Nonparametric Multiple Expectile Regression via ER-Boost

Yi Yang and Hui Zou

First version: April, 2013
Revision: October, 2013

Abstract

Expectile regression (Newey & Powell 1987) is a nice tool for estimating the conditional expectiles of a response variable given a set of covariates. Expectile regression at 50% level is the classical conditional mean regression. In many real applications having multiple expectiles at different levels provides a more complete picture of the conditional distribution of the response variable. Multiple linear expectile regression model has been well studied (Newey & Powell 1987, Efron 1991), but it can be too restrictive for many real applications. In this paper, we derive a regression tree based gradient boosting estimator for nonparametric multiple expectile regression. The new estimator, referred to as ER-Boost, is implemented in an R package `erboost` publicly available at `http://cran.r-project.org/web/packages/erboost/index.html`. We use two homoscedastic/heteroscedastic random-function-generator models in simulation to show the high predictive accuracy of ER-Boost. As an application, we apply ER-Boost to analyze North Carolina County crime data. From the nonparametric expectile regression analysis of this dataset, we draw several interesting conclusions that are consistent with the previous study using the economic model of crime. This real data example also provides a good demonstration of some nice features of ER-Boost, such as its ability to handle different types of covariates and its model interpretation tools.

Keywords: Asymmetric least squares, Expectile regression, Functional gradient descent, Gradient boosting, Regression tree.

1 Introduction

The goal of regression analysis is to gain knowledge about a response variable Y through a model (parametric or nonparametric) of explanatory variables X. There are several approaches to regression analysis and modeling. The most commonly used one is the conditional mean regression, which aims to estimate the optimal prediction function $E(Y|X)$ under the L_2 loss. However, in many applications one wants to know more about the relation between the response and the explanatory variables besides the conditional mean. Quantile regression (Koenker 2005) is a nice tool for such a purpose, providing estimates of the conditional quantiles of Y given X. Koenker & Bassett (1978) showed that one could estimate the conditional α-quantile by minimizing the empirical check loss. Note that the check loss function is defined as $\psi_\alpha(t) = |I(t \leq 0) - \alpha||t|$. Following the spirit of quantile regression, Newey & Powell (1987) considered estimating the conditional expectiles of Y given X. Similar to conditional quantiles, a series of conditional expectiles can summarize the relation between Y and X. Newey & Powell (1987) showed that one can estimate the conditional ω-expectile by minimizing the empirical asymmetric least squares (ALS), which has the expression $\phi(t \mid \omega) = |I(t \leq 0) - \omega|t^2$. Many authors studied the connection between quantile regression and expectile regression (Koenker 1992, 1993, Jones 1994, Yao & Tong 1996, Efron 1991). It is interesting to observe that quantile regression includes the conditional median regression as a special case ($\alpha = 0.5$) and expectile regression includes the conditional mean regression as a special case ($\omega = 0.5$). Quantile regression and expectile regression have their advantages over each other. Quantile regression can be more robust to outliers than expectile regression; Newey & Powell (1987) argued that expectile regression has at least two major advantages over quantile regression:

(1). it is computationally friendlier. Note that the ALS loss is differentiable everywhere while the check loss is singular at zero.

(2). the calculation of the asymptotic covariance matrix of the multiple linear expectile regression estimator does not involve calculating the values of the density function of the errors.

Because neither approach is uniformly superior, both methods have received a lot of attention in the literature.

Parametric expectile regression models can be too rigid for real applications. Yao & Tong (1996) considered the nonparametric expectile regression when the explanatory variable is one-dimensional and proposed local linear regression estimator, for which the asymptotic normality and the uniform consistency were established. However, the local fitting approach is not well suited for estimating a nonparametric multiple expectile regression function when the dimension of explanatory variables is more than five. In the current literature nonparametric multiple expectile regression is understudied, which motivates us to fulfill this need.

In this paper, we adopt the gradient tree boosting algorithm to derive a fully nonparametric multiple expectile regression method. Our proposal is motivated by the proven success of gradient tree boosting for classification and conditional mean regression problems (Friedman et al. 2000, Friedman 2001). Our proposal has several nice features. The method can easily handle many types of explanatory variables (numeric, binary, categorical) and is invariant under monotone transformations of explanatory variables. The method can easily incorporate complex interactions in the final estimator, reducing the potential modeling bias when interaction terms have non-ignorable effects. The gradient tree boosting estimator also provides useful model interpretation tools such as relative variable importance scores and partial dependence plots.

The rest of the paper is organized as follows. In Section 2 we briefly review quantiles and expectiles. The main methodological development of ER-Boost is presented in Section 3 where we also discuss some important implementation aspects of ER-Boost. We use simulation to show the high predictive accuracy of ER-Boost in Section 4. As an application, we apply ER-Boost to analyze North Carolina crime data in Section 5.

2 Expectiles and Quantiles

Due to the historical reason we first discuss quantile functions. Recall that the α-quantile of Y given $X = x$, denoted by $q_\alpha(x)$, is defined as

$$\alpha = P\{Y \leq q_\alpha(x) \mid X = x\}. \tag{2.1}$$

Quantile regression is based on the following key observation (Koenker & Bassett 1978)

$$q_\alpha(x) = \arg\min_f E\{\psi(Y, f \mid \alpha) \mid X = x\}, \tag{2.2}$$

where $\psi(y, f \mid \alpha)$ is the so-called check loss and

$$\psi(y, f \mid \alpha) = \begin{cases} (1-\alpha)|y-f| & y \le f, \\ \alpha|y-f| & y > f. \end{cases} \tag{2.3}$$

Consider a random sample of size N, $(y_i, X_i)_{1 \le i \le N}$. Then we can derive an estimator of $q_\alpha(x)$ by

$$\hat{q}_\alpha(x) = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \psi(y_i, f(x_i) \mid \alpha), \tag{2.4}$$

where $\mathcal{F}$ denotes a "parameter space". For example, in multiple linear quantile regression, $\mathcal{F}$ is the collection of all linear functions of X.

Obviously when $\alpha = 0.5$, $q_\alpha(x)$ is the conditional median and the check loss becomes the standard least absolute deviation (LAD) loss. We see that, by putting different weights on the positive and negative residuals, quantile regression can estimate more than just the median. Following the same spirit of asymmetric weights for positive and negative residuals, expectile regression uses a different loss function for regression analysis. Define the asymmetric least squares (ALS) loss as

$$\phi(y, f \mid \omega) = \begin{cases} (1-\omega)(y-f)^2 & y \le f, \\ \omega(y-f)^2 & y > f. \end{cases} \tag{2.5}$$

The conditional ω-expectile f_ω, $\omega \in (0, 1)$, of Y given $X = x$ can be defined as the minimizer of the expected loss (Newey & Powell 1987)

$$f_\omega(x) = \arg\min_f E\{\phi(Y, f \mid \omega) \mid X = x\}. \tag{2.6}$$

When $\omega = 0.5$, ALS loss reduces to the usual least squares loss and $f_{0.5}(x) = E(Y|X = x)$. Consider a random sample of size N, $(y_i, x_i)_{1 \le i \le N}$, expectile regression derives an estimator of $f_\omega(x)$ by minimizing the empirical ALS loss within a "parameter space" $\mathcal{F}$:

$$\hat{f}_\omega(x) = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \phi(y_i, f(x_i) \mid \omega). \tag{2.7}$$

Quantiles and expectiles are different but closely related. Newey & Powell (1987) pointed out that expectiles are determined by tail expectations while quantiles are determined the distribution function. More specifically, to make (2.6) hold, the expectile $f_\omega(x)$ must satisfy

$$\omega = \frac{E\{|Y - f_\omega(x)|I_{\{Y \le f_\omega(x)\}} \mid X = x\}}{E\{|Y - f_\omega(x)| \mid X = x\}}. \tag{2.8}$$

For comparison, we can rewrite the quantile function (2.1) as

$$\alpha = \frac{E\{I_{\{Y \le q_\alpha(x)\}} \mid X = x\}}{E\{1 \mid X = x\}}. \tag{2.9}$$

There exists a one-one mapping $\alpha \mapsto \omega = \omega(\alpha, x)$ such that $f_{\omega(\alpha,x)}(x) = q_\alpha(x)$, i.e., the conditional $\omega(\alpha, x)$-expectile equals the conditional α-quantile. Specifically, we have

$$\omega(\alpha, x) = \frac{\alpha q_\alpha(x) - \int_{-\infty}^{q_\alpha(x)} y dG_x(y)}{2[\alpha q_\alpha(x) - \int_{-\infty}^{q_\alpha(x)} y dG_x(y)] + [E(Y \mid x) - q_\alpha(x)]}, \tag{2.10}$$

where $G_x(y)$ is the conditional CDF of Y given $X = x$.

To make the connection more interesting, let us consider the canonical nonparametric regression model

$$Y = m(X) + \sigma(X) \cdot \epsilon, \tag{2.11}$$

where ϵ is the random error, which is assumed to be independent of X. Now, under model (2.11) it is easy to see that

$$q_\alpha(x) = m(x) + \sigma(x) q^*_\alpha, \quad f_\omega(x) = m(x) + \sigma(x) f^*_\omega, \tag{2.12}$$

where q^*_α and f^*_ω are the α-quantile and ω-expectile of ϵ. To match $q_\alpha(x)$ and $f_\omega(x)$, it is necessary and sufficient to choose $\omega = \omega(\alpha)$ such that $f^*_{\omega(\alpha)} = q^*_\alpha$. It is important to note that under model (2.11) the one-one mapping between expectile and quantile is independent of X. By (2.10) we have

$$\omega(\alpha) = \frac{\alpha q_\alpha - \int_{-\infty}^{q_\alpha} \epsilon dG(\epsilon)}{2[\alpha q_\alpha - \int_{-\infty}^{q_\alpha} \epsilon dG(\epsilon)] + [\mu - q_\alpha]}, \tag{2.13}$$

where $\mu = E(\epsilon)$. For example, if the error distribution is $N(0, 1)$ then

$$\omega(\alpha) = \frac{(2\pi)^{-1/2} \exp(-q^2_\alpha/2) + \alpha q_\alpha}{(2/\pi)^{1/2} \exp(-q^2_\alpha/2) + (2\alpha - 1) q_\alpha}.$$

Yao & Tong (1996) discussed the above connection between expectiles and quantiles under model (2.11). They further developed a local linear estimator of $f_\omega(x)$ when X

is one-dimensional and established its asymptotic normality. In theory their method can be extended to higher dimension settings, but in practice it is not easy to do so, because local regression suffers severely from the so-called "curse-of-dimensionality". Alternatively, in this work we introduce a tree-based boosting estimator for multiple expectile regression.

3 ER-Boost

In this section we develop the gradient tree boosting method for fitting a nonparametric multiple expectile regression function. We consider minimizing the empirical ALS loss in (2.7) by doing functional gradient descent in the "parameter space" of regression trees.

3.1 Algorithm

Boosting (Freund & Schapire 1997, 1996) is one of the most successful machine learning algorithms applied to both regression and classification problems. Its basic idea is to combine many prediction models called *base learners* in a smart way such that the combined model has a superior prediction performance. The first popular boosting algorithm was *AdaBoost* (Freund & Schapire 1997, 1996) designed to solve binary classification problems. Later, Breiman (1998, 1999) revealed that AdaBoost could be viewed as a functional gradient descent algorithm. Friedman et al. (2000) and Friedman (2001) further developed gradient boosting algorithms, which naturally extend AdaBoost to regression problems. In the literature there are many papers on the numerical and theoretical study of boosting. Due to space limitation we could not possibly list all the references here. For a nice comprehensive review of boosting algorithms, we refer interested readers to Bühlmann & Hothorn (2007). In this paper we adopt the gradient boosting algorithm introduced by Friedman (2001) to estimate the conditional expectile functions $f_\omega(x)$ defined in (2.6).

Let us start with the observed data $\{y_i, x_i\}_1^N$. Gradient boosting uses an iterative procedure to sequentially update the estimator and then stops after a sufficient number

of iterations. The initial estimate is given by

$$\hat{f}^{[0]}(x) = \arg\min_{\beta} \frac{1}{N} \sum_{i=1}^{N} \phi(y_i, \beta \mid \omega). \tag{3.1}$$

Write $\hat{f}^{[m-1]}(x)$ as the current fit at the m step of the gradient boosting procedure. Compute the negative gradient of $\phi(\cdot \mid \omega)$ evaluated at $f = \hat{f}^{[m-1]}(x_i)$:

$$u_i^{[m]} = -\frac{\partial \phi(y_i, f \mid \omega)}{\partial f}\bigg|_{f=\hat{f}^{[m-1]}(x_i)} \tag{3.2}$$

$$= \begin{cases} 2(1-\omega)(y_i - \hat{f}^{[m-1]}(x_i)) & y_i \le \hat{f}^{[m-1]}(x_i), \\ 2\omega(y_i - \hat{f}^{[m-1]}(x_i)) & y_i > \hat{f}^{[m-1]}(x_i). \end{cases} \tag{3.3}$$

Then we find a base learner $b(x; \widehat{\boldsymbol{a}}^{[m]})$ to approximate the negative gradient vector $(u_1^{[m]}, \ldots, u_N^{[m]})$ under a least-squares criterion

$$\widehat{\boldsymbol{\alpha}}^{[m]} = \arg\min_{\boldsymbol{a}} \sum_{i=1}^{N} [u_i^{[m]} - b(x_i; \boldsymbol{a})]^2. \tag{3.4}$$

Our base learner is an L-terminal regression tree that partitions the explanatory variable space into L disjoint regions R_l, $j = 1, 2, \ldots, L$ and predicts a constant h_l to each region. In other words, each base learner has the expression

$$b(x; \widehat{\boldsymbol{\alpha}}^{[m]}) = \sum_{l=1}^{L} \bar{u}_l^{[m]} I(x \in R_l^{[m]}), \tag{3.5}$$

with parameters $\widehat{\boldsymbol{\alpha}}^{[m]} = \{R_l^{[m]}, \bar{u}_l^{[m]}\}_{l=1}^{L}$. Note that within each region $R_l^{[m]}$,

$$\bar{u}_l^{[m]} = \text{mean}_{x_i \in R_l^{[m]}}(u_i^{[m]}).$$

Friedman et al. (2000) proposed a fast top-down "best-fit" algorithm to find the fitted terminal regions $\{R_l^{[m]}\}_{l=1}^{L}$. We use the same algorithm in our work to build the tree. However, in principle any good regression tree building algorithm can be used at this step.

The next step is to update the current estimate based on the base learner. We choose a best constant $\gamma_l^{[m]}$ to improve the current estimate in region $R_l^{[m]}$ in the sense that

$$\hat{\gamma}_l^{[m]} = \arg\min_{\gamma} \frac{1}{N} \sum_{x_i \in R_l^{[m]}} \phi(y_i, \hat{f}^{[m-1]}(x_i) + \gamma \mid \omega), \qquad l = 1, \ldots, L. \tag{3.6}$$

Algorithm 1 Solving the optimization problem $\min_\beta S^{-1}\sum_i \phi(z_s, \beta \mid \omega)$

1. Sort $\{z_s\}_1^S$ increasingly as $\{z_{(s)}\}_1^S$, and let $z_{(0)} = -\infty$, and $z_{(S+1)} = \infty$.
2. Compute $\hat{\beta}_k = \frac{\sum_{s=1}^S (1-\omega) z_{(s)} I(s \le k) + \omega z_{(s)} I(s \ge k+1)}{\sum_{s=1}^S (1-\omega) I(s \le k) + \omega I(s \ge k+1)}$ for $k = 0, 1, \cdots, S$.
3. For $k = 0, 1, \cdots, S$, find the only k^* that satisfies

$$z_{(k^*)} \le \hat{\beta}_{k^*} \le z_{(k^*+1)}.$$

4. The minimizer of the problem is $\hat{\beta}_{k^*}$.

Having found the parameters $\gamma_l^{[m]}, 1 \le l \le L$, we then update the current estimate $\hat{f}^{[m-1]}(x)$ by

$$\hat{f}^{[m]}(x) = \hat{f}^{[m-1]}(x) + \nu \gamma_l^{[m]} I(x \in R_l^{[m]}), \qquad \text{for } x \in R_l^{[m]}, \tag{3.7}$$

where $0 < \nu \le 1$ is the shrinkage parameter (Friedman 2001) that controls the learning rate. Friedman (2001) has found that the shrinkage factor improves estimation.

Finally, we discuss how to carry out the computations in (3.1) and (3.6). Mathematically they are the same optimization problem:

$$\min_\beta \sum_s \phi(z_s, \beta \mid \omega).$$

In (3.1) z is the response variable y, while in (3.6) z is the current residual $y - \hat{f}^{[m-1]}(x)$ evaluated inside region $R_l^{[m]}$. The following lemma shows how to calculate the unique minimizer exactly.

Lemma 1. *Given $\{z_s\}_1^S$, the unique minimizer of $\sum_{s=1}^S \phi(z_s, \beta \mid \omega)$, denoted by $\hat{\beta}$, can be rigorously calculated using Algorithm 1.*

With Lemma 1 and Algorithm 1 we can do all the needed computations for completing the update in (3.7), a boosting step. The boosting step is repeated M times and then we report $\hat{f}^{[M]}(x)$ as the final estimate. In summary, the complete ER-Boost algorithm for expectile regression is shown in Algorithm 2.

3.2 Implementation

We now discuss some important implementation details of ER-Boost. In principle, one could use other types of base learners in functional gradient descent to derive a

boosting algorithm for expectile regression. We prefer regression trees for several good reasons. First, gradient tree boosting has proven to be very successful for conditional mean regression. Second, regression trees are invariant under monotone transformation of explanatory variables and naturally handles all types of explanatory variables. ER-Boost inherits those nice features. Third, but not last, using L-terminal trees allow us to include $L-1$ way interactions in the final estimate. This flexibility is very convenient and important in real applications.

Tuning There are three meta parameters in Algorithm 2: L (the size of the trees), ν (the shrinkage constant) and M (the number of boosting steps). For mean regression and logistic regression Friedman (2001) has found that smaller values of ν result in better predictive accuracy at a cost of large M values and hence more computing time. Following Friedman (2001) we fix ν as $\nu = 0.005$ throughout. Then only L and M are to be determined by the data. Selection of L is very important. If we want to fit an additive model, then we can fix $L = 2$. Likewise, if we only want to fit a model with main effects and two-way interactions, we can fix $L = 3$. However, in many applications we do not have such prior knowledge or preference about the underlying model, then we should use data to determine which L value is the best. If N is reasonably large, we can split the observed data into two parts – training and validation. For a given L, we run ER-Boost and report the validation ALS loss at each boosting step

$$VALS(L, M) = \sum_{validation} \phi(y_i, \hat{f}^{[M]}(x_i) \mid \omega).$$

Then we stop ER-Boost when the minimum validation ALS loss is reached, i.e. $M_L^* = \arg\min_M VALS(L, M)$. If we need to choose L too, then we repeat the process for several L (say, $L = 2, 3, 4, 5, 6$) and report the one with the smallest minimum validation ALS loss, i.e. $L^* = \arg\min_L VALS(L, M_L^*)$.

Measure of relative importance Following Friedman (2001) and Ridgeway (2007), we define a measure of importance of any explanatory variable X_j for the ER-Boost model with a combination of M regression trees. Specifically, the relative

importance $\mathcal{I}_j$ of variable X_j is defined as the averaged importance over regression trees $\{T_1, \ldots, T_M\}$,

$$\mathcal{I}_j = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_j(T_m), \tag{3.8}$$

where $\mathcal{I}_j(T)$ is the importance of variable X_j in tree T. In ER-Boost, each regression tree is used to fit the gradient under the squared error loss. Thus we follow Breiman et al. (1984) to define $\mathcal{I}_j(T)$ as

$$\mathcal{I}_j(T) = \sqrt{\sum_t \hat{\xi}_t^2 I(X_j \text{ is the splitting variable for node } t)}. \tag{3.9}$$

Inside the square root is the sum of $\hat{\xi}_t^2$ over all internal nodes when X_j is chosen as the splitting variable, and $\hat{\xi}_t^2$ is the maximal squared error reduction induced by the partition of the region associated with node t into two sub-regions.

The value $\mathcal{I}_j$ alone as a measure of variable importance is not enough. Because even if there is no correlation between Y and X_j, X_j can still be possibly selected as splitting variable, hence the relative importance of X_j is non-zero by (3.9). Following Breiman (2001) and Kriegler & Berk (2010), we compute the "relative importance baseline" for each explanatory variable by re-sampling explanatory variables one at time and calculating the corresponding relative importance. The following is the procedure for computing each explanatory variable's baseline relative importance:

For $j = 1, \ldots, p$, repeat steps 1–4.

1. Randomly re-shuffle the values of X_j while keeping all other explanatory variables' values unchanged.
2. Fit the ER-Boost model using the modified dataset in step 1 and compute the relative importance for X_j. The same tuning method is used.
3. Repeat step 1 and 2 for 100 times, each time a relative importance of X_j is computed from a re-shuffled dataset.
4. Report the averaged relative importance of X_j of 100 repetitions.

Partial dependence plots Using relative importance measure we can rank explanatory variables. The next natural step is to look at the main effect of each important variable and their possible significant interactions. For that, Friedman (2001)

suggested using partial dependence plots. Let X_S be the sub-vector of p-dimensional explanatory variables X, where $S \subset \{1, 2, \ldots, p\}$ and $S \cup S^c = \{1, 2, \ldots, p\}$. For example, for the main effect of variable j $S = \{j\}$ and for the two-way interaction of variables i and j $S = \{i, j\}$. The partial dependence of $\hat{f}(X_S)$ on X_S can be estimated by (Friedman 2001),

$$\bar{f}(X_S) = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(X_S, x_{iS^c}),$$

where $\{x_{iS^c}\}_i^N$ are values corresponding to X_{S^c} in the training data. We plot $\bar{f}(X_S)$ against X_S to make the partial dependence plots.

Software We provide an implementation of the ER-Boost algorithm, along with discussed model interpretation tools, in the R package `erboost` which is publicly available at `http://cran.r-project.org/web/packages/erboost/index.html`.

4 Simulation

In this section we evaluate the performance of ER-Boost by simulation. All numerical experiments were carried out on an Intel Xeon X5560 (Quad-core 2.8 GHz) processor.

Setting I: Homoscedastic model In the first set of simulations we adopt the "random function generator" model by Friedman (2001). The idea is to see the performance of the estimator on a variety of randomly generated targets. We generated data $\{y_i, x_i\}_1^N$ according to

$$y_i = f(x_i) + \epsilon_i,$$

where ϵ_is are independent generated from some error distribution. Each of f functions is randomly generated as a linear combination of functions $\{g_l\}_1^{20}$:

$$f(x) = \sum_{l=1}^{20} a_l g_l(z_l), \tag{4.1}$$

where coefficients $\{a_l\}_1^{20}$ are randomly generated from a uniform distribution $a_l \sim \mathrm{U}[-1, 1]$. Each $g_l(z_l)$ is a function of a randomly selected p_l-size subset of the p-dimensional variable x, where the size of each subset p_l is randomly chosen by $p_l =$

Algorithm 2 ER-Boost

1. Initialize $\hat{f}^{[0]}(x)$.

$$\hat{f}^{[0]}(x) = \arg\min_{\beta} \frac{1}{N} \sum_{i=1}^{N} \phi(y_i, \beta \mid \omega).$$

(Call Algorithm 1 with $z_i = y_i$ for $i = 1, \ldots, N$.)

2. For $m = 1, \ldots, M$ repeatedly do steps 2.(a)–2.(d)

2.(a) Compute the negative gradient

$$u_i^{[m]} = \begin{cases} 2(1-\omega)(y_i - \hat{f}^{[m-1]}(x_i)) & y_i - \hat{f}^{[m-1]}(x_i) \le 0, \\ 2\omega(y_i - \hat{f}^{[m-1]}(x_i)) & y_i - \hat{f}^{[m-1]}(x_i) > 0. \end{cases} \qquad i = 1, \ldots, N.$$

2.(b) Fit the negative gradient vector $u_1^{[m]}, \ldots, u_N^{[m]}$ to $x_1, \ldots, x_N$ by an L-terminal node regression tree, giving us the partitions $\{R_l^{[m]}\}_{l=1}^{L}$.

2.(c) Compute the optimal terminal node predictions $\hat{\gamma}_l^{[m]}$ for each region $R_l^{[m]}$, $l = 1, 2, \ldots, L$.

$$\hat{\gamma}_l^{[m]} = \arg\min_{\gamma} \frac{1}{N} \sum_{x_i \in R_l^{[m]}} \phi(y_i, \hat{f}^{[m-1]}(x_i) + \gamma \mid \omega).$$

(Call Algorithm 1 with $z_i = y_i - \hat{f}^{[m-1]}(x_i)$ for $\{i : x_i \in R_l^{[m]}\}$.)

2.(d) Update $\hat{f}^{[m]}(x)$ for each region $R_l^{[m]}$, $l = 1, 2, \ldots, L$.

$$\hat{f}^{[m]}(x) = \hat{f}^{[m-1]}(x) + \nu\gamma_l^{[m]} I(x \in R_l^{[m]}), \qquad \text{if } x \in R_l^{[m]}.$$

3. Report $\hat{f}^{[M]}(x)$ as the final estimate.

$\min(\lfloor 1.5 + r \rfloor, p)$, and r is generated from an exponential distribution $r \sim \text{Exp}(0.5)$ with mean 2. Each z_l is defined as

$$z_l = \{x_{W_l(j)}\}_{j=1}^{p_l}, \tag{4.2}$$

where each W_l is an independent permutation of the integers $\{1, \ldots, p\}$. Each function $g_l(z_l)$ is an p_l-dimensional Gaussian function:

$$g_l(z_l) = \exp\left[-\frac{1}{2}(z_l - \mu_l)^T \mathbf{V}_l (z_l - \mu_l)\right], \tag{4.3}$$

where each of the mean vectors $\{\mu_l\}_1^{20}$ is randomly generated from the same distribution as that of the input variables x. The $p_l \times p_l$ covariance matrix $\mathbf{V}_l$ is also randomly generated by

$$\mathbf{V}_l = \mathbf{U}_l \mathbf{D}_l \mathbf{U}_l^T, \tag{4.4}$$

where $\mathbf{U}_l$ is a random orthonormal matrix and $\mathbf{D}_l = diag\{d_{1l} ... d_{p_l l}\}$. The variables d_{jl} are randomly generated from a uniform distribution $\sqrt{d_{jl}} \sim \text{U}[0.1, 2.0]$. In this section we generated X from joint normal distribution $x \sim \text{N}(0, \mathbf{I}_p)$ with $p = 10$. We considered three types of error distribution:

1. Normal distribution $\epsilon \sim \text{N}(0, 1)$.
2. Student's t-distribution with 4 degrees of freedom $\epsilon \sim \text{t}_4$.
3. Mixed normal distribution $\epsilon \sim 0.9\text{N}(0, 1) + 0.1\text{N}(1, 5)$.

Setting II: Heteroscedastic model In the second set of simulations we modified the "random function generator" model to include heteroscedastic error. Everything stayed the same except that we generated data $\{y_i\}_1^N$ according to

$$y_i = f(x_i) + |\sigma(x_i)|\epsilon_i,$$

where both f and σ were independently generated by the random function generator.

We used ER-Boost to estimate the expectile functions at seven levels:

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

The shrinkage constant ν is 0.005. For each model we generated three independent datasets: a training set with N observations for model estimation, a validation set

with N' observations for selecting the optimal (M, L) pair, and a test set with N'' observations for evaluating the performance of the final estimate. Following Friedman (2001) the test error is measured by the mean absolution deviation

$$\text{MAD} = \frac{1}{N''} \sum_{i=1}^{N''} |f_\omega(x_i) - \hat{f}_\omega(x_i)|.$$

Note that the target function $f_\omega(x)$ is equal to $f(x) + b_\omega(\epsilon)$ in the homoscedastic model and $f(x) + |\sigma(x)| b_\omega(\epsilon)$ in the heteroscedastic model, where $b_\omega(\epsilon)$ is the ω-expectile of the error distribution. See also (2.12). In our study $N = 500$, $N' = 200$ and $N'' = 2,000$.

We show box-plots of MADs in Figure 1 and 2 and report the average MADs and standard errors in Table 1. We can see that the prediction accuracy is very good in all examples, although the estimation appears to be more difficult in the heteroscedastic model as expected. Normal and t_4 are symmetric distributions. Their prediction MADs also appear to be symmetric around $\omega = 0.5$ (the conditional mean). However, the prediction MAD is asymmetric in the skewed mixed-normal distribution case.

We also study the effect of sample size on predictive performance. For this analysis, we fit the ER-Boost model using various sizes of training sets with $N \in \{400,\ 800,\ 1600,\ 4800\}$ and validation sets with $N' \in \{100,\ 200,\ 400,\ 1200\}$, and evaluate the performance of the final estimate using an independent test set of size $N'' = 6,000$. The models are fitted over a range of values for $L \in \{1,\ 2,\ 3,\ 5,\ 7,\ 10\}$ while the shrinkage constant ν is fixed at 0.005. We then report the minimum predicted ALS loss achieved by the chosen L and the corresponding optimal choice of M. Since the results are mostly similar for different simulation settings, here we only show the result from the heteroscedastic model with mixed-normal distribution defined in Setting II.

As shown in Figure 3, sample size strongly influences predictive performance: large samples produce models with lower predictive error. Gains in prediction accuracy from the increased tree size are greater with larger data sets, presumably because more data contain more detailed information, and larger sized trees can better model the complexity in that information. Decision stumps ($L = 1$) always produce higher predictive error but for small samples there was no advantage and even little sacrifice in prediction accuracy for using very large trees (higher L).

Homoscedastic model						
ω	Normal		t_4		Mixed-normal	
	MAD	Time	MAD	Time	MAD	Time
0.05	0.355 (0.003)	2.86	0.474 (0.006)	2.85	0.361 (0.003)	2.86
0.1	0.334 (0.003)	2.89	0.422 (0.005)	2.89	0.339 (0.003)	2.90
0.25	0.314 (0.002)	2.92	0.369 (0.003)	2.88	0.318 (0.002)	2.91
0.5	0.307 (0.002)	2.89	0.350 (0.003)	2.92	0.312 (0.002)	2.89
0.75	0.315 (0.003)	2.88	0.373 (0.003)	2.88	0.319 (0.002)	2.89
0.9	0.333 (0.003)	2.87	0.427 (0.005)	2.87	0.339 (0.002)	2.87
0.95	0.353 (0.003)	2.89	0.473 (0.006)	2.85	0.360 (0.003)	2.86

Table 1: Setting I, homoscedastic models. The averaged MADs and the corresponding standard errors based on 200 independent replications. $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$. The corresponding averaged computation times (in seconds) are also reported.

Heteroscedastic model						
ω	Normal		t_4		Mixed-normal	
	MAD	Time	MAD	Time	MAD	Time
0.05	0.549 (0.007)	2.88	0.774 (0.013)	2.87	0.521 (0.007)	2.86
0.1	0.455 (0.005)	2.96	0.600 (0.010)	2.93	0.431 (0.005)	2.93
0.25	0.359 (0.004)	2.89	0.424 (0.007)	2.90	0.346 (0.005)	2.91
0.5	0.321 (0.004)	2.89	0.363 (0.006)	2.90	0.321 (0.004)	2.89
0.75	0.355 (0.004)	2.88	0.426 (0.007)	2.88	0.373 (0.005)	2.89
0.9	0.450 (0.005)	2.87	0.601 (0.009)	2.90	0.477 (0.006)	2.87
0.95	0.543 (0.007)	2.86	0.773 (0.012)	2.87	0.579 (0.008)	2.87

Table 2: Setting II, heteroscedastic models. The averaged MADs and the corresponding standard errors based on 200 independent replications. $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$. The corresponding averaged computation times (in seconds) are also reported.

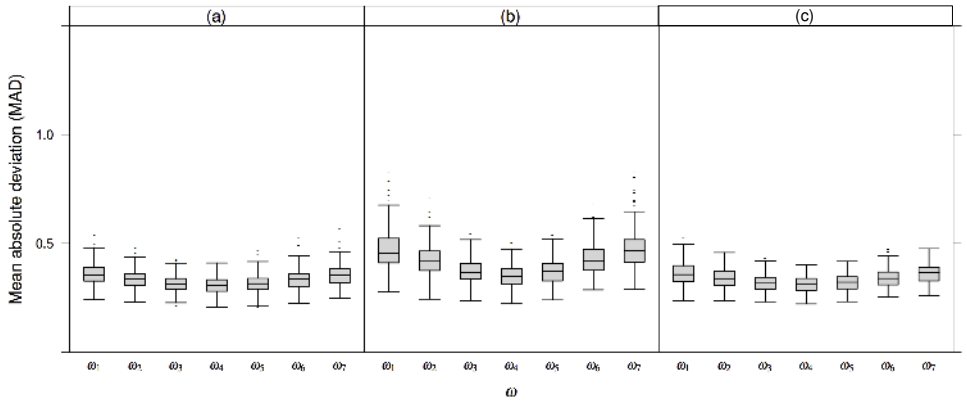

Figure 1: Setting I, homoscedastic models. Box-plots of MADs for expectiles $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$ based on 200 independent replications. The error distribution: (a) normal, (b) t_4 distribution, (c) mixed normal.

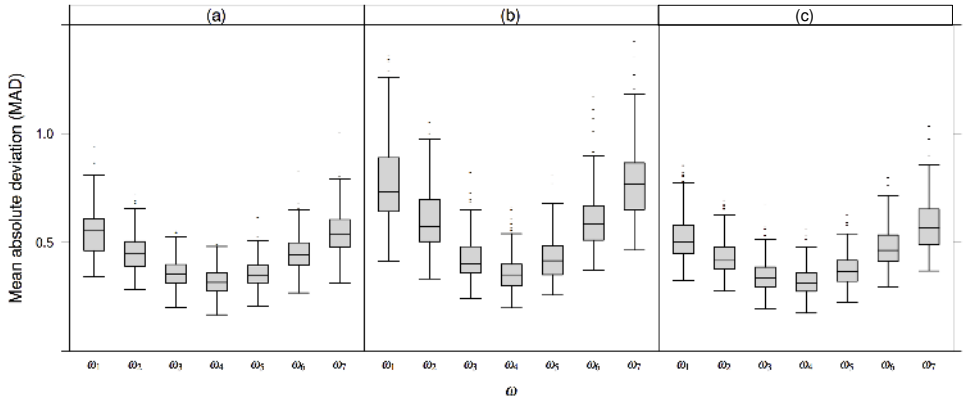

Figure 2: Setting II, heteroscedastic models. Box-plots of MADs for expectiles $\omega \in \{0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95\}$ based on 200 independent replications. The error distribution: (a) normal, (b) t_4 distribution, (c) mixed normal.

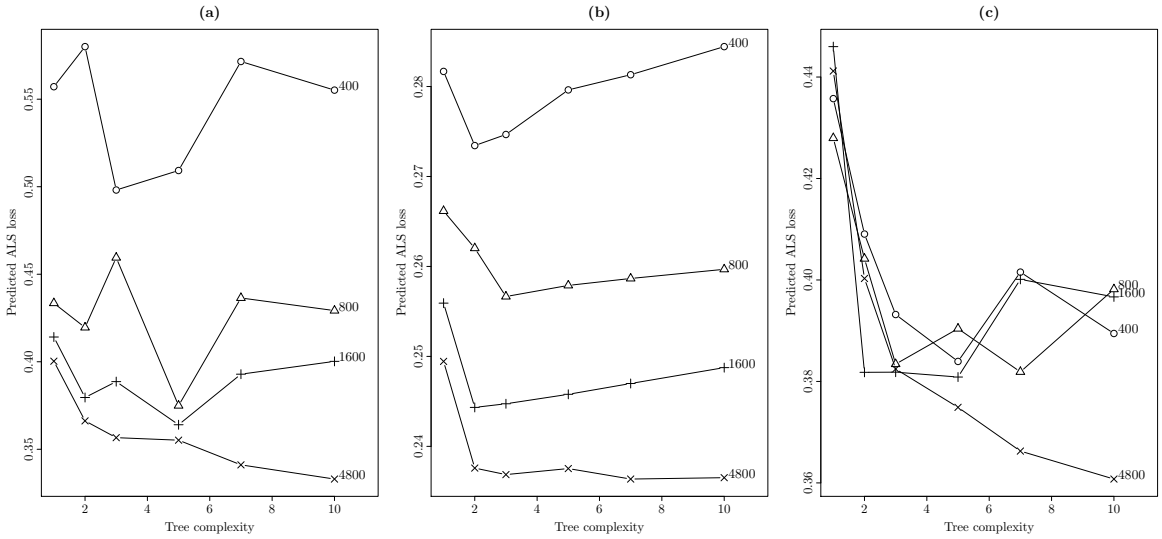

Figure 3: Predicted ALS loss as a function of sample size and tree complexity. Models are fitted on the training sets of 400-4800 observations, and minimum predicted ALS loss is estimated on an independent test data set of 6000. (a) $\omega = 0.1$, (b) $\omega = 0.5$, (c) $\omega = 0.9$.

5 North Carolina Crime Data

In this section we apply ER-Boost to analyze the North Carolina crime data. In previous study by Cornwell & Trumbull (1994) and Baltagi (2006) the crime rates (the ratio of FBI index crimes to county population) of North Carolina counties were related to a set of explanatory variables, including deterrent variables and variables measuring returns of legal opportunities, as well as other county characteristics. The dataset contains 630 records measured over the period 1981-1987 for 90 countries in North Carolina. Table 3 summarizes 19 explanatory variables for each sample. The economic model of crime is based on the assumption that individual's participation in the criminal sector depends on the relative monetary benefits against the costs the illegal activities (cf. Becker 1968, Ehrlich 1973, Block & Heineke 1975). Cornwell & Trumbull (1994) showed both labor market and criminal justice strategies are important in deterring crimes. The skewed distribution of the crime rate and the presence of county heterogeneity in the data shown by previous study (Cornwell & Trumbull 1994) suggest that by estimating several expectiles—including the conditional mean as one of them—we could gain more information about the crime rate.

ID	Variables	Type	Details
1	P_A	1	the ratio of arrests to offenses
2	P_C	1	the ratio of of convictions to offenses
3	P_P	1	the ratio of of prison sentences to offenses
4	S	1	average prison sentence in days
5	POLICE	1	police per capita
6	WCON	2	weekly wage in construction
7	WTUC	2	weekly wage in transportation, utilities, communications
8	WTRD	2	weekly wage in wholesales and retail trade
9	WFIR	2	weekly wage in finance, insurance and real estate
10	WSER	2	weekly wage in service industry
11	WMFG	2	weekly wage in manufacturing
12	WFED	2	weekly wage of federal employees
13	WSTA	2	weekly wage of state employees
14	WLOC	2	weekly wage of local governments employees
15	DENSITY	3	population per square mile
16	PCTMIN	3	percentage minority or non-white
17	PCTYMLE	3	percentage of young males between the ages of 15-24
18	REGION	3	one of 'other', 'west' or 'central'
19	URBAN	3	'yes' or 'no' if the county is in the SMSA[a]

[a]whether the county is a U.S. metropolitan statistical area and populations are over 50,000.

Table 3: Explanatory variables in the North Carolina crime data. Type 1, deterrent variables; Type 2, variables measuring returns of legal opportunities; Type 3, county characteristics.

Timings: North Carolina Crime Data					
ω	0.1	0.25	0.5	0.75	0.9
Time (sec.)	107.48	105.87	108.34	106.93	105.76

Table 4: Timings (in seconds) for conducting five-fold cross-validation and fitting the final model with conditional expectiles $\omega \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$ for the North Carolina crime data.

We use five-fold cross-validation for choosing the optimal tuning parameters (L, M). During each fold the data is randomly split into a training set and a validation set with ratio 4:1. Model building is conducted on the training sets, and the optimal (L, M) pair is chosen from the model with a minimal cross-validation error. We then fit the final model with the chosen (L, M) using all of the data. The total computation times for conducting cross-validation and fitting the final model with conditional expectiles $\omega \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$ are also reported in Table 4.

See section 3.2. Figure 4 shows the five estimated expectiles for 20 randomly chosen explanatory variables. The varying width of the expectile band across observations suggests a moderate amount of heteroscedasticity. This figure also suggests that the conditional distribution of the crime rate tends to be skewed, which is consistent with the previous study (Cornwell & Trumbull 1994).

Figure 5 shows the relative importance and baseline value of each explanatory variable for $\omega \in \{0.1, 0.5, 0.9\}$. If the relative importance (the dot) is larger than the baseline (the line-length), it indicates that the importance of that explanatory variable is real. We found that for all expectiles, *DENSITY*, *PCTMIN*, *POLICE*, P_A and *REGION* are the most important explanatory variables and their relative importance scores are significantly above the corresponding baselines. Interestingly, we find that the deterrent effect of S is small and insignificant. This result confirms the conclusion in Cornwell & Trumbull (1994) that the severity of punishment is not effective means of deterring crime, as opposed to previous studies (cf. Hirsch 1988). It is also notable that the relative importance of P_C and *PCTMIN* varies across different expectiles: the

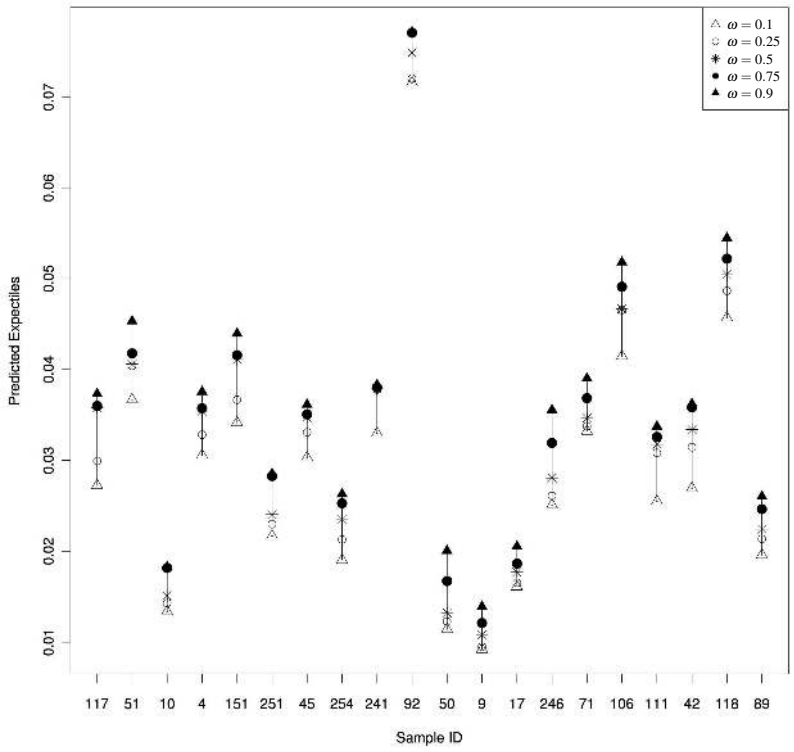

Figure 4: Five non-crossed estimated expectiles for 20 randomly chosen covariates (the sample IDs are on x-axis).

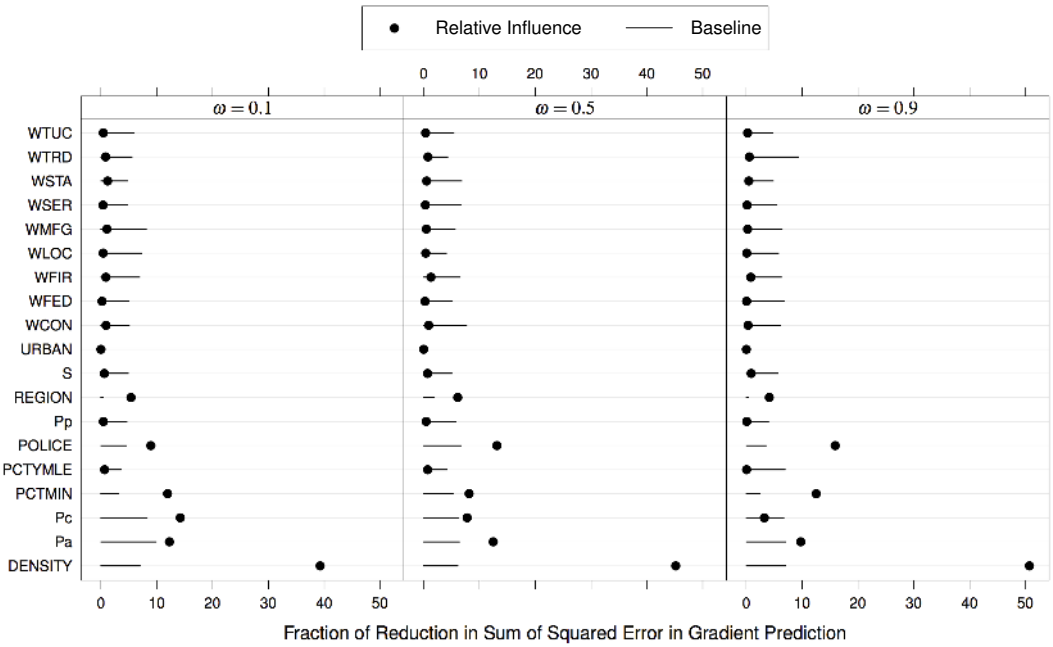

Figure 5: The relative importance and baselines of 19 explanatory variables for the models with conditional expectiles $\omega \in \{0.1, 0.5, 0.9\}$.

importance of P_C is more significant in low-crime-rate counties ($\omega = 0.1$), while the importance of *PCTMIN* is more significant in both low-crime-rate ($\omega = 0.1$) and high-crime-rate ($\omega = 0.9$) counties, but relatively less significant in the moderate-crime-rate ($\omega = 0.5$) counties.

To visualize the marginal effects of those significant explanatory variables on the crime rate, in Figure 6 we plot the partial dependence (Friedman 2001) of six explanatory variables, which have the highest relative importance values. In general, the dependence is more noticeable for the high crime rate case ($\omega = 0.9$) than for the low crime rate case ($\omega = 0.1$). We see that both P_A and P_C have deterrent effects on the crime rate. But when P_A passes a threshold 0.54 or when P_C passes 1.56, the crime rate curves become flat, which suggests higher ratio of punishment has little effect on crime rate once the former reaches a certain level. On the other hand, the partial dependence plots suggest that *POLICE*, *DENSITY* and *PCTMIN* have strong positive effects on the crime rate. The crime rate is positively associated with *POLICE*. This could be explained by the fact that a higher crime rate leads to hiring more policemen (cf. Cornwell & Trumbull 1994). We see that "central" region has higher crime rate

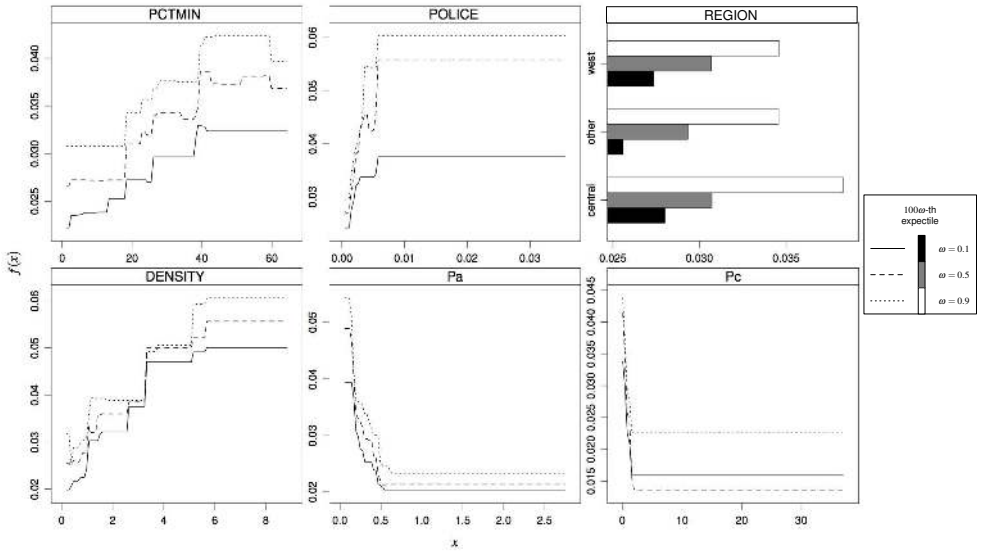

Figure 6: Partial dependence plots of crime rate versus 6 most significant explanatory variables for the models using conditional expectiles $\omega \in \{0.1, 0.5, 0.9\}$.

than "other" region. The partial dependence plots also indicate heteroscedasticity, as the marginal effects vary across different expectiles.

In our analysis it turned out that the data-driven choice for L is 3, which means that our ER-boost model has two-way interactions. We found that an important two-way interaction for $\omega = 0.1$ is $PCTMIN \times P_A$. A high $PCTMIN$ and high P_A are accompanied by high crime rate, and low $PCTMIN$ and low P_A are related to low crime rate. There are also strong $REGION \times POLICE$ interactions for $\omega = 0.5$ and 0.9. To visualize the marginal effect of two-way interactions we made the joint partial dependence plots as shown in Figure 7.

Acknowledgement

This work is supported in part by NSF grant DMS-0846068.

Appendix

Proof of Lemma 1. It is easy to see that $\phi(\cdot \mid \omega)$ is strictly convex and continuously differentiable as a function of β, and it goes to $+\infty$ as β goes to $-\infty$ or ∞. This

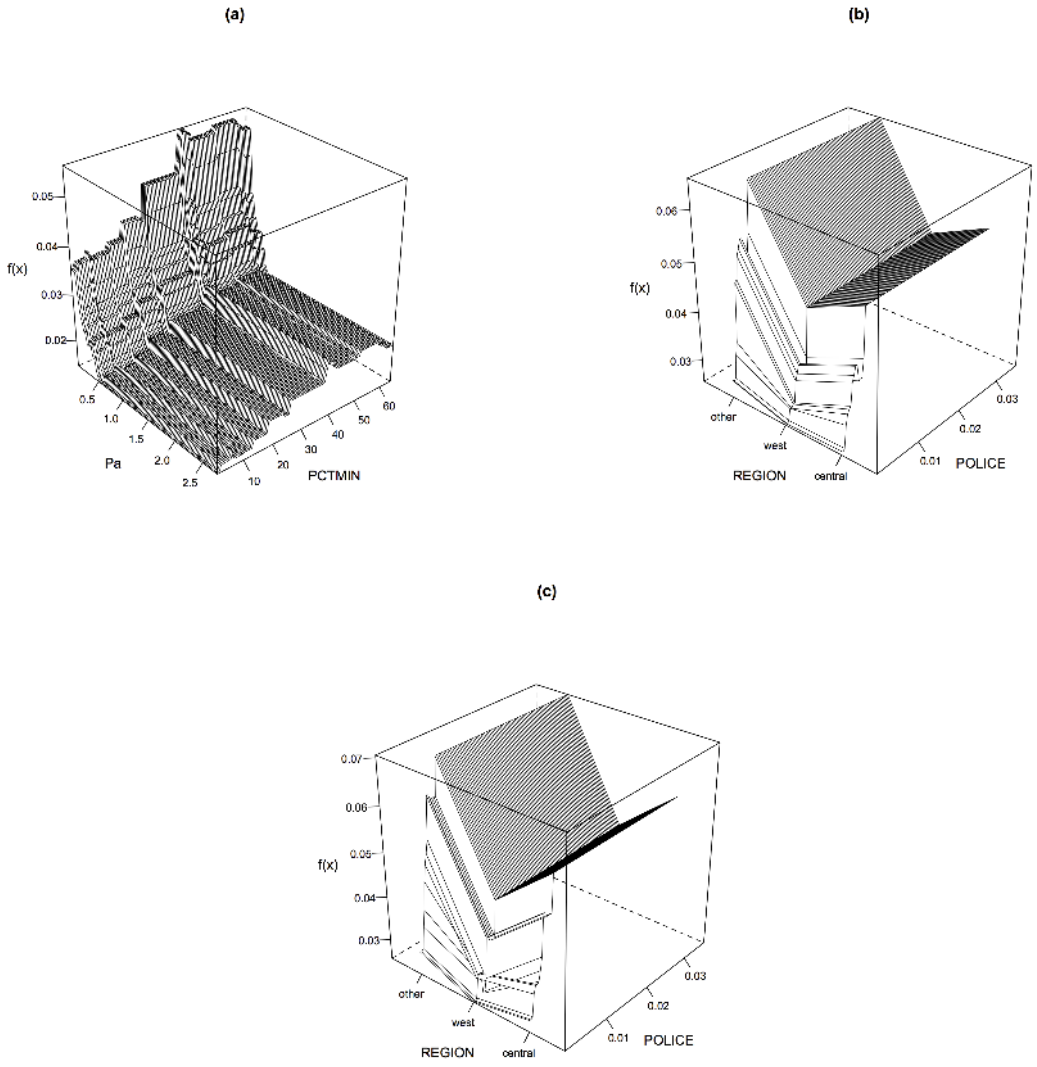

Figure 7: Partial dependence plots of the strong pairwise interactions. (a) $\omega = 0.1$, (b) $\omega = 0.5$ and (c) $\omega = 0.9$.

suggests that $\sum_s \phi(z_s, \beta \mid \omega)$ has a unique minimizer (either local or global).

With sorted $\{z_{(s)}\}_1^S$ and let $z_{(0)} = -\infty$ and $z_{(S+1)} = \infty$. $\hat{\beta}$ must be in $[z_{(k)}, z_{(k+1)}]$ for some k. Then the following equation holds,

$$\begin{aligned} \frac{\partial}{\partial \beta} \sum_{s=1}^{S} \phi(z_{(s)}, \beta \mid \omega)\Big|_{\beta=\beta_k} &= \frac{\partial}{\partial \beta} \left\{ \sum_{s=1}^{S} [(1-\omega) I(s \le k) + \omega I(s \ge k+1)](z_{(s)} - \beta)^2 \right\} \Big|_{\beta=\beta_k} \\ &= 0. \end{aligned}$$

Subsequently $\hat{\beta}$ should equal to a certain β_k, which is determined by

$$\beta_k = \frac{\sum_{s=1}^{S} (1-\omega) z_{(s)} I(s \le k) + \omega z_{(s)} I(s \ge k+1)}{\sum_{s=1}^{S} (1-\omega) I(s \le k) + \omega I(s \ge k+1)}.$$

We use the above formula to compute β_k for $k = 0, 1, \ldots, S$ and only one of them is $\hat{\beta}$. On the other hand, we note that β_k is a local minimizer in $[z_{(k)}, z_{(k+1)}]$ and hence the global minimizer $\hat{\beta}$, if and only if β_k is located in $[z_{(k)}, z_{(k+1)}]$. This suggests a way for finding $\hat{\beta}$: for $k = 0, \ldots, S$, we check whether β_k is located in $[z_{(k)}, z_{(k+1)}]$.

Another method for finding $\hat{\beta}$ is by directly comparing the objective function evaluated at β_k for $k = 0, 1, \ldots, S$, and $\hat{\beta}$ is the one yielding the smallest value. However, this method is computationally more expensive than Algorithm 1. □

References

Baltagi, B. (2006), 'Estimating an economic model of crime using panel data from north carolina', *Journal of Applied Econometrics* **21**(4), 543–547.

Becker, G. (1968), 'Crime and punishment: An economic approach', *Journal of Political Economy* **76**(2), 169–217.

Block, M. & Heineke, J. (1975), 'A labor theoretic analysis of the criminal choice', *The American Economic Review* **65**(3), 314–325.

Breiman, L. (1998), 'Arcing classifier (with discussion and a rejoinder by the author)', *The Annals of Statistics* **26**(3), 801–849.

Breiman, L. (1999), 'Prediction games and arcing algorithms', *Neural Computation* **11**(7), 1493–1517.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32.

Breiman, L., Friedman, J., Olshen, R., Stone, C., Steinberg, D. & Colla, P. (1984), 'Cart: Classification and regression trees', *Wadsworth* .

Bühlmann, P. & Hothorn, T. (2007), 'Boosting algorithms: Regularization, prediction and model fitting', *Statistical Science* **22**(4), 477–505.

Cornwell, C. & Trumbull, W. (1994), 'Estimating the economic model of crime with panel data', *The Review of Economics and Statistics* **76**, 360–366.

Efron, B. (1991), 'Regression percentiles using asymmetric squared error loss', *Statistica Sinica* **1**(93), 125.

Ehrlich, I. (1973), 'Participation in illegitimate activities: A theoretical and empirical investigation', *The Journal of Political Economy* **81**, 521–565.

Freund, Y. & Schapire, R. (1996), Experiments with a new boosting algorithm, *in* 'Machine learning: Proceedings of the Thirteenth International Conference', Morgan Kaufmann Publishers, Inc., pp. 148–156.

Freund, Y. & Schapire, R. (1997), 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of Computer and System Sciences* **55**, 119–139.

Friedman, J. (2001), 'Greedy function approximation: A gradient boosting machine', *The Annals of Statistics* **29**(5), 1189–1232.

Friedman, J., Hastie, T. & Tibshirani, R. (2000), 'Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors)', *The Annals of Statistics* **28**(2), 337–407.

Hirsch, W. (1988), *Law and economics: An introductory analysis. 2ed ed.*, Academic Press.

Jones, M. (1994), 'Expectiles and m-quantiles are quantiles', *Statistics & Probability Letters* **20**(2), 149–153.

Koenker, R. (1992), 'When are expectiles percentiles?', *Econometric Theory* **8**(03), 423–424.

Koenker, R. (1993), 'When are expectiles percentiles? (solution)', *Econometric Theory* **9**(03), 526–527.

Koenker, R. (2005), *Quantile regression*, Econometric Society Monograph Series, Cambridge University Press.

Koenker, R. & Bassett, G. (1978), 'Regression quantiles', *Econometrica* **46**, 33–50.

Kriegler, B. & Berk, R. (2010), 'Small area estimation of the homeless in los angeles: An application of cost-sensitive stochastic gradient boosting', *The Annals of Applied Statistics* **4**(3), 1234–1255.

Newey, W. & Powell, J. (1987), 'Asymmetric least squares estimation and testing', *Econometrica* **55**, 819–847.

Ridgeway, G. (2007), 'Generalized boosted regression models', *R package manual* . **URL:** *http://cran.r-project.org/web/packages/gbm/gbm.pdf*

Yao, Q. & Tong, H. (1996), 'Asymmetric least squares regression estimation: A nonparametric approach', *Journal of Nonparametric Statistics* **6**(2-3), 273–292.