
`D6`	0.0379 ± 0.0034	0.1111 ± 0.0194	0.0055	0.0020
`D7`	0.0509 ± 0.0060	0.1722 ± 0.0306	0.0036	0.0020
`D8`	0.0385 ± 0.0037	0.1445 ± 0.0224	0.0017	0.0020
	`homo` ($n = 200$)			
	MISE		Significance (p-value)	
	`KQR-path`	`MDN-10`	`t`	`signrank`
`D1`	0.0309 ± 0.0035	0.0626 ± 0.0077	0.0100	0.0195
`D2`	0.0428 ± 0.0049	0.0821 ± 0.0083	0.0047	0.0039
`D3`	0.1284 ± 0.0066	0.1906 ± 0.0100	0.0000	0.0020
`D4`	0.1383 ± 0.0108	0.1901 ± 0.0070	0.0000	0.0020
`D5`	0.4403 ± 0.0546	0.6375 ± 0.1023	0.0948	0.1309
`D6`	0.0251 ± 0.0022	0.0759 ± 0.0096	0.0003	0.0020
`D7`	0.0338 ± 0.0026	0.0795 ± 0.0091	0.0007	0.0020
`D8`	0.0277 ± 0.0012	0.0853 ± 0.0099	0.0003	0.0020
	`hetero` ($n = 200$)			
	MISE		Significance (p-value)	
	`KQR-path`	`MDN-10`	`t`	`signrank`
`D1`	0.0320 ± 0.0043	0.0483 ± 0.0060	0.1069	0.1602
`D2`	0.0459 ± 0.0057	0.0774 ± 0.0078	0.0380	0.0488
`D3`	0.1250 ± 0.0092	0.1634 ± 0.0103	0.0039	0.0059
`D4`	0.1375 ± 0.0105	0.1926 ± 0.0105	0.0003	0.0039
`D5`	0.4341 ± 0.0602	0.6010 ± 0.0504	0.0665	0.0840
`D6`	0.0286 ± 0.0031	0.0444 ± 0.0057	0.0216	0.0273
`D7`	0.0375 ± 0.0038	0.0740 ± 0.0097	0.0145	0.0273
`D8`	0.0295 ± 0.0026	0.0642 ± 0.0084	0.0038	0.0020

Table 4: Detailed simulation results (`KQR-path` vs. `MDN-CV`)

	`homo` ($n = 100$)			
	MISE		Significance (p-value)	
	`KQR-path`	`MDN-CV`	`t`	`signrank`
`D1`	0.0385 ± 0.0039	0.1005 ± 0.0182	0.0031	0.0039
`D2`	0.0545 ± 0.0085	0.0899 ± 0.0135	0.0328	0.0371
`D3`	0.1460 ± 0.0034	0.2377 ± 0.0193	0.0008	0.0020
`D4`	0.1566 ± 0.0108	0.2485 ± 0.0160	0.0001	0.0020
`D5`	0.4994 ± 0.0671	0.8535 ± 0.0578	0.0015	0.0059
`D6`	0.0342 ± 0.0029	0.1048 ± 0.0165	0.0022	0.0039
`D7`	0.0492 ± 0.0052	0.1197 ± 0.0172	0.0015	0.0020
`D8`	0.0404 ± 0.0034	0.1139 ± 0.0196	0.0057	0.0098
	`hetero` ($n = 100$)			
	MISE		Significance (p-value)	
	`KQR-path`	`MDN-CV`	`t`	`signrank`
`D1`	0.0357 ± 0.0046	0.1781 ± 0.0560	0.0334	0.0098
`D2`	0.0546 ± 0.0083	0.1338 ± 0.0201	0.0082	0.0039
`D3`	0.1393 ± 0.0077	0.2107 ± 0.0197	0.0136	0.0039
`D4`	0.1445 ± 0.0110	0.2933 ± 0.0716	0.0832	0.0195
`D5`	0.4926 ± 0.0588	0.7522 ± 0.0701	0.0129	0.0195
`D6`	0.0379 ± 0.0034	0.1054 ± 0.0464	0.1919	0.0840
`D7`	0.0509 ± 0.0060	0.1372 ± 0.0257	0.0093	0.0137
`D8`	0.0385 ± 0.0037	0.0520 ± 0.0065	0.0713	0.0645
	`homo` ($n = 200$)			
	MISE		Significance (p-value)	
	`KQR-path`	`MDN-CV`	`t`	`signrank`
`D1`	0.0309 ± 0.0035	0.0442 ± 0.0092	0.2139	0.2754
`D2`	0.0428 ± 0.0049	0.0650 ± 0.0129	0.1244	0.1934
`D3`	0.1284 ± 0.0066	0.1560 ± 0.0063	0.0011	0.0020
`D4`	0.1383 ± 0.0108	0.1808 ± 0.0105	0.0031	0.0059
`D5`	0.4403 ± 0.0546	0.7700 ± 0.0994	0.0081	0.0098
`D6`	0.0251 ± 0.0022	0.0459 ± 0.0070	0.0234	0.0273
`D7`	0.0338 ± 0.0026	0.0554 ± 0.0056	0.0056	0.0059
`D8`	0.0277 ± 0.0012	0.0587 ± 0.0140	0.0520	0.0195
	`hetero` ($n = 200$)			
	MISE		Significance (p-value)	
	`KQR-path`	`MDN-CV`	`t`	`signrank`
`D1`	0.0320 ± 0.0043	0.0248 ± 0.0060	0.3435	0.2754
`D2`	0.0459 ± 0.0057	0.0668 ± 0.0173	0.1628	0.3223
`D3`	0.1250 ± 0.0092	0.1542 ± 0.0097	0.0041	0.0098
`D4`	0.1375 ± 0.0105	0.1710 ± 0.0083	0.0162	0.0137
`D5`	0.4341 ± 0.0602	0.6397 ± 0.0589	0.0655	0.1934
`D6`	0.0286 ± 0.0031	0.0280 ± 0.0042	0.9010	0.4922
`D7`	0.0375 ± 0.0038	0.0746 ± 0.0259	0.1738	0.1055
`D8`	0.0295 ± 0.0026	0.0415 ± 0.0109	0.3329	0.3750

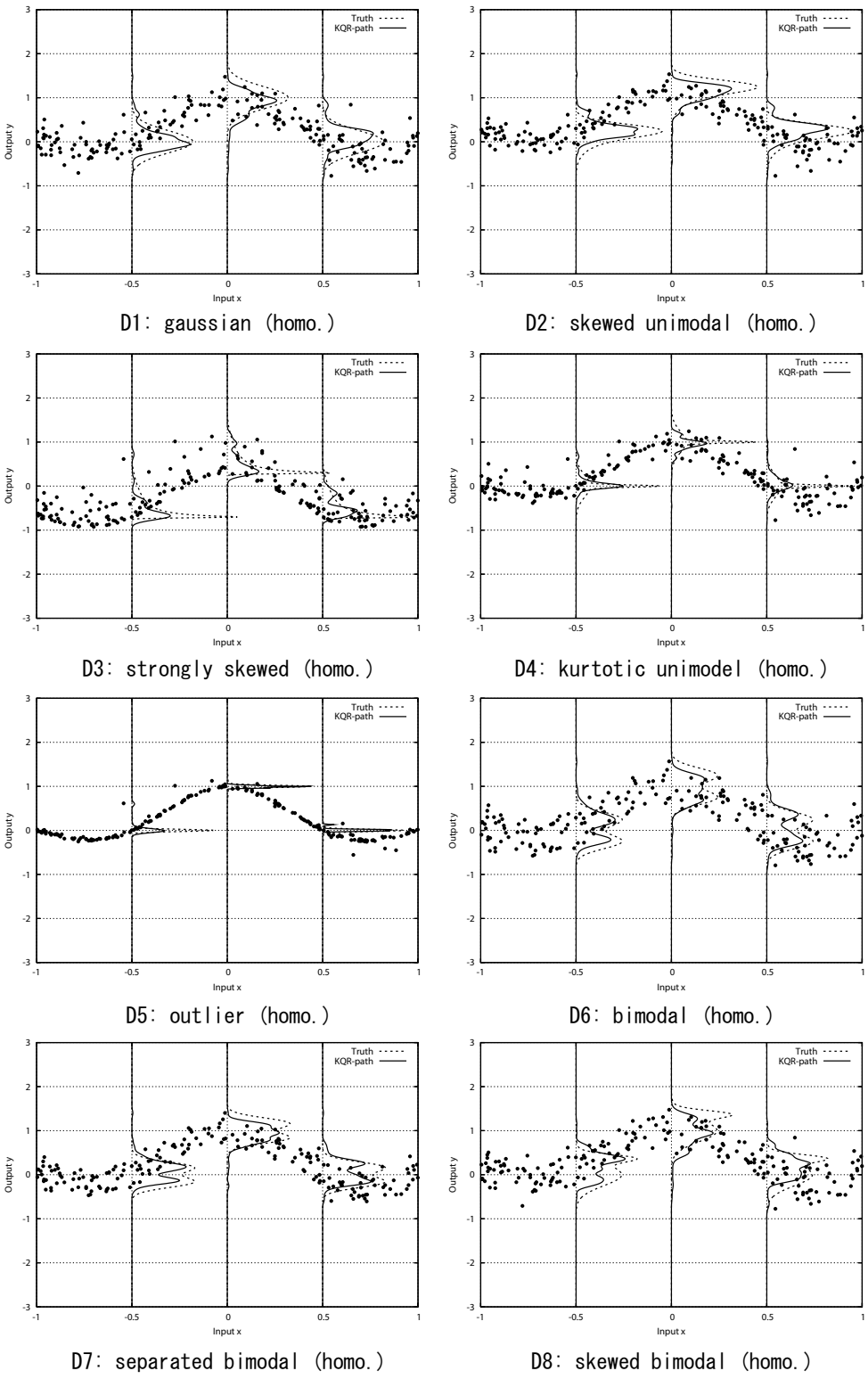


Figure 4: Conditional density estimates by `KQR-path` in `homo`scedatic simulated data sets.

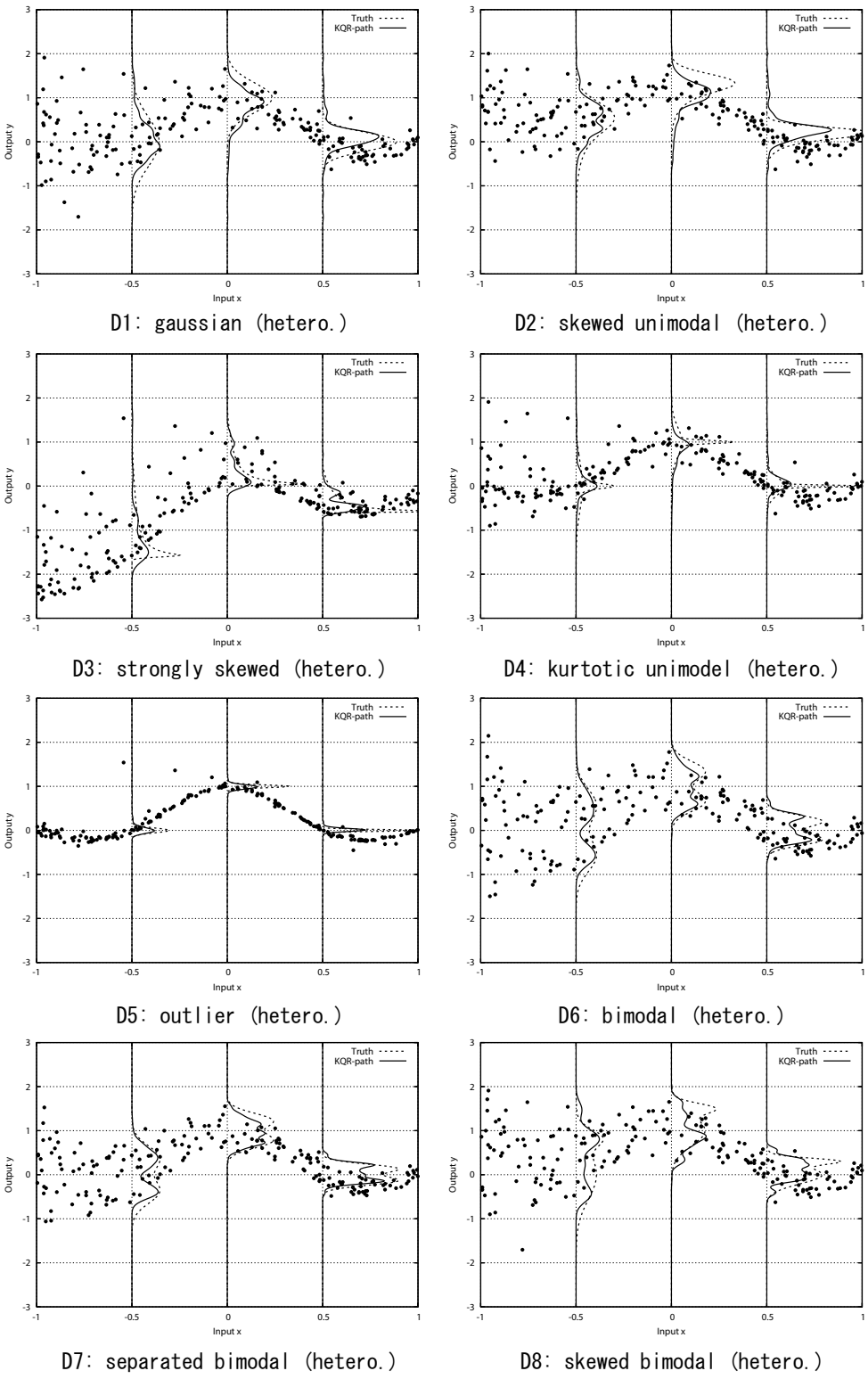
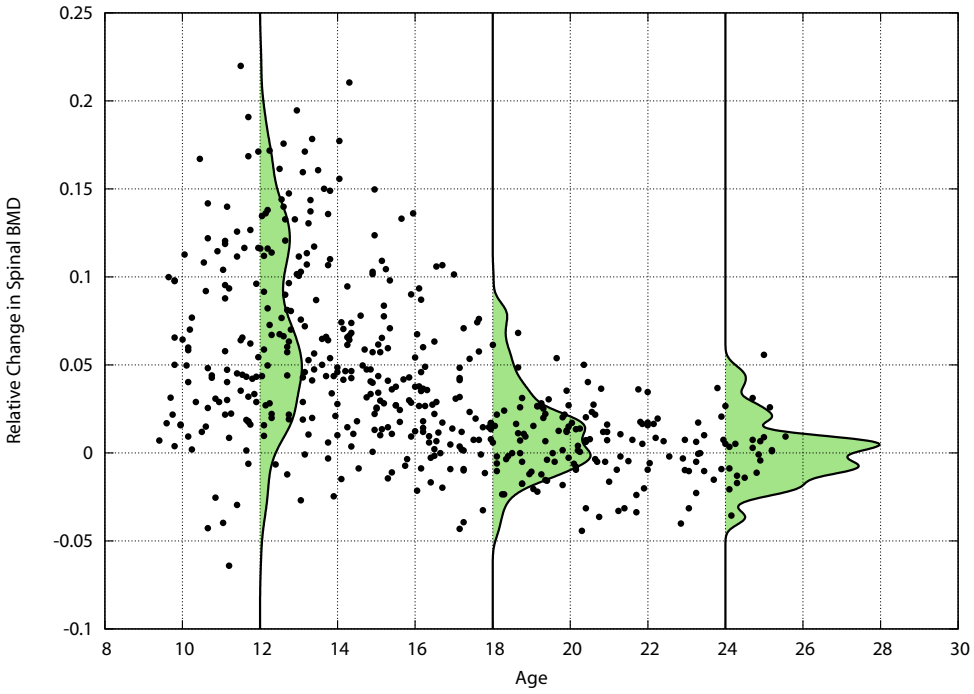
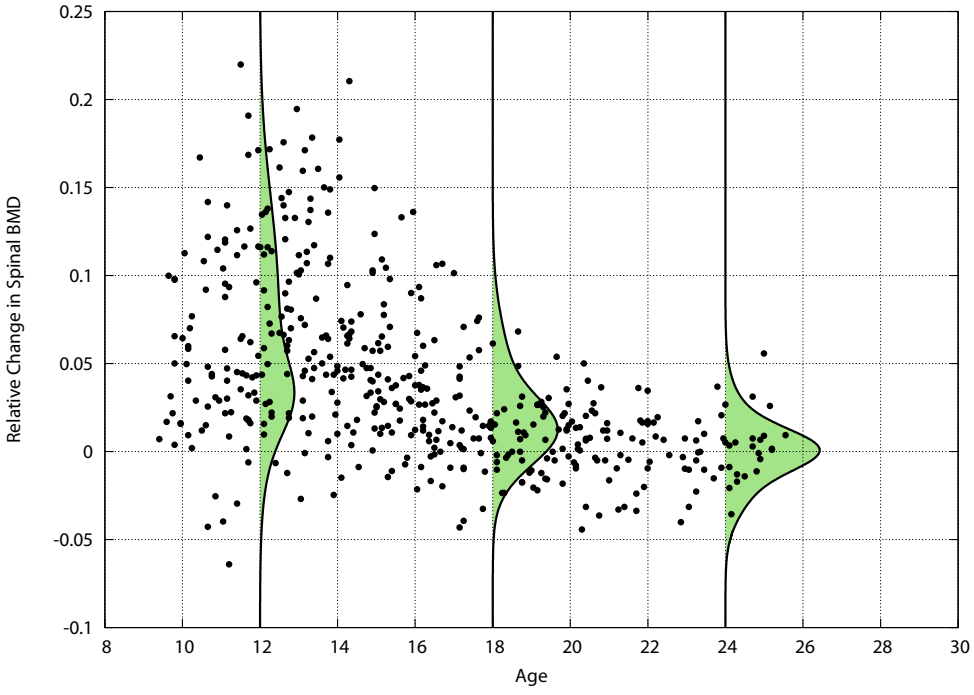


Figure 5: Conditional density estimates by `KQR-path` in `hetero`scedatic simulated data sets.

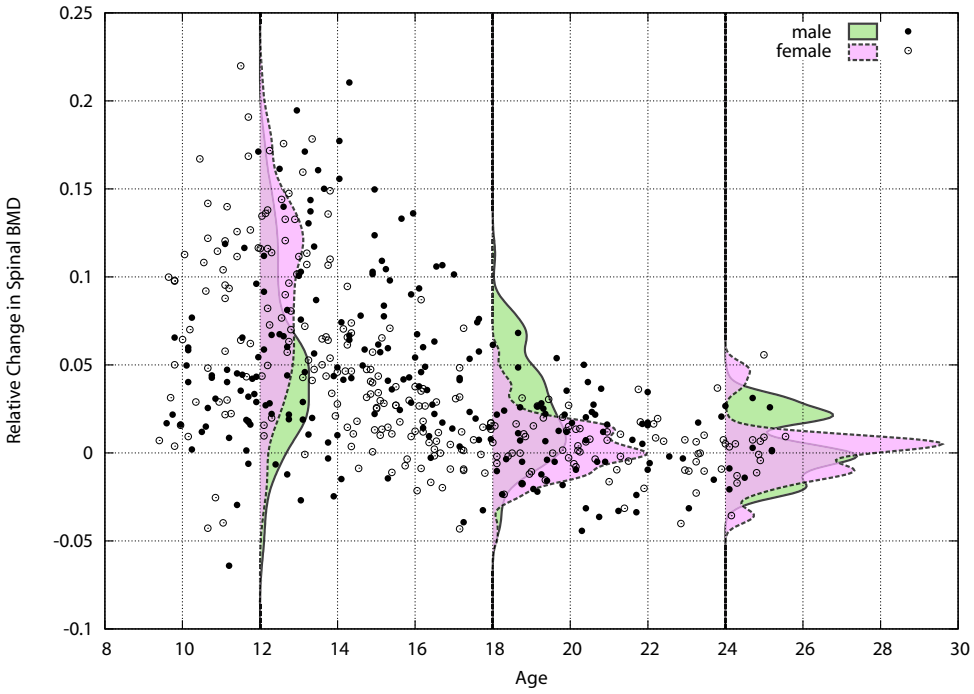


(a) KQR-path

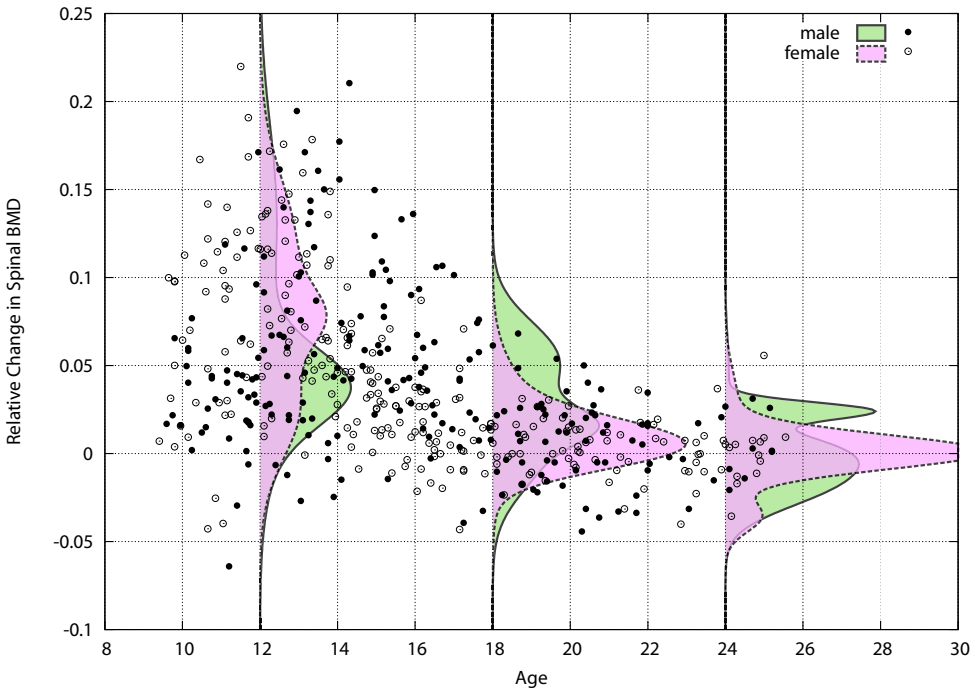


(b) MDN(-CV)

Figure 6: Conditional density estimates in BMD data set.

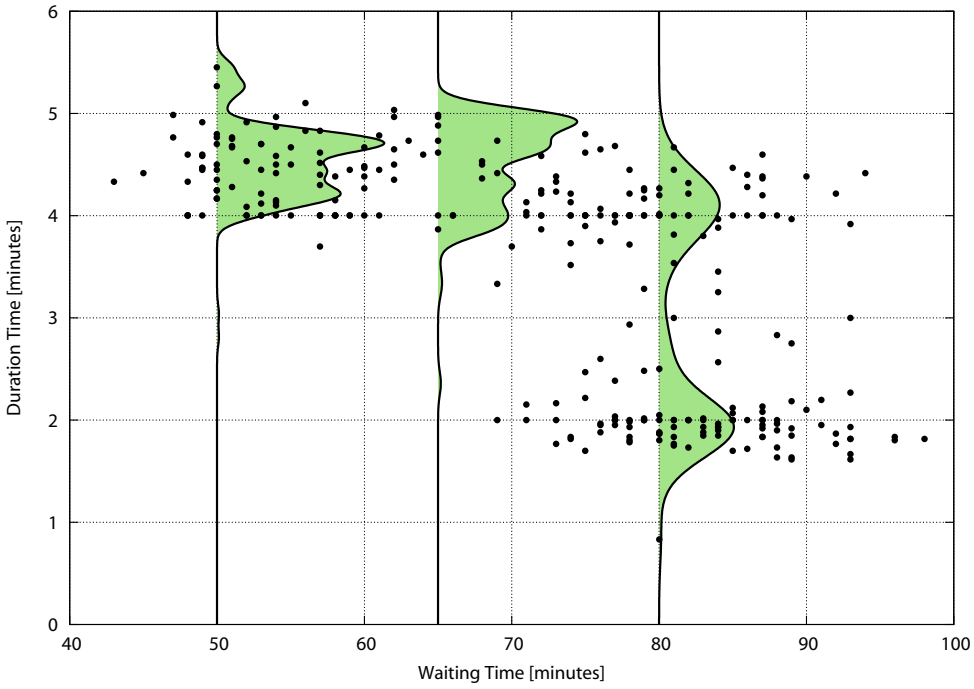


(a) KQR-path

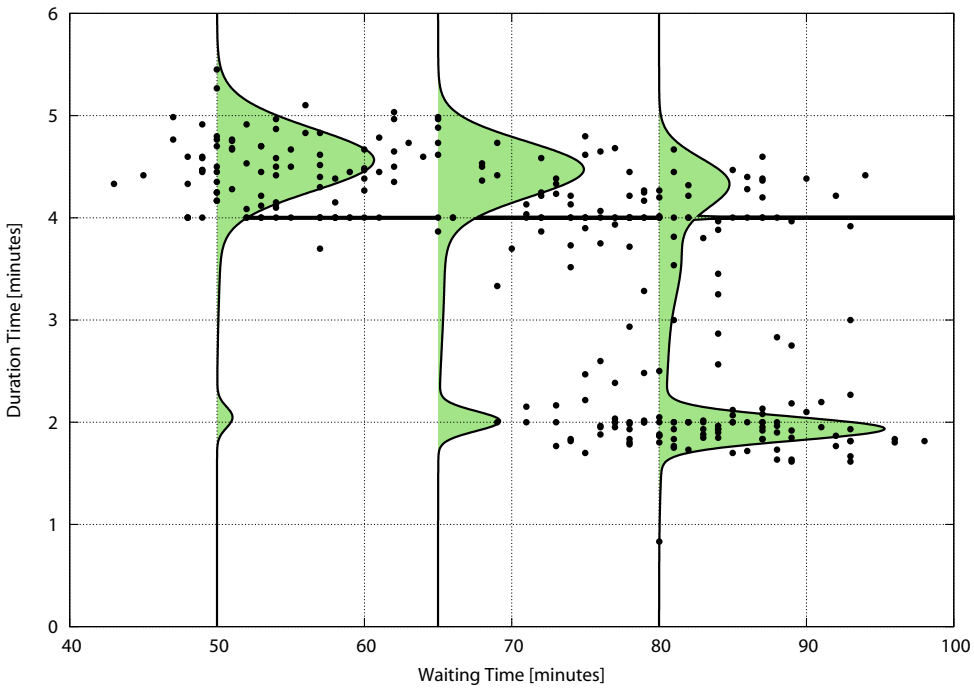


(b) MDN(-CV)

Figure 7: Conditional density estimates of male and female samples in BMD data set.

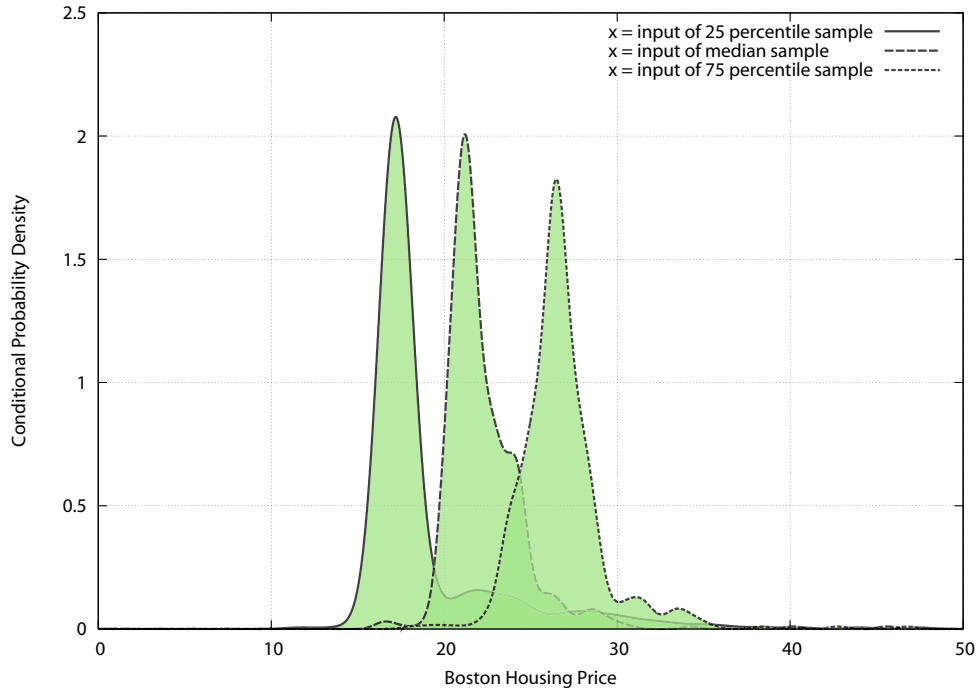


(a) KQR-path

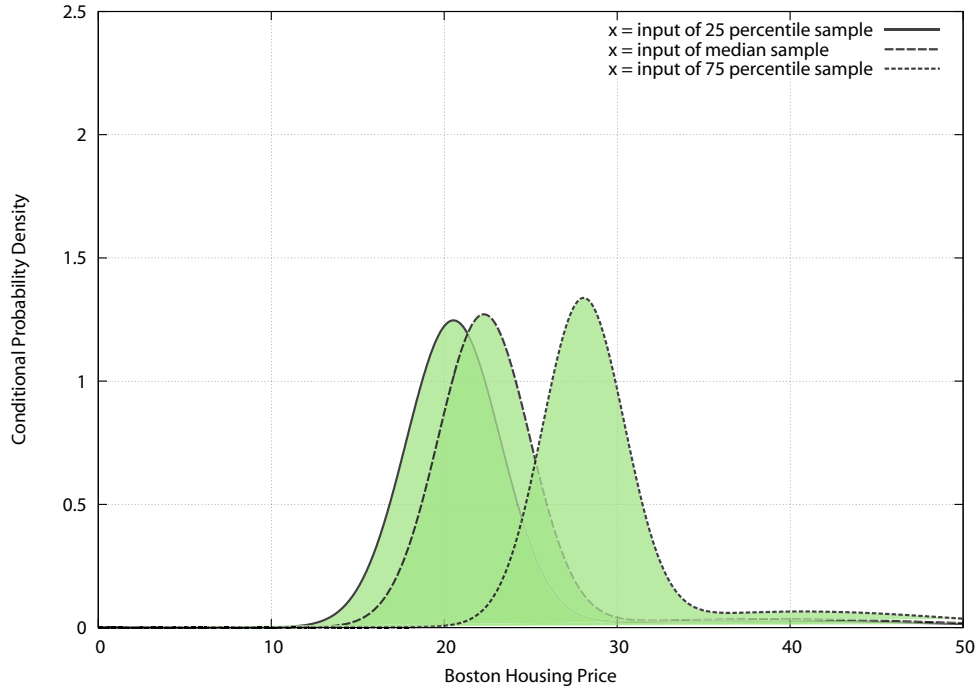


(b) MDN(-CV)

Figure 8: Conditional density estimates in Geiser data set.



(a) KQR-path



(b) MDN(-CV)

Figure 9: Conditional density estimates in Boston housing data set.

Acknowledgments

The authors thank Masakazu Kojima and Akiko Takeda for helpful discussions. IT was partially supported by Japanese Grants-in Aid for Scientific Research 16700258 and 19700261.

References

A. Azzalini and A. W. Bowman. A look at some data on the old fithful geyer. *Applied Statistics*, 39: 357–365, 1990.

F. R. Bach, D. Heckerman, and E. Horvits. Considering cost asymmetry in learning classifiers. *Journal of Machine Learning Research*, 7:1713–41, 2006.

C. Bishop. Mixture density networks. Technical Report NCRG/4288, Aston University, 1994.

G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems*, volume 13, pages 409–415, 2001.

H. Dette, N. Neumeyer, and K. F. Pilz. A simple nonparametric estimator of a strictly monotone regressin function. *Bernoulli*, 12(3):469–490, 2006.

J. Fan, Q. Yao, and H. Tong. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83:189–206, 1996.

P. Hall, R.C.L.Wolff, and Q. Yao. Methods for estimating a conditional distribution function. *Journal of the American Statistical Association*, 94:154–63, 1999.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–415, 2004.

X. He. Quantile curves without crossing. *The American Statistician*, 51:186–92, 1997.

R.J. Hyndman, D.M. Bashtannyk, and G.K. Grunwald. Estimating and visualizing conditional densities. *Journal of Computational and Graphical Statistics*, 5:315–36, 1996.

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

P. Laskov, C. Gehl, S. Kruger, and K. R. Muller. Incremental support vector learning: analysis, implementation and applications. *Journal of Machine Learning Research*, 7:1909–1936, 2006.

Y. Li, Y. Liu, and J. Zhu. Quantile regression in reproducing kernel hilbert spaces. *Journal of the American Statistical Association*, 102(477):255–68, 2007.

E. Mammen, J. S. Marron, B. A. Turlach, and M. P. Wand. A gerenal projection framework for constrained smoothing. *Statistical Science*, 16(3):232–248, 2001.

J. S. Marron and M. P. Wand. Exact mean integrated squared error. *The Annals of Statistics*, 20: 712–736, 1992.

A. D. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 55–60, 1994.

D. Pollard. *Convergence of stochastic processes*. Springer-Verlag, New York, 1984.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

J. O. Ramsay. Estimating smooth monotone functions. *Journal of The Royal Statistical Society B*, 60: 365–375, 1998.

E. F. Schuster and C. G. Gregory. On the nonconsistency of maximum likelihood nonparametric density estimators. In *Proceedings of the 13th Symposium on the Inference*, pages 295–298, 1981.

B. W. Silverman. *Density Estimation*. Chapman & Hall, 1986.

I. Takeuchi, Q. V. Le, T. Sears, and A. J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.

M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, 1994.

P. M. Williams. Using neural networks to model conditional multivariate densities. *Neural Computation*, 8:834–54, 1996.

T. Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research*, 2:527–550, 2002.

T. Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004.