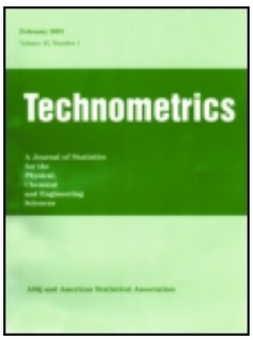

Technometrics

ISSN: 0040-1706 (Print) 1537-2723 (Online) Journal homepage: http://www.tandfonline.com/loi/utch20

Flexible Expectile Regression in Reproducing Kernel Hilbert Spaces

Yi Yang, Teng Zhang & Hui Zou

To cite this article: Yi Yang, Teng Zhang & Hui Zou (2017): Flexible Expectile Regression in Reproducing Kernel Hilbert Spaces, Technometrics, DOI: 10.1080/00401706.2017.1291450

To link to this article: http://dx.doi.org/10.1080/00401706.2017.1291450

TECHNOMETRICS
2017, VOL. 0, NO. 0, 1–10
https://doi.org/10.1080/00401706.2017.1291450

Taylor & Francis
Taylor & Francis Group

Flexible Expectile Regression in Reproducing Kernel Hilbert Spaces

Yi Yang[a], Teng Zhang[b], and Hui Zou[c]

[a]Department of Mathematics and Statistics, McGill University, Montréal, QC, Canada; [b]Department of Mathematics, University of Central Florida, Orlando, FL; [c]School of Statistics, University of Minnesota, Minneapolis, MN

ABSTRACT

Expectile, first introduced by Newey and Powell in 1987 in the econometrics literature, has recently become increasingly popular in risk management and capital allocation for financial institutions due to its desirable properties such as coherence and elicitability. The current standard tool for expectile regression analysis is the multiple linear expectile regression proposed by Newey and Powell in 1987. The growing applications of expectile regression motivate us to develop a much more flexible nonparametric multiple expectile regression in a reproducing kernel Hilbert space. The resulting estimator is called KERE, which has multiple advantages over the classical multiple linear expectile regression by incorporating nonlinearity, nonadditivity, and complex interactions in the final estimator. The kernel learning theory of KERE is established. We develop an efficient algorithm inspired by majorization-minimization principle for solving the entire solution path of KERE. It is shown that the algorithm converges at least at a linear rate. Extensive simulations are conducted to show the very competitive finite sample performance of KERE. We further demonstrate the application of KERE by using personal computer price data. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received October 2015
Revised January 2017

KEYWORDS
Asymmetry least squares; Expectile regression; MM principle; Reproducing kernel Hilbert space

1. Introduction

The *expectile* introduced by Newey and Powell (1987) is becoming an increasingly popular tool in risk management and capital allocation for financial institutions. Let Y be a random variable, the ω-expectile of Y, denoted as f_ω, is defined by

$$\omega = \frac{E\{|Y - f_\omega| I_{Y \le f_\omega}\}}{E\{|Y - f_\omega|\}}, \qquad \omega \in (0, 1). \tag{1}$$

In financial applications, the expectile has been widely used as a tool for efficient estimation of the expected shortfall (ES) through a one-one mapping between the two (Taylor 2008; Hamidi, Maillet, and Prigent 2014; Xie, Zhou, and Wan 2014). More recently, many researchers started to advocate the use of the expectile as a favorable alternative to other two commonly used risk measures—Value at Risk (VaR) and ES, due to its desirable properties such as *coherence* and *elicitability* (Kuan, Yeh, and Hsu 2009; Gneiting 2011; Ziegel 2016). VaR has been criticized mainly for two drawbacks: first, it does not reflect the magnitude of the extreme losses for the underlying risk as it is only determined by the probability of such losses; second, VaR is not a coherent risk measure due to the lack of the *subadditive* property (Emmer, Kratz, and Tasche 2013; Embrechts et al. 2014). Hence, the risk of merging portfolios together could get worse than adding the risks separately, which contradicts the notion that risk can be reduced by diversification (Artzner et al. 1999). Unlike VaR, ES is coherent and it considers the magnitude of the losses when the VaR is exceeded. However, a major problem with ES is that it cannot be reliably backtested in the sense that competing forecasts of ES cannot be properly evaluated through comparison with realized observations. Gneiting (2011) attributed this weakness to the fact that ES does not have *elicitability*. Ziegel (2016) further showed that the expectile are the only risk measure that is both coherent and elicitable.

In applications, we often need to estimate the conditional expectile of the response variable given a set of covariates. This is called expectile regression. Statisticians and Econometricians pioneered the study of expectile regression. Theoretical properties of the multiple linear expectile were firstly studied in Newey and Powell (1987) and Efron (1991). Yao and Tong (1996) studied a nonparametric estimator of conditional expectiles based on local linear polynomials with a one-dimensional covariate, and established the asymptotic property of the estimator. A semiparametric expectile regression model relying on penalized splines is proposed by Sobotka and Kneib (2012). Yang and Zou (2015) adopted the gradient tree boosting algorithm for expectile regression.

In this article, we propose a flexible nonparametric expectile regression estimator constructed in a reproducing kernel Hilbert space (RKHS) (Wahba 1990). Our contributions in this article are twofold: first, we extend the parametric expectile model to a fully nonparametric multiple regression setting and develop the corresponding kernel learning theory. Second, we propose an efficient algorithm that adopts the majorization-minimization principle for computing the entire solution path of the kernel expectile regression. We provide numerical convergence analysis for the algorithm. Moreover, we provide an accompanying R package that allows other researchers and practitioners to use the kernel expectile regression.

The rest of the article is organized as follows. In Section 2, we present the kernel expectile regression and develop an

CONTACT Hui Zou zouxx019@umn.edu
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/TECH.

asymptotic learning theory. Section 3 derives the fast algorithm for solving the solution paths of the kernel expectile regression. The numerical convergence of the algorithm is examined. In Section 4, we use simulation models to show the high prediction accuracy of the kernel expectile regression. We analyze the personal computer price data in Section 5. The technical proofs are relegated to an appendix.

2. Kernel Expectile Regression

2.1 Methodology

Newey and Powell (1987) showed that the ω-expectile f_ω of Y has an equivalent definition given by

$$f_\omega = \arg\min_f E\{\phi_\omega(Y - f)\}, \tag{2}$$

where

$$\phi_\omega(t) = \begin{cases} (1-\omega)t^2 & t \le 0, \\ \omega t^2 & t > 0. \end{cases} \tag{3}$$

Consequently, Newey and Powell (1987) showed that the ω-expectile f_ω of Y given the set of covariates $X = \mathbf{x}$, denoted by $f_\omega(\mathbf{x})$, can be defined as

$$f_\omega(\mathbf{x}) = \arg\min_f E\{\phi_\omega(Y - f) \mid X = \mathbf{x}\}. \tag{4}$$

Newey and Powell (1987) developed the multiple linear expectile regression based on (4). Given n random observations $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, Newey and Powell (1987) proposed the following formulation:

$$(\hat{\boldsymbol{\beta}}, \hat{\beta}_0) = \arg\min_{(\boldsymbol{\beta}, \beta_0)} \frac{1}{n} \sum_{i=1}^n \phi_\omega\left(y_i - \mathbf{x}_i^\intercal \boldsymbol{\beta} - \beta_0\right). \tag{5}$$

Then the estimated conditional ω-expectile is $\mathbf{x}_i^\intercal \hat{\boldsymbol{\beta}} + \hat{\beta}_0$. Efron (1991) proposed an efficient algorithm for computing (5).

The linear expectile estimator can be too restrictive in many real applications. Researchers have also considered more flexible expectile regression estimators. For example, Yao and Tong (1996) studied a local linear-polynomial expectile estimator with a one-dimensional covariate. However, the local fitting approach is not suitable when the dimension of explanatory variables is more than five. This limitation of local smoothing motivated Yang and Zou (2015) to develop a nonparametric expectile regression estimator based on the gradient tree boosting algorithm. The tree-boosted expectile regression tries to minimize the empirical expectile loss:

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \phi_\omega(y_i - f(\mathbf{x}_i)), \tag{6}$$

where each candidate function $f \in \mathcal{F}$ is assumed to be an ensemble of regression trees.

In this article, we consider another nonparametric approach to the multiple expectile regression. To motivate our method, let us first look at the special expectile regression with $\omega = 0.5$. It is easy to see from (3) and (4) that if $\omega = 0.5$, expectile regression actually reduces to ordinary conditional mean regression. A host of flexible regression methods has been well-studied for the conditional mean regression, such as generalized additive model, regression trees, boosted regression trees, and function estimation in a reproducing kernel Hilbert space (RKHS). Hastie, Tibshirani, and Friedman (2009) provided excellent introductions to all these methods. In particular, mean regression in an RKHS has a long history and a rich success record (Wahba 1990). So in the present work we propose the kernel expectile regression in an RKHS.

Denote by $\mathbb{H}_K$ the Hilbert space generated by a positive definite kernel K. By the Mercer's theorem, kernel K has an eigen-expansion $K(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^\infty \nu_i \varphi_i(\mathbf{x}) \varphi_i(\mathbf{x}')$ with $\nu_i \ge 0$ and $\sum_{i=1}^\infty \nu_i^2 < \infty$. The function f in $\mathbb{H}_K$ can be expressed as an expansion of these eigen-functions $f(\mathbf{x}) = \sum_{i=1}^\infty c_i \varphi_i(\mathbf{x})$ with the kernel induced squared norm $\|f\|_{\mathbb{H}_K}^2 \equiv \sum_{i=1}^\infty c_i^2/\nu_i < \infty$. Some most widely used kernel functions are

- Gaussian RBF kernel $K(\mathbf{x}, \mathbf{x}') = \exp(\frac{-\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma^2})$;
- sigmoidal kernel $K(\mathbf{x}, \mathbf{x}') = \tanh(\kappa\langle \mathbf{x}, \mathbf{x}'\rangle + \theta)$;
- polynomial kernel $K(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}'\rangle + \theta)^d$.

Other kernels can be found in Smola, Schölkopf, and Müller (1998) and Hastie, Tibshirani, and Friedman (2009).

Given n observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the kernel expectile regression estimator (KERE) is defined as

$$(\hat{f}_n(\mathbf{x}), \hat{\alpha}_0) = \arg\min_{f \in \mathbb{H}_K, \alpha_0 \in \mathbb{R}} \sum_{i=1}^n \phi_\omega(y_i - \alpha_0 - f(\mathbf{x}_i)) + \lambda \|f\|_{\mathbb{H}_K}^2, \tag{7}$$

where $\mathbf{x}_i \in \mathbb{R}^p$, $\alpha_0 \in \mathbb{R}$. The estimated conditional ω-expectile is $\hat{\alpha}_0 + \hat{f}_n(\mathbf{x})$. Sometimes, one can absorb the intercept term into the nonparametric function f. We keep the intercept term to make a direct comparison to the multiple linear expectile regression.

Although (7) is often an optimization problem in an infinite-dimensional space, depending on the choice of the kernel, the representer theorem (Wahba 1990) ensures that the solution to (7) always lies in a finite-dimensional subspace spanned by kernel functions on observational data, that is,

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{x}), \tag{8}$$

for some $\{\alpha_i\}_{i=1}^n \subset \mathbb{R}$.

By (8) and the reproducing property of RKHS (Wahba 1990), we have

$$\|f\|_{\mathbb{H}_K}^2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{9}$$

Based on (8) and (9), we can rewrite the minimization problem (7) in a finite-dimensional space

$$\{\hat{\alpha}_i\}_{i=0}^n = \arg\min_{\{\alpha_i\}_{i=0}^n} \sum_{i=1}^n \phi_\omega\left(y_i - \alpha_0 - \sum_{j=1}^n \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)\right) + \lambda \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j). \tag{10}$$

The corresponding KERE estimator is $\hat{\alpha}_0 + \sum_{i=1}^n \hat{\alpha}_i K(\mathbf{x}_i, \mathbf{x})$.

The computation of KERE is based on (10) and we use both (7) and (10) for the theoretical analysis of KERE.

2.2. Kernel Learning Theory

In this section, we develop a kernel learning theory for KERE. We first discuss the criterion for evaluating an estimator in the context of expectile regression. Given the loss function ϕ_ω, the risk is $\mathcal{R}(f, \alpha_0) = E_{(\mathbf{x},y)}\phi_\omega(y - \alpha_0 - f(\mathbf{x}))$. It is argued that $\mathcal{R}(f, \alpha_0)$ is a more appropriate evaluation measure in practice than the squared error risk defined as $E_{\mathbf{x}}\|f(\mathbf{x}) + \alpha_0 - f^*_\omega(\mathbf{x})\|^2$, where f^*_ω is the true conditional expectile of Y given $X = \mathbf{x}$. The reason is simple: Let $\hat{f}, \hat{\alpha}_0$ be any estimator based on the training data. By law of large number, we see that

$$\begin{aligned}\mathcal{R}(\hat{f}, \hat{\alpha}_0) &= E_{\{y_j,\mathbf{x}_j\}_{j=1}^m} \frac{1}{m}\sum_{j=1}^m \phi_\omega(y_j - \hat{\alpha}_0 - \hat{f}(\mathbf{x}_j)) \\ &= \lim_{m\to\infty} \frac{1}{m}\sum_{j=1}^m \phi_\omega(y_j - \hat{\alpha}_0 - \hat{f}(\mathbf{x}_j)),\end{aligned}$$

where $\{(\mathbf{x}_j, y_j)\}_{j=1}^m$ is another independent test sample. Thus, one can use techniques such as cross-validation to estimate $\mathcal{R}(f, \alpha_0)$. Additionally, the squared error risk depends on the function $f^*_\omega(\mathbf{x})$, which is usually unknown. Thus, we prefer to use $\mathcal{R}(\hat{f}, \hat{\alpha}_0)$ over the squared error risk. Of course, if we assume a classical regression model (when $\omega = 0.5$) such as $y = f(\mathbf{x}) +$ error, where the error is independent of $\mathbf{x}$ with mean zero and constant variance, $\mathcal{R}(\hat{f}, \hat{\alpha}_0)$ then just equals the squared error risk plus a constant. Unfortunately, such equivalence breaks down for other values of ω and more general models.

After choosing the risk function, the goal is to minimize the risk. Since typically the estimation is done in a function space, the minimization is carried out in the chosen function space. In our case, the function space is RKHS generated by a kernel function K. Thus, the ideal risk is defined as

$$\mathcal{R}^*_{f,\alpha_0} = \inf_{f\in\mathbb{H}_K,\alpha_0\in\mathbb{R}} \mathcal{R}(f, \alpha_0).$$

Consider the kernel expectile regression estimator $(\hat{f}, \hat{\alpha}_0)$ as defined in (7) based on a training sample D_n, where $D_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ are iid drawn from an unknown distribution. The observed risk of KERE is

$$\mathcal{R}(\hat{f}, \hat{\alpha}_0) = E_{(\mathbf{x},y)}\phi_\omega(y - \hat{\alpha}_0 - \hat{f}(\mathbf{x})).$$

It is desirable to show that $\mathcal{R}(\hat{f}, \hat{\alpha}_0)$ approaches the ideal risk $\mathcal{R}^*_{f,\alpha_0}$.

It is important to note that $\mathcal{R}(\hat{f}, \hat{\alpha}_0)$ is a random quantity that depends on the training sample D_n. So it is not the usual risk function, which is deterministic. However, we can consider the expectation of $\mathcal{R}(\hat{f}, \hat{\alpha}_0)$ and call it *expected observed risk*. The formal definition is

Expected observed risk:

$$E_{D_n}\mathcal{R}(\hat{f}, \hat{\alpha}_0) = E_{D_n}\left\{E_{(\mathbf{x},y)}\phi_\omega(y - \hat{\alpha}_0 - \hat{f}(\mathbf{x}))\right\}. \quad (11)$$

Our goal is to show that $\mathcal{R}(\hat{f}, \hat{\alpha}_0)$ converges to $\mathcal{R}^*_{f,\alpha_0}$. We achieve this by showing that the expected observed risk converges to the ideal risk, that is, $\lim_{n\to\infty} E_{D_n}\mathcal{R}(\hat{f}, \hat{\alpha}_0) = \mathcal{R}^*_{f,\alpha_0}$. By definition, we always have $\mathcal{R}(\hat{f}, \hat{\alpha}_0) \geq \mathcal{R}^*_{f,\alpha_0}$. Then by Markov inequality, for any $\varepsilon > 0$

$$P\left(\mathcal{R}(\hat{f}, \hat{\alpha}_0) - \mathcal{R}^*_{f,\alpha_0} > \varepsilon\right) \leq \frac{E_{D_n}\mathcal{R}(\hat{f}, \hat{\alpha}_0) - \mathcal{R}^*_{f,\alpha_0}}{\varepsilon} \to 0.$$

The rigorous statement of our result is as follows:

Theorem 1. Let $M = \sup_{\mathbf{x}} K(\mathbf{x}, \mathbf{x})^{1/2}$. Assume $M < \infty$ and $Ey^2 < D < \infty$ where M and D are two constants. If λ is chosen such that as $n \to \infty$, $\lambda/n^{2/3} \to \infty$, $\lambda/n \to 0$, then we have

$$E_{D_n}\mathcal{R}(\hat{f}, \hat{\alpha}_0) \to \mathcal{R}^*_{f,\alpha_0} \quad \text{as } n \to \infty,$$

and hence

$$\mathcal{R}(\hat{f}, \hat{\alpha}_0) - \mathcal{R}^*_{f,\alpha_0} \to 0 \text{ in probability.}$$

The Gaussian kernel is perhaps the most popular kernel for nonlinear learning. For the Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2/c)$, we have $M = 1$. For any radial kernel with the form $K(\mathbf{x}, \mathbf{x}') = h(\|\mathbf{x} - \mathbf{x}'\|)$ where h is a smooth decreasing function, we see $M = h(0)^{\frac{1}{2}}$, which is finite as long as $h(0) < \infty$.

3. Algorithm

3.1. Derivation

Majorization-minimization (MM) algorithm is a very successful technique for solving a wide range of statistical models (Lange, Hunter, and Yang 2000; Hunter and Lange 2004; Wu and Lange 2010; Zhou and Lange 2010; Lange and Zhou 2014). In this section, we develop an algorithm inspired by MM principle for solving the optimization problem (10). Note that the loss function ϕ_ω in (10) does not have the second derivative. We adopt the MM principle to find the minimizer by iteratively minimizing a surrogate function that majorizes the objective function in (10).

To further simplify the notation, we write $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n)^\intercal$, and

$$\mathbf{K}_i = (1, K(\mathbf{x}_i, \mathbf{x}_1), \ldots, K(\mathbf{x}_i, \mathbf{x}_n)),$$

$$\mathbf{K} = \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix},$$

$$\mathbf{K}_0 = \begin{pmatrix} 0 & \mathbf{0}^\intercal \\ \mathbf{0} & \mathbf{K} \end{pmatrix}.$$

Then (10) is simplified to a minimization problem as

$$\widehat{\boldsymbol{\alpha}} = \operatorname*{argmin}_{\boldsymbol{\alpha}} F_{\omega,\lambda}(\boldsymbol{\alpha}), \quad (12)$$

$$F_{\omega,\lambda}(\boldsymbol{\alpha}) = \sum_{i=1}^n \phi_\omega\left(y_i - \mathbf{K}_i\boldsymbol{\alpha}\right) + \lambda\boldsymbol{\alpha}^\intercal\mathbf{K}_0\boldsymbol{\alpha}, \quad (13)$$

where ω is given for computing the corresponding level of the conditional expectile. We also assume that λ is given for the time being. A smart algorithm for computing the solution for a sequence of λ will be studied in Section 3.3.

Our approach is to minimize (12) by iteratively update $\boldsymbol{\alpha}$ using the minimizer of a majorization function of

$F_{\omega,\lambda}(\boldsymbol{\alpha})$. Specifically, at the kth step of the algorithm, where $k = 0, 1, 2, \ldots$, assume that $\boldsymbol{\alpha}^{(k)}$ is the current value of $\boldsymbol{\alpha}$ at iteration k, we find a majorization function $Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)})$ for $F_{\omega,\lambda}(\boldsymbol{\alpha})$ at current $\boldsymbol{\alpha}^{(k)}$ that satisfies

$$Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}) > F_{\omega,\lambda}(\boldsymbol{\alpha}) \quad \text{when} \quad \boldsymbol{\alpha} \neq \boldsymbol{\alpha}^{(k)}, \tag{14}$$

$$Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}) = F_{\omega,\lambda}(\boldsymbol{\alpha}) \quad \text{when} \quad \boldsymbol{\alpha} = \boldsymbol{\alpha}^{(k)}. \tag{15}$$

Then we update $\boldsymbol{\alpha}$ by minimizing $Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)})$ rather than the actual objective function $F_{\omega,\lambda}(\boldsymbol{\alpha})$:

$$\boldsymbol{\alpha}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}). \tag{16}$$

To construct the majorization function $Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)})$ for $F_{\omega,\lambda}(\boldsymbol{\alpha})$ at the kth iteration, we use the following lemma:

Lemma 1. The expectile loss ϕ_ω has a Lipschitz continuous derivative ϕ'_ω, that is,

$$|\phi'_\omega(a) - \phi'_\omega(b)| \le L|a - b| \qquad \forall a, b \in \mathbb{R}, \tag{17}$$

where $L = 2\max(1 - \omega, \omega)$. This further implies that ϕ_ω has a quadratic upper bound

$$\phi_\omega(a) \le \phi_\omega(b) + \phi'_\omega(b)(a - b) + \frac{L}{2}|a - b|^2 \qquad \forall a, b \in \mathbb{R}. \tag{18}$$

Note that "=" is taken only when $a = b$.

Assume the current "residual" is $r_i^{(k)} = y_i - \mathbf{K}_i\boldsymbol{\alpha}^{(k)}$, then it is equivalent in (12) that $y_i - \mathbf{K}_i\boldsymbol{\alpha} = r_i^{(k)} - \mathbf{K}_i(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)})$. By Lemma 1, we obtain

$$\left|\phi'_\omega\left(r_i^{(k)} - \mathbf{K}_i(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)})\right) - \phi'_\omega(r_i^{(k)})\right| \le 2\max(1 - \omega, \omega)\, |\mathbf{K}_i(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)})|,$$

and the quadratic upper bound

$$\phi_\omega\left(r_i^{(k)} - \mathbf{K}_i\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)}\right)\right) \le q_i(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}),$$

where

$$q_i(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}) = \phi_\omega(r_i^{(k)}) - \phi'_\omega(r_i^{(k)})\mathbf{K}_i(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)}) + \max(1 - \omega, \omega)(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)})^\intercal \mathbf{K}_i\mathbf{K}_i^\intercal(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)}).$$

Therefore, the majorization function of $F_{\omega,\lambda}(\boldsymbol{\alpha})$ can be written as

$$Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}) = \sum_{i=1}^{n} q_i(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}) + \lambda\boldsymbol{\alpha}^\intercal\mathbf{K}_0\boldsymbol{\alpha}, \tag{19}$$

which has an alternatively form that can be written as

$$Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}) = F_{\omega,\lambda}(\boldsymbol{\alpha}^{(k)}) + \nabla F_{\omega,\lambda}(\boldsymbol{\alpha}^{(k)})(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)}) + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)})^\intercal\mathbf{K}_u(\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)}), \tag{20}$$

where

$$\mathbf{K}_u = \lambda\mathbf{K}_0 + \max(1 - \omega, \omega)\sum_{i=1}^{n}\mathbf{K}_i\mathbf{K}_i^\intercal \tag{21}$$

$$= \max(1 - \omega, \omega)\begin{pmatrix} n & \mathbf{1}^\intercal\mathbf{K} \\ \mathbf{K}\mathbf{1} & \mathbf{K}\mathbf{K} + \frac{\lambda}{\max(1-\omega,\omega)}\mathbf{K} \end{pmatrix}, \tag{22}$$

and $\mathbf{1}$ is an $n \times 1$ vector of all ones. Our algorithm updates $\boldsymbol{\alpha}$ using the minimizer of the quadratic majorization function (20):

$$\boldsymbol{\alpha}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha} \mid \boldsymbol{\alpha}^{(k)}) = \boldsymbol{\alpha}^{(k)} + \mathbf{K}_u^{-1}\Big(-\lambda\mathbf{K}_0\boldsymbol{\alpha}^{(k)} + \frac{1}{2}\sum_{i=1}^{n}\phi'_\omega(r_i^{(k)})\mathbf{K}_i\Big). \tag{23}$$

The details of the whole procedures for solving (12) are described in Algorithm 1.

Algorithm 1 The algorithm for the minimization of (12).

- Let $\{y_i\}_1^n$ be observations of the response, $\{K(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^n$ be the kernel of all observations, and $\boldsymbol{\alpha} := (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_n)$.
- Initialize $\boldsymbol{\alpha}^{(0)}$ and $k = 0$.
- Iterate Steps 1–3 until convergence:

1. Calculated the residue of the response by $r_i^{(k)} = y_i - \mathbf{K}_i\boldsymbol{\alpha}^{(k)}$ for all $1 \le i \le n$.
2. Obtain $\boldsymbol{\alpha}^{(k+1)}$ by:

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + \mathbf{K}_u^{-1}\left(-\lambda\mathbf{K}_0\boldsymbol{\alpha}^{(k)} + \frac{1}{2}\sum_{i=1}^{n}\phi'_\omega(r_i^{(k)})\mathbf{K}_i\right),$$

where

$$\mathbf{K}_u = \max(1 - \omega, \omega)\begin{pmatrix} n & \mathbf{1}^\intercal\mathbf{K} \\ \mathbf{K}\mathbf{1} & \mathbf{K}\mathbf{K} + \frac{\lambda}{\max(1-\omega,\omega)}\mathbf{K} \end{pmatrix}.$$

3. $k := k + 1$.

3.2. Convergence Analysis

Now we provide the convergence analysis of Algorithm 1. Lemma 2 shows that the sequence $(\boldsymbol{\alpha}^{(k)})$ in the algorithm converges to the unique global minimum $\widehat{\boldsymbol{\alpha}}$ of the optimization problem.

Lemma 2. If we update $\boldsymbol{\alpha}^{(k+1)}$ by using (23), then the following results hold:

1. The descent property of the objective function. $F_{\omega,\lambda}(\boldsymbol{\alpha}^{(k+1)}) \le F_{\omega,\lambda}(\boldsymbol{\alpha}^{(k)})$, $\forall k$.
2. The convergence of $\boldsymbol{\alpha}$. Assume that $\sum_{i=1}^n \mathbf{K}_i\mathbf{K}_i^\intercal$ is a positive definite matrix, then $\lim_{k\to\infty}\|\boldsymbol{\alpha}^{(k+1)} - \boldsymbol{\alpha}^{(k)}\| = 0$.
3. The sequence $(\boldsymbol{\alpha}^{(k)})$ converges to $\widehat{\boldsymbol{\alpha}}$, which is the unique global minimum of (12).

Theorem 2. Denote by $\widehat{\boldsymbol{\alpha}}$ the unique minimizer of (12) and

$$\Lambda_k = \frac{Q(\widehat{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha}^{(k)}) - F_{\omega,\lambda}(\widehat{\boldsymbol{\alpha}})}{(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^{(k)})^\intercal\mathbf{K}_u(\widehat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^{(k)})}. \tag{24}$$

Note that when $\Lambda_k = 0$, it is just a trivial case $\boldsymbol{\alpha}^{(j)} = \widehat{\boldsymbol{\alpha}}$ for $j > k$. We define

$$\Gamma = 1 - \gamma_{\min}\left(\mathbf{K}_u^{-1}\mathbf{K}_l\right),$$

where

$$\mathbf{K}_l = \lambda\mathbf{K}_0 + \min(1 - \omega, \omega)\sum_{i=1}^{n}\mathbf{K}_i\mathbf{K}_i^\intercal.$$

Assume that $\sum_{i=1}^{n} \mathbf{K}_i \mathbf{K}_i^\intercal$ is a positive definite matrix. Then we have the following results:

1. $F_{\omega,\lambda}(\boldsymbol{\alpha}^{(k+1)}) - F_{\omega,\lambda}(\widehat{\boldsymbol{\alpha}}) \le \Lambda_k (F_{\omega,\lambda}(\boldsymbol{\alpha}^{(k)}) - F_{\omega,\lambda}(\widehat{\boldsymbol{\alpha}}))$.
2. The sequence $(F_{\omega,\lambda}(\boldsymbol{\alpha}^{(k)}))$ has a linear convergence rate no greater than Γ, and $0 \le \Lambda_k \le \Gamma < 1$.
3. The sequence $(\boldsymbol{\alpha}^{(k)})$ has a linear convergence rate no greater than $\sqrt{\Gamma \gamma_{\max}(\mathbf{K}_u)/\gamma_{\min}(\mathbf{K}_l)}$, that is,

$$\|\boldsymbol{\alpha}^{(k+1)} - \widehat{\boldsymbol{\alpha}}\| \le \sqrt{\Gamma \frac{\gamma_{\max}(\mathbf{K}_u)}{\gamma_{\min}(\mathbf{K}_l)}} \|\boldsymbol{\alpha}^{(k)} - \widehat{\boldsymbol{\alpha}}\|.$$

Theorem 2 says that the convergence rate of Algorithm 1 is at least linear. In our numeric experiments, we have found that Algorithm 1 converges very fast: the convergence criterion is usually met after 15 iterations.

3.3. Implementation

We discuss some techniques used in our implementation to further improve the computational speed of the algorithm.

Usually expectile models are computed by applying Algorithm 1 on a descending sequence of λ values. To create a sequence $\{\lambda_m\}_{m=1}^{M}$, we place $M-2$ points uniformly (in the log-scale) between the starting and ending point $\lambda_{\max}$ and $\lambda_{\min}$ such that the λ sequence length is M. The default number for M is 100, hence $\lambda_1 = \lambda_{\max}$, and $\lambda_{100} = \lambda_{\min}$. We adopt the warm-start trick to implement the solution paths along λ values: suppose that we have already obtained the solution $\widehat{\boldsymbol{\alpha}}_{\lambda_m}$ at λ_m, then $\widehat{\boldsymbol{\alpha}}_{\lambda_m}$ will be used as the initial value for computing the solution at λ_{m+1} in Algorithm 1.

Another computational trick adopted is based on the fact that in Algorithm 1, the inverse of $\mathbf{K}_u$ does not have to be recalculated for each λ. There is an easy way to update $\mathbf{K}_u^{-1}$ for $\lambda_1, \lambda_2, \ldots$. Because $\mathbf{K}_u$ can be partitioned into two rows and two columns of submatrices, by Theorem 8.5.11 of Harville (2008), $\mathbf{K}_u^{-1}$ can be expressed as

$$\begin{aligned}\mathbf{K}_u^{-1}(\lambda) &= \frac{1}{\max(1-\omega,\omega)} \begin{pmatrix} n & \mathbf{1}^\intercal \mathbf{K} \\ \mathbf{K}\mathbf{1} & \mathbf{K}\mathbf{K} + \frac{\lambda}{\max(1-\omega,\omega)}\mathbf{K} \end{pmatrix}^{-1} \\ &= \frac{1}{\max(1-\omega,\omega)} \left[\begin{pmatrix} \frac{1}{n} & \mathbf{0}_{1\times n} \\ \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times n} \end{pmatrix} \right. \\ &\quad \left. + \begin{pmatrix} -\frac{1}{n}\mathbf{1}^\intercal \mathbf{K} \\ \mathbf{I}_n \end{pmatrix} \mathbf{Q}_\lambda^{-1} \left(-\frac{1}{n}\mathbf{K}\mathbf{1}, \mathbf{I}_n \right) \right], \end{aligned} \tag{25}$$

where

$$\mathbf{Q}_\lambda^{-1} = \left[\left(\mathbf{K}\mathbf{K} + \frac{\lambda}{\max(1-\omega,\omega)}\mathbf{K} \right) - \frac{1}{n}\mathbf{K}\mathbf{1}\mathbf{1}^\intercal\mathbf{K} \right]^{-1}.$$

In (25), only $\mathbf{Q}_\lambda^{-1}$ changes with λ, therefore the computation of $\mathbf{K}_u^{-1}$ for a different λ only requires the updating of $\mathbf{Q}_\lambda^{-1}$. Observing that $\mathbf{Q}_\lambda^{-1}$ is the inverse of the sum of two submatrices $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{A}_\lambda = \mathbf{K}\mathbf{K} + \frac{\lambda}{\max(1-\omega,\omega)}\mathbf{K}, \qquad \mathbf{B} = -\frac{1}{n}\mathbf{K}\mathbf{1}\mathbf{1}^\intercal\mathbf{K}.$$

By Sherman–Morrison formula (Sherman and Morrison 1950),

$$\mathbf{Q}_\lambda^{-1} = [\mathbf{A}_\lambda + \mathbf{B}]^{-1} = \mathbf{A}_\lambda^{-1} - \frac{1}{1+g}\mathbf{A}_\lambda^{-1}\mathbf{B}\mathbf{A}_\lambda^{-1}, \tag{26}$$

where $g = \operatorname{trace}(\mathbf{B}\mathbf{A}_\lambda^{-1})$, we find that to get $\mathbf{Q}_\lambda^{-1}$ for a different λ one just needs to get $\mathbf{A}_\lambda^{-1}$, which can be efficiently computed by using eigen-decomposition $\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{U}^\intercal$:

$$\begin{aligned}\mathbf{A}_\lambda^{-1} &= \left(\mathbf{K}\mathbf{K} + \frac{\lambda}{\max(1-\omega,\omega)}\mathbf{K} \right)^{-1} \\ &= \mathbf{U} \left(\mathbf{D}^2 + \frac{\lambda}{\max(1-\omega,\omega)}\mathbf{D} \right)^{-1} \mathbf{U}^\intercal. \end{aligned} \tag{27}$$

Equation (27) implies that the computation of $\mathbf{K}_u^{-1}(\lambda)$ depends only on λ, $\mathbf{D}$, $\mathbf{U}$, and ω. Since $\mathbf{D}$, $\mathbf{U}$, and ω stay unchanged, we only need to calculate them once. To get $\mathbf{K}_u^{-1}(\lambda)$ for a different λ in the sequence, we just need to plug in a new λ in (27).

The following is the implementation for computing KERE for a sequence of λ values using Algorithm 1:

- Calculate $\mathbf{U}$ and $\mathbf{D}$ according to $\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{U}^\intercal$.
- Initialize $\widehat{\boldsymbol{\alpha}}_{\lambda_0} = [0, 0, \ldots, 0]$.
- **for** $m = 1, 2, \ldots, M$, repeat steps 1–3:
  1. Initialize $\boldsymbol{\alpha}_{\lambda_m}^{(0)} = \widehat{\boldsymbol{\alpha}}_{\lambda_{m-1}}$.
  2. Compute $\mathbf{K}_u^{-1}(\lambda_m)$ using (25), (26), and (27).
  3. Call Algorithm 1 to compute $\widehat{\boldsymbol{\alpha}}_{\lambda_m}$.

Our algorithm has been implemented in an official R package `KERE`, which is publicly available from the Comprehensive R Archive Network at *http://cran.r-project.org/web/packages/KERE/index.html*.

4. Simulation

In this section, we conduct extensive simulations to show the excellent finite performance of KERE. We investigate how the performance of KERE is affected by various model and error distribution settings, training sample sizes, and other characteristics. Throughout this section, we consider Gaussian radial basis function (RBF) kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}$, Laplacian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma}}$, and the hyperbolic tangent kernel $K(\mathbf{x}, \mathbf{x}') = \tanh(\sigma\langle \mathbf{x}, \mathbf{x}'\rangle + 1)$ (Hastie, Tibshirani, and Friedman 2009). We select the best pair of kernel parameter σ and regularization parameter λ by two-dimensional five-fold cross-validation. All computations were done on an Intel Core i7-3770 processor at 3.40GHz.

Simulation I: Single Covariate Case

The model used for this simulation is defined as

$$y_i = \sin(0.7x_i) + \frac{x_i^2}{20} + \frac{|x_i| + 1}{5}\epsilon_i, \tag{28}$$

which is heteroscedastic as error depends on a single covariate $x \sim U[-8, 8]$. We used a single covariate such that the estimator can be visualized nicely.

We used two different error distributions: Laplace distribution and a mixed normal distribution,

$$\epsilon_i \sim 0.5N\left(0, \frac{1}{4}\right) + 0.5N\left(1, \frac{1}{16}\right).$$

We generated $n = 400$ training observations from (28), on which five expectile models with levels $\omega = \{0.05, 0.2, 0.5, 0.8, 0.95\}$ were fitted. We selected the best (σ, λ)

Table 1. The averaged MADs and the corresponding standard errors of expectile regression predictions for single covariate heteroscedastic models with mixed normal and Laplace error.

ω	0.05	0.2	0.5	0.8	0.95
			KERE (Gaussian)		
Mixture	0.236 (0.003)	0.138 (0.003)	0.376 (0.002)	0.610 (0.002)	0.788 (0.002)
Laplace	2.346 (0.013)	1.037 (0.007)	0.179 (0.005)	1.033 (0.006)	2.333 (0.027)
			KERE (Laplacian)		
Mixture	0.242 (0.003)	0.149 (0.003)	0.377 (0.002)	0.599 (0.002)	0.778 (0.002)
Laplace	2.348 (0.015)	1.036 (0.007)	0.210 (0.005)	1.019 (0.006)	2.301 (0.011)
			KERE (Hyperbolic tangent)		
Mixture	1.015 (0.003)	0.867 (0.001)	0.980 (0.002)	1.280 (0.004)	1.995 (0.008)
Laplace	2.574 (0.011)	1.404 (0.005)	0.880 (0.002)	1.541 (0.007)	3.015 (0.016)

NOTES: The models are fitted on five expectile levels $\omega = \{0.05, 0.2, 0.5, 0.8, 0.95\}$. The results are based on 300 independent runs.

pair by using two-dimensional, five-fold cross-validation. We generated an additional $n' = 2000$ test observations for evaluating the mean absolute deviation (MAD) of the final estimate. Assume that the true expectile function is f_ω and the predicted expectile is $\hat{f}_\omega$, then the mean absolute deviation is defined as

$$\mathrm{MAD}(\omega) = \frac{1}{n'}\sum_{i=1}^{n'} |f_\omega(\mathbf{x}_i) - \hat{f}_\omega(\mathbf{x}_i)|.$$

The true expectile f_ω is equal to $\sin(0.7x) + \frac{x^2}{20} + \frac{|x|+1}{5} b_\omega(\epsilon)$, where $b_\omega(\epsilon)$ is the ω-expectile of ϵ, which is the theoretical minimizer of $E\phi_\omega(\epsilon - b)$.

The simulations were repeated for 100 times under the above settings. We recorded MADs for different expectile levels in Table 1. We find that the accuracy of the expectile prediction with mixed normal errors is generally better than that with Laplace errors. For the symmetric Laplace case, the prediction MADs are also symmetric around $\omega = 0.5$, while for the skewed mixed-normal case the MADs are skewed. Overall, KERE with the Gaussian Kernel and the Laplacian kernel achieve better performance than KERE with the hyperbolic tangent kernel. To show that KERE works as expected, in Figure 1 we also compared the theoretical and predicted expectile curves based on KERE (Gaussian kernel) with $\omega = \{0.05, 0.2, 0.5, 0.8, 0.95\}$ in Figure 1. We can see that the corresponding theoretical and predicted curves are very close. Theoretically the two should become the same curves as $n \to \infty$.

Simulation II: Multiple Covariate Case

In this part, we illustrate that KERE can work very well for target functions that are nonadditive and/or with complex interactions. We generated data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ according to

$$y_i = f_1(\mathbf{x}_i) + |f_2(\mathbf{x}_i)|\epsilon_i,$$

where predictors $\mathbf{x}_i$, $i = 1, \ldots, n$ were generated from a joint normal distribution $N(0, \mathbf{I}_p)$ with $p = 10$. For the error term ϵ_i, we consider three types of distributions:

1. Normal distribution $\epsilon_i \sim N(0, 1)$.
2. Student's t-distribution with four degrees of freedom $\epsilon_i \sim t_4$.
3. Mixed normal distribution $\epsilon_i \sim 0.9N(0, 1) + 0.1N(1, 4)$.

We now describe the construction of f_1 and f_2. In the homoscedastic model, we let $f_2(\mathbf{x}_i) = 1$ and f_1 is generated by the "random function generator" model (Friedman 2000), according to

$$f(\mathbf{x}) = \sum_{l=1}^{20} a_l g_l(\mathbf{x}_l),$$

where $\{a_l\}_{l=1}^{20}$ are sampled from uniform distribution $a_l \sim U[-1, 1]$, and $\mathbf{x}_l$ is a random subset of p-dimensional predictor $\mathbf{x}$, with size $p_l = \min(\lfloor 1.5 + r, p \rfloor)$, where r was sampled from exponential distribution $r \sim \mathrm{Exp}(0.5)$. The function $g_l(\mathbf{x}_l)$ is a p_l-dimensional Gaussian function:

$$g_l(x_l) = \exp\Big[-\frac{1}{2}(\mathbf{x}_l - \boldsymbol{\mu}_l)^\intercal \mathbf{V}_l (\mathbf{x}_l - \boldsymbol{\mu}_l)\Big],$$

where $\boldsymbol{\mu}_l$ follows the distribution $N(0, \mathbf{I}_{p_l})$. The $p_l \times p_l$ covariance matrix $\mathbf{V}_l$ is defined by $\mathbf{V}_l = \mathbf{U}_l \mathbf{D}_l \mathbf{U}_l^\intercal$, where $\mathbf{U}_l$ is a random orthogonal matrix, and $\mathbf{D}_l = \mathrm{diag}(d_{1l}, d_{2l}, \ldots, d_{p_l l})$ with $\sqrt{d_{jl}} \sim U[0.1, 2]$.

In the heteroscedastic model, f_1 is the same as in the homoscedastic model and f_2 is independently generated by the "random function generator" model.

We generated $n = 300$ observations as the training set, on which the estimated expectile functions $\hat{f}_\omega$ were computed at seven levels:

$$\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}.$$

An additional test set with $n' = 1200$ observations was generated for evaluating MADs between the fitted expectile $\hat{f}_\omega$ and the true expectile f_ω. Note that the expectile function $f_\omega(\mathbf{x})$ is equal to $f_1(\mathbf{x}) + b_\omega(\epsilon)$ in the homoscedastic model and $f_1(\mathbf{x}) + |f_2(\mathbf{x})| b_\omega(\epsilon)$ in the heteroscedastic model, where $b_\omega(\epsilon)$ is the ω-expectile of the error distribution. Under the above settings, we repeated the simulations for 300 times and record the MAD and timing each time.

In Figures 2 and 3, we show the boxplots of empirical distributions of MADs, and in Tables 2–4, we report the average values of MADs and corresponding standard errors. We see that KERE can deliver accurate expectile prediction results in all cases, although relatively the prediction error is more volatile in the heteroscedastic case as expected: in the mean regression case ($\omega = 0.5$), the averaged MADs in homoscedastic and heteroscedastic models are very close. But this difference grows larger as ω moves away from 0.5. We also observe that

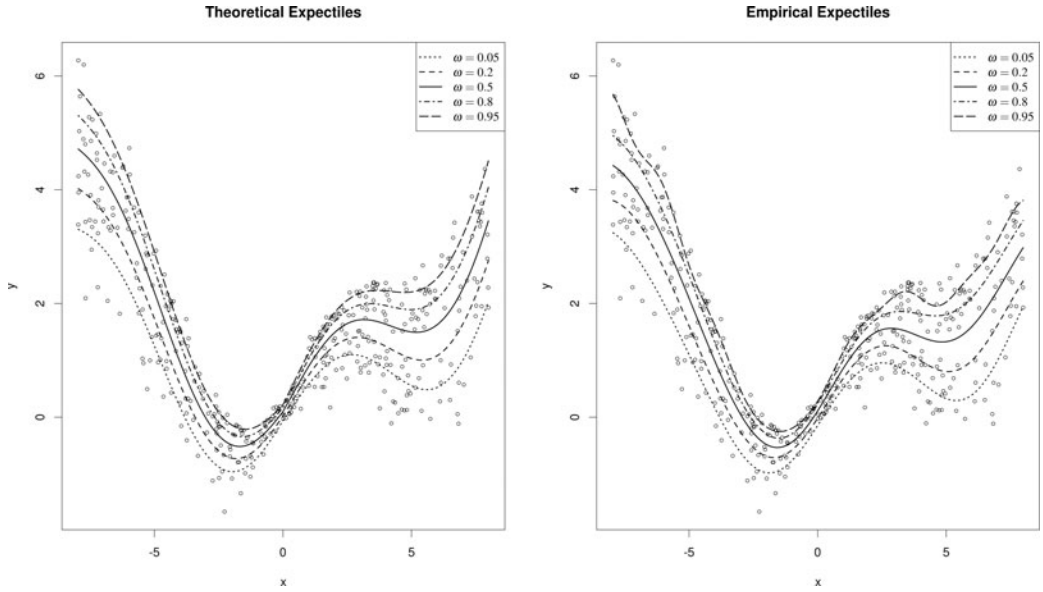

Figure 1. Theoretical expectiles and empirical expectiles (Gaussian kernel) for a covariate heteroscedastic model with mixed normal error. The model is fitted on five expectile levels $\omega = \{0.05, 0.2, 0.5, 0.8, 0.95\}$.

the prediction MADs for symmetric distributions, normal and t_4, also appear to be symmetric around the conditional mean $\omega = 0.5$, and that the prediction MADs in the skewed mixed-normal distribution cases are asymmetric. We can also see that the Gaussian kernel and the Laplacian kernel give better performance than the Hyperbolic tangent kernel. The total computation times for conducting two-dimensional, five-fold cross-validation and fitting the final model (Gaussian kernel) with the chosen parameters (σ, λ) for conditional expectiles are also reported in Table 5. We find that the algorithm can efficiently solve all models under 20 sec, regardless of choices of error distributions.

We next study how sample size affects predictive performance and computational time. We fit expectile models (KERE with the Gaussian kernel) with $\omega \in \{0.1, 0.5, 0.9\}$ using various sizes of training sets $n \in \{250, 500, 750, 1000\}$ and evaluate the prediction accuracy of the estimate using an independent test set of size $n' = 2000$. We then report the averaged MADs and the corresponding averaged timings in Table 6. Since the results are very close for different model settings, only the result from the heteroscedastic model with mixed-normal error is presented. We find that the sample size strongly affects predictive performance and timings: large samples give models with higher predictive accuracy at the expense of computational cost – the timings as least quadruple as one doubles sample size.

5. Real Data Application

In this section, we illustrate KERE by applying it to the personal computer price data studied in Stengos and Zacharias (2006). The data collected from *PC Magazine* from January of 1993 to November of 1995 have 6259 observations, each of which consists of the advertised price and features of personal computers

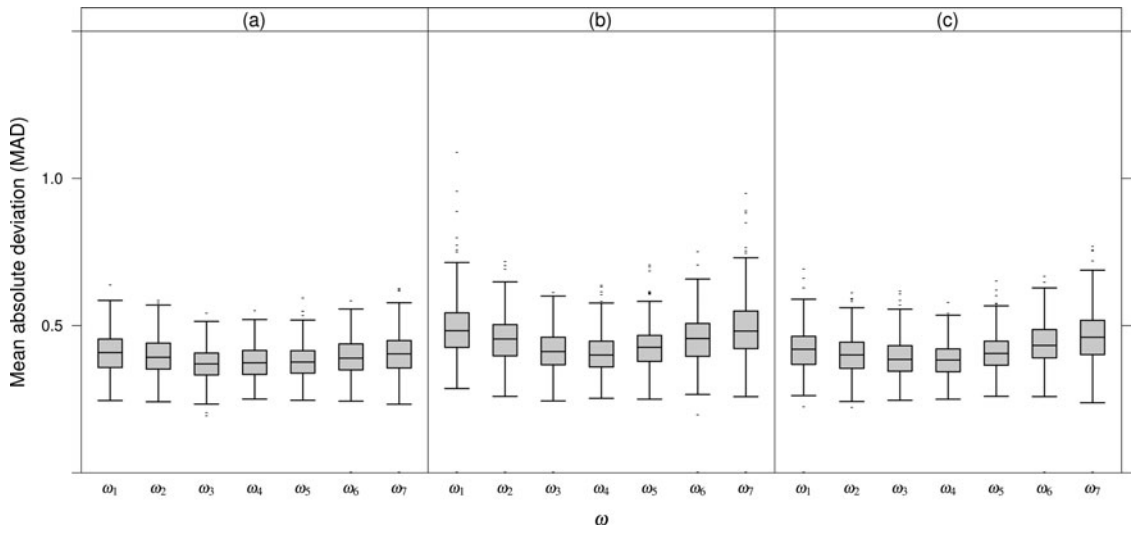

Figure 2. Homoscedastic models (KERE with the Gaussian kernel) with error distribution: (a) normal, (b) t_4 distribution, (c) mixed normal. Boxplots show MADs based on 300 independent runs for expectiles $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$.

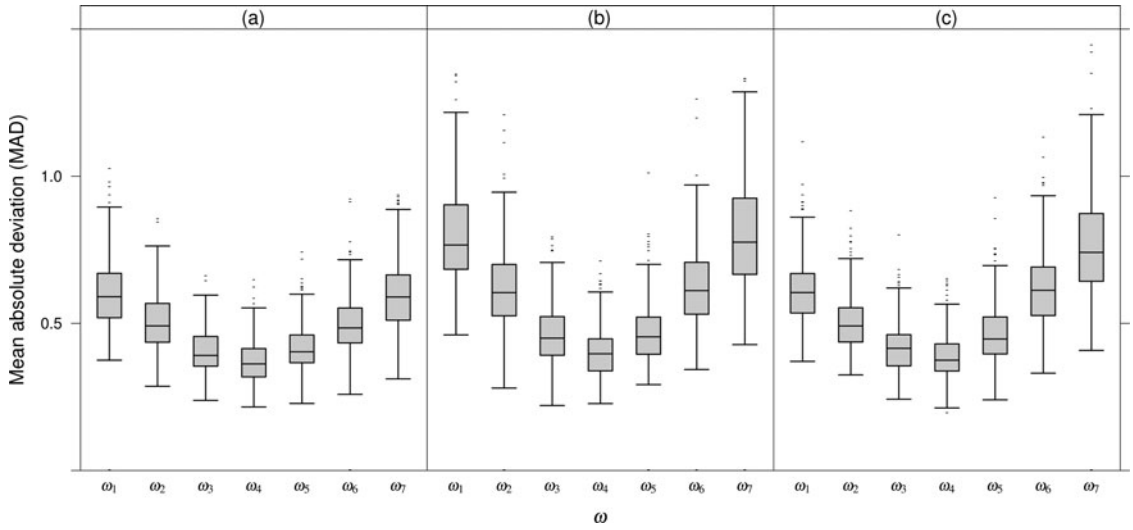

Figure 3. Heteroscedastic models with error distribution: (a) normal, (b) t_4 distribution, (c) mixed normal. Boxplots show MADs based on 300 independent runs for expectiles $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$.

sold in United States. There are nine main price detriments of PCs summarized in Table 7. The price and the continuous variables except the time trend are in logarithmic scale. We consider a hedonic analysis, where the price of a product is considered to be a function of the implicit prices of its various components, see Triplett (1989). The intertemporal effect of the implicit PC-component prices is captured by incorporating the time trend as one of the explanatory variables. The presence of nonlinearity and the interactions of the components with the time trend in the data, shown by Stengos and Zacharias (2006), suggest that the linear expectile regression may lead to a misspecified model. Since there lacks of a general theory about any particular functional form for the PC prices, we use KERE with three types of kernels (Gaussian, Laplacian, and hyperbolic tangent) to capture the nonlinear effects and higher order interactions of characteristics on price and avoid severe model misspecification.

We randomly sampled $1/R$ observations for training and tuning with two-dimensional five-fold cross-validation for selecting an optimal (σ, λ) pair, and the remaining $(R-1)/R$ observations as the test set for calculating the prediction error defined by

$$\text{prediction error} = \frac{1}{n'} \sum_{i=1}^{n'} \phi_\omega(y_i - \hat{f}_\omega(\mathbf{x}_i)).$$

To show the stability of our results with respect to different splitting ratios, we consider two cases $R = 3$ and $R = 10$. For comparison, we also computed the prediction errors using the linear expectile model and the ER-Boost method, which is a gradient tree boosting algorithm for expectile regression proposed by Yang and Zou (2015). All prediction errors are computed for seven expectile levels $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$.

Table 2. Normal models: The averaged MADs and the corresponding standard errors for fitting homoscedastic and heteroscedastic models based on 300 independent runs.

Homoscedastic model			
ω	KERE (G)	KERE (L)	KERE (H)
0.05	0.407 (0.004)	0.415 (0.004)	0.481 (0.006)
0.1	0.398 (0.004)	0.395 (0.003)	0.462 (0.005)
0.25	0.372 (0.003)	0.372 (0.003)	0.436 (0.004)
0.5	0.375 (0.003)	0.366 (0.003)	0.425 (0.004)
0.75	0.378 (0.003)	0.371 (0.003)	0.434 (0.004)
0.9	0.393 (0.004)	0.393 (0.003)	0.451 (0.004)
0.95	0.404 (0.005)	0.407 (0.004)	0.475 (0.006)
Heteroscedastic model			
ω	KERE (G)	KERE (L)	KERE (H)
0.05	0.601 (0.008)	0.608 (0.007)	0.707 (0.007)
0.1	0.507 (0.006)	0.505 (0.004)	0.598 (0.006)
0.25	0.407 (0.004)	0.408 (0.004)	0.465 (0.004)
0.5	0.371 (0.004)	0.373 (0.004)	0.436 (0.004)
0.75	0.419 (0.005)	0.403 (0.004)	0.477 (0.004)
0.9	0.497 (0.006)	0.502 (0.005)	0.593 (0.005)
0.95	0.594 (0.007)	0.598 (0.006)	0.703 (0.007)

NOTES: The expectile levels are $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$. KERE (G) uses the Gaussian kernel; KERE (L) uses the Laplacian kernel; KERE (H) uses the hyperbolic tangent kernel.

Table 3. t_4 models: The averaged MADs and the corresponding standard errors for fitting homoscedastic and heteroscedastic models based on 300 independent runs.

Homoscedastic model			
ω	KERE (G)	KERE (L)	KERE (H)
0.05	0.492 (0.006)	0.480 (0.006)	0.493 (0.006)
0.1	0.453 (0.005)	0.450 (0.005)	0.471 (0.005)
0.25	0.415 (0.004)	0.419 (0.004)	0.446 (0.004)
0.5	0.407 (0.004)	0.407 (0.004)	0.441 (0.005)
0.75	0.426 (0.004)	0.413 (0.004)	0.441 (0.004)
0.9	0.455 (0.005)	0.447 (0.004)	0.472 (0.005)
0.95	0.493 (0.006)	0.485 (0.006)	0.499 (0.006)
Heteroscedastic model			
ω	KERE (G)	KERE (L)	KERE (H)
0.05	0.804 (0.010)	0.811 (0.010)	0.864 (0.011)
0.1	0.632 (0.009)	0.624 (0.007)	0.672 (0.007)
0.25	0.465 (0.006)	0.458 (0.005)	0.507 (0.005)
0.5	0.404 (0.005)	0.398 (0.004)	0.454 (0.005)
0.75	0.470 (0.006)	0.451 (0.005)	0.521 (0.005)
0.9	0.623 (0.008)	0.627 (0.007)	0.681 (0.008)
0.95	0.808 (0.013)	0.789 (0.011)	0.861 (0.012)

NOTES: The expectile levels are $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$. KERE (G) uses the Gaussian kernel; KERE (L) uses the Laplacian kernel; KERE (H) uses the hyperbolic tangent kernel.

Table 4. Mixture models: The averaged MADs and the corresponding standard errors for fitting homoscedastic and heteroscedastic models based on 300 independent runs.

Homoscedastic model			
ω	KERE (G)	KERE (L)	KERE (H)
0.05	0.418 (0.005)	0.410 (0.004)	0.477 (0.006)
0.1	0.402 (0.004)	0.405 (0.004)	0.453 (0.005)
0.25	0.389 (0.004)	0.387 (0.003)	0.438 (0.004)
0.5	0.385 (0.003)	0.385 (0.003)	0.432 (0.004)
0.75	0.410 (0.004)	0.396 (0.003)	0.441 (0.004)
0.9	0.436 (0.005)	0.433 (0.004)	0.473 (0.005)
0.95	0.463 (0.005)	0.461 (0.004)	0.485 (0.005)
Heteroscedastic model			
ω	KERE (G)	KERE (L)	KERE (H)
0.05	0.614 (0.007)	0.614 (0.007)	0.702 (0.007)
0.1	0.505 (0.005)	0.508 (0.005)	0.582 (0.005)
0.25	0.417 (0.005)	0.411 (0.004)	0.475 (0.005)
0.5	0.389 (0.005)	0.382 (0.004)	0.440 (0.004)
0.75	0.464 (0.006)	0.450 (0.005)	0.504 (0.005)
0.9	0.620 (0.008)	0.606 (0.007)	0.663 (0.006)
0.95	0.763 (0.010)	0.747 (0.010)	0.824 (0.010)

NOTES: The expectile levels are $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$. KERE (G) uses the Gaussian kernel; KERE (L) uses the Laplacian kernel; KERE (H) uses the hyperbolic tangent kernel.

Table 5. The averaged computation times (in seconds) for fitting homoscedastic and heteroscedastic models (KERE with the Gaussian kernel) based on 300 independent runs.

	Homoscedastic model			Heteroscedastic model		
ω	Normal	t_4	Mixture	Normal	t_4	Mixture
0.05	19.04	21.47	17.10	16.90	17.60	17.95
0.1	14.25	16.89	13.91	14.38	14.60	15.21
0.25	11.67	15.25	13.59	12.30	12.49	12.36
0.5	10.54	14.09	12.18	10.92	11.13	11.01
0.75	8.24	15.33	10.47	12.48	12.48	12.38
0.9	10.08	14.39	12.46	14.67	15.25	14.52
0.95	12.16	19.90	15.17	17.34	17.75	16.61

NOTE: The expectile levels are $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$.

Table 6. The averaged MADs and the corresponding averaged computation times (in seconds) are reported. The size of the training set varies from 250 to 1000. The size of the test dataset is 2000. All models (KERE with the Gaussian kernel) are fitted on three expectile levels: (a) $\omega = 0.1$, (b) $\omega = 0.5$, and (c) $\omega = 0.9$.

	Error				Timing			
n	250	500	750	1000	250	500	750	1000
$\omega = 0.1$	0.4824	0.4084	0.4047	0.3887	8.739	56.188	168.636	382.897
$\omega = 0.5$	0.3329	0.2977	0.2732	0.2544	6.028	43.802	159.398	329.646
$\omega = 0.9$	0.6341	0.5861	0.5563	0.5059	9.167	56.533	173.359	386.345

Table 7. Explanatory variables in the Personal Computer Price Data (Stengos and Zacharias 2006).

ID	Variable	Explanation
1	SPEED	clock speed in MHz
2	HD	size of hard drive in MB
3	RAM	size of RAM in MB
4	SCREEN	size of screen in inches
5	CD	if a CD-ROM present
6	PREMIUM	if the manufacturer was a "premium" firm (IBM, COMPAQ)
7	MULTI	if a multimedia kit (speakers, sound card) included
8	ADS	number of 486 price listings for each month
9	TREND	time trend indicating month starting from Jan. 1993 to Nov. 1995

Table 8. The averaged prediction error and the corresponding standard errors for the Personal Computer Price Data based on 10 independent runs.

Personal Computer Price Data							
ω	0.05	0.1	0.25	0.5	0.75	0.9	0.95
KERE (G)	2.259	3.483	5.679	6.901	5.959	3.989	2.747
	(0.028)	(0.032)	(0.088)	(0.055)	(0.079)	(0.021)	(0.102)
KERE (L)	3.174	5.119	8.614	10.600	9.016	5.781	3.775
	(0.013)	(0.014)	(0.024)	(0.021)	(0.021)	(0.025)	(0.037)
KERE (H)	2.482	3.548	5.156	6.008	5.490	3.930	2.967
	(0.061)	(0.076)	(0.071)	(0.090)	(0.115)	(0.084)	(0.112)
ER-Boost	2.512	3.695	5.768	6.689	6.104	4.249	3.044
	(0.049)	(0.069)	(0.093)	(0.054)	(0.086)	(0.067)	(0.111)
Linear	13.019	19.319	28.958	32.661	28.470	19.523	13.606
	(0.064)	(0.107)	(0.214)	(0.211)	(0.188)	(0.133)	(0.055)

NOTES: The expectile levels are $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$. This case uses 1/10 observations for training and 9/10 for testing. The numbers in this table are of the order of 10^{-2}.

Table 9. The averaged prediction error and the corresponding standard errors for the Personal Computer Price Data based on 10 independent runs.

Personal Computer Price Data							
ω	0.05	0.1	0.25	0.5	0.75	0.9	0.95
KERE (G)	1.926	2.952	5.144	6.320	5.462	3.307	2.204
	(0.027)	(0.024)	(0.031)	(0.039)	(0.033)	(0.022)	(0.020)
KERE (L)	3.101	5.000	8.512	10.522	8.956	5.622	3.653
	(0.017)	(0.021)	(0.038)	(0.029)	(0.032)	(0.029)	(0.021)
KERE (H)	1.733	2.459	4.157	4.892	4.458	2.586	1.804
	(0.036)	(0.030)	(0.031)	(0.034)	(0.042)	(0.040)	(0.032)
ER-Boost	1.953	2.950	5.137	6.121	5.412	3.375	2.286
	(0.031)	(0.028)	(0.038)	(0.029)	(0.038)	(0.036)	(0.031)
Linear	12.598	18.168	24.235	27.718	23.801	18.234	12.972
	(0.070)	(0.152)	(0.228)	(0.082)	(0.132)	(0.154)	(0.084)

NOTES: The expectile levels are $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$. This case uses 1/3 observations for training and 2/3 for testing. The numbers in this table are of the order of 10^{-2}.

We repeated this process 10 times and reported the average prediction error and their corresponding standard errors in Tables 8 and 9. We can see that for all expectile levels, the linear expectile model always has the worst performance in terms of both prediction error and the corresponding standard errors. The best performers are KERE (Gaussian) for $R = 10$, and KERE (hyperbolic) for $R = 3$, followed by the ER-Boost. The ER-Boost and KERE models both offer much more flexible and accurate predictions than the linear model by guarding against model misspecification bias.

Supplementary Materials

Appendix: This supplementary file contains technical proofs for the theorems and lemmas not shown in the main article. (appendix.pdf)

R package KERE: This supplementary file is the R package KERE, which implements the algorithms proposed in this article. (KERE_1.1.0.tar.gz)

Acknowledgments

Yi and Teng are joint first authors. The authors sincerely thank the editor, associate editor and referees for their helpful comments and suggestions.

Zou's research is partially supported by NSF grant DMS-1505111. Yang's research is partially supported by NSERC RGPIN-2016-05174.

References

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999), "Coherent Measures of Risk," *Mathematical Finance*, 9, 203–228. [1]

Efron, B. (1991), "Regression Percentiles Using Asymmetric Squared Loss," *Statistica Sinica*, 55, 93–125. [1,2]

Embrechts, P., Puccetti, G., Rüschendorf, L., Wang, R., and Beleraj, A. (2014), "An Academic Response to Basel 3.5," *Risks*, 2, 25–48. [1]

Emmer, S., Kratz, M., and Tasche, D. (2013), "What is the Best Risk Measure in Practice? A Comparison of Standard Measures," *Journal of Risk*, 18, 31–60. [1]

Friedman, J. H. (2000), "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, 29, 1189–1232. [6]

Gneiting, T. (2011), "Making and Evaluating Point Forecasts," *Journal of the American Statistical Association*, 106, 746–762. [1]

Hamidi, B., Maillet, B., and Prigent, J.-L. (2014), "A Dynamic Autoregressive Expectile for Time-Invariant Portfolio Protection Strategies," *Journal of Economic Dynamics and Control*, 46, 1–29. [1]

Harville, D. A. (2008), *Matrix Algebra From a Statistician's Perspective*, New York: Springer. [5]

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics. [2,5]

Hunter, D., and Lange, K. (2004), "A Tutorial on MM Algorithms," *The American Statistician*, 58, 30–37. [3]

Kuan, C.-M., Yeh, J.-H., and Hsu, Y.-C. (2009), "Assessing Value at Risk With Care, the Conditional Autoregressive Expectile Models," *Journal of Econometrics*, 150, 261–270. [1]

Lange, K., Hunter, D., and Yang, I. (2000), "Optimization Transfer Using Surrogate Objective Functions," *Journal of Computational and Graphical Statistics*, 9, 1–20. [3]

Lange, K., and Zhou, H. (2014), "MM Algorithms for Geometric and Signomial Programming," *Mathematical Programming*, 143, 339–356. [3]

Newey, W. K., and Powell, J. L. (1987), "Asymmetric Least Squares Estimation and Testing," *Econometrica*, 55, 819–847. [2]

Sherman, J., and Morrison, W. J. (1950), "Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix," *The Annals of Mathematical Statistics*, 21, 124–127. [5]

Smola, A. J., Schölkopf, B., and Müller, K.-R. (1998), "The Connection Between Regularization Operators and Support Vector Kernels," *Neural Networks*, 11, 637–649. [2]

Sobotka, F., and Kneib, T. (2012), "Geoadditive Expectile Regression," *Computational Statistics & Data Analysis*, 56, 755–767. [1]

Stengos, T., and Zacharias, E. (2006), "Intertemporal Pricing and Price Discrimination: A Semiparametric Hedonic Analysis of the Personal Computer Market," *Journal of Applied Econometrics*, 21, 371–386. [7]

Taylor, J. W. (2008), "Estimating Value at Risk and Expected Shortfall Using Expectiles," *Journal of Financial Econometrics*, 6, 231–252. [1]

Triplett, J. E. (1989), "Price and Technological Change in a Capital Good: A Survey of Research on Computers," *Technology and Capital Formation*, 127–213. [8]

Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia, PA: SIAM. [1,2]

Wu, T., and Lange, K. (2010), "The MM Alternative to EM," *Statistical Science*, 4, 492–505. [3]

Xie, S., Zhou, Y., and Wan, A. T. K. (2014), "A Varying-Coefficient Expectile Model for Estimating Value at Risk," *Journal of Business & Economic Statistics*, 32, 576–592. [1]

Yang, Y., and Zou, H. (2015), "Nonparametric Multiple Expectile Regression via ER-Boost," *Journal of Statistical Computation and Simulation*, 85, 1442–1458. [1,2,8]

Yao, Q., and Tong, H. (1996), "Asymmetric Least Squares Regression Estimation: A Nonparametric Approach," *Journal of Nonparametric Statistics*, 55, 273–292. [1,2]

Zhou, H., and Lange, K. (2010), "MM Algorithms for Some Discrete Multivariate Distributions," *Journal of Computational and Graphical Statistics*, 19, 645–665. [3]

Ziegel, J. F. (2016), "Coherence and Elicitability," *Mathematical Finance*, 26, 901–918. [1]