A Boosted Tweedie Compound Poisson Model for Insurance Premium

Yi Yang[*], Wei Qian[†] and Hui Zou[‡]

July 26, 2015

Abstract

The Tweedie GLM is a widely used method for predicting insurance premiums. However, the linear model assumption can be too rigid for many applications. As a better alternative, a boosted Tweedie model is considered in this paper. We propose a TDboost estimator of pure premiums and use a profile likelihood approach to estimate the index and dispersion parameters. Our method is capable of fitting a flexible nonlinear Tweedie model and capturing complex interactions among predictors. A simulation study confirms the excellent prediction performance of our method. As an application, we apply our method to an auto insurance claim data and show that the new method is superior to the existing methods in the sense that it generates more accurate premium predictions, thus helping solve the adverse selection issue. We have implemented our method in a user-friendly R package that also includes a nice visualization tool for interpreting the fitted model.

1 Introduction

One of the most important problems in insurance business is to set the premium for the customers (policyholders). By insurance contracts, a large number of policyholders' individual losses are transformed into a more predictable, aggregate loss of the insurer. In a

[*]McGill University
[†]Rochester Institute of Technology
[‡]Corresponding author, zoux019@umn.edu, University of Minnesota

competitive market, it is advantageous for the insurer to charge a fair premium according to the expected loss of the policyholder. In personal car insurance, for instance, if an insurance company charges too much for old drivers and charges too little for young drivers, then the old drivers will switch to its competitors, and the remaining policies for the young drivers would be underpriced. This results in the *adverse selection* issue (Chiappori and Salanie, 2000; Dionne et al., 2001): the insurer loses profitable policies and is left with bad risks, resulting in economic loss both ways.

To appropriately set the premiums for the insurer's customers, one crucial task is to predict the size of actual (currently unforeseeable) claims. In this paper, we will focus on modeling claim loss, although other ingredients such as safety loadings, administrative costs, cost of capital, and profit are also important factors for setting the premium. One difficulty in modeling the claims is that the distribution is usually highly right-skewed, mixed with a point mass at zero. Such type of data cannot be transformed to normality by power transformation, and special treatment on zero claims is often required. As an example, Figure 1 shows the histogram of an auto insurance claim data (Yip and Yau, 2005), in which there are 6,290 policy records with zero claims and 4,006 policy records with positive losses.

The need for predictive models emerges from the fact that the expected loss is highly dependent on the characteristics of an individual policy such as age and annual income of the policyholder, population density of the policyholder's residential area, and age and model of the vehicle. Traditional methods used generalized linear models (GLM; Nelder and Wedderburn, 1972) for modeling the claim size (e.g. Renshaw, 1994; Haberman and Renshaw, 1996). However, all of these works performed their analyses on a subset of the policies, which have at least one claim. Alternative approaches have employed Tobit models by treating zero outcomes as censored below some cutoff points (Van de Ven and van Praag, 1981; Showers and Shotick, 1994), but these approaches rely on a normality assumption of the latent response. Alternatively, Jørgensen and de Souza (1994) and Smyth and Jørgensen (2002) used GLMs with a Tweedie distributed (Jørgensen, 1987, 1997) outcome to simultaneously model frequency and severity of insurance claims. They assume Poisson arrival of claims and gamma distributed amount for individual claims so that the size of the total claim amount follows a Tweedie compound Poisson distribution. Due to its ability to simultaneously model the zeros and the continuous positive outcomes, the Tweedie GLM

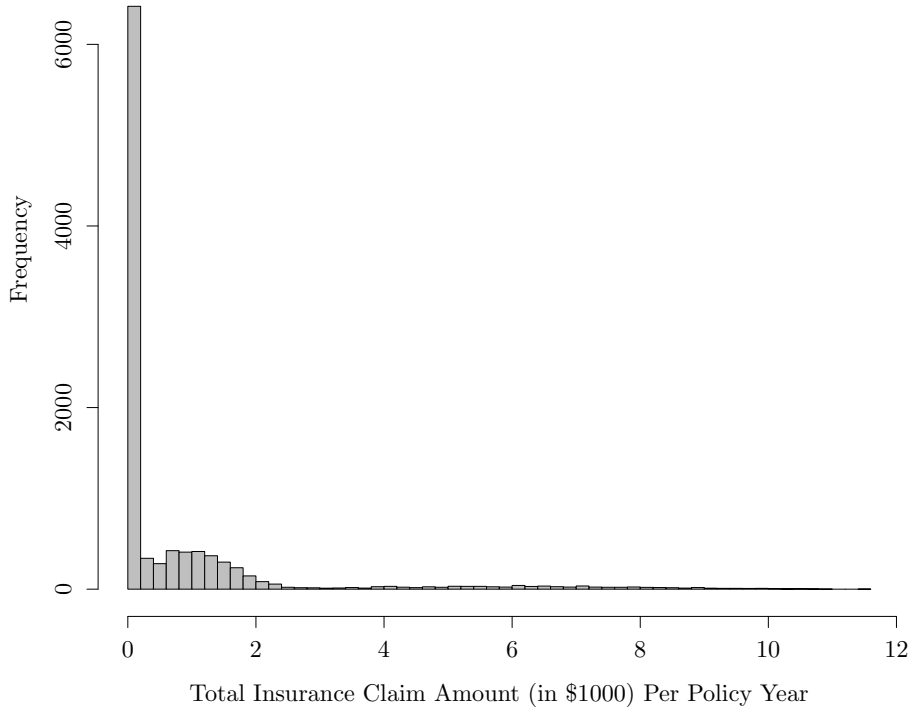

Figure 1: Histogram of the auto insurance claim data as analyzed in Yip and Yau (2005). It shows that there are 6290 policy records with zero total claims per policy year, while the remaining 4006 policy records have positive losses.

has been a widely used method in actuarial studies (Mildenhall, 1999; Murphy et al., 2000; Peters et al., 2008; Quijano Xacur et al., 2011).

Despite of the popularity of the Tweedie GLM, a major limitation is that the structure of the link function is restricted to a linear form, which can be too rigid for real applications. In auto insurance, for example, it is known that the risk does not monotonically decrease as age increases (Owsley et al., 1991; McCartt et al., 2003; Anstey et al., 2005). Although nonlinearity may be modeled by adding splines (Zhang, 2011), low-degree splines are often inadequate to capture the non-linearity in the data, while high-degree splines often result in the over-fitting issue that produces unstable estimates. Generalized additive models (GAM; Hastie and Tibshirani, 1990; Wood, 2006) overcome the restrictive linear assumption of GLMs, and can model the continuous variables by smooth functions estimated from data. The structure of the model, however, has to be determined *a priori*. That is, one has to specify the main effects and interaction effects to be used in the model. As a result, misspecification of non-ignorable effects is likely to adversely affect prediction accuracy.

In this paper, we aim to model the claim size by a nonparametric Tweedie compound Poisson model, and we propose a gradient tree boosting algorithm (TDBoost henceforth) to fit this model. Gradient boosting (Freund and Schapire, 1997, 1996) is one of the most successful machine learning algorithms for nonparametric regression and classification. Boosting adaptively combines a large number of relatively simple prediction models called *base learners* into an ensemble learner to achieve high prediction performance. The seminal work on the boosting algorithm called *AdaBoost* (Freund and Schapire, 1997, 1996) was originally proposed for classification problems. Later Breiman (1998) and Breiman (1999) pointed out an important connection between the AdaBoost algorithm and a functional gradient descent algorithm. Friedman et al. (2000) , Friedman (2001) and Hastie et al. (2009) developed a statistical view of boosting and proposed gradient boosting methods for both classification and regression. There is a large body of literature on boosting. We refer interested readers to Bühlmann and Hothorn (2007) for a comprehensive review of boosting algorithms.

The TDBoost model is motivated by the proven success of boosting in machine learning for classification and regression problems (Friedman, 2001, 2002; Hastie et al., 2009). Its advantages are threefold. First, the model structure of TDBoost is learned from data and not predetermined, thereby avoiding an explicit model specification. Non-linearities,

discontinuities, complex and higher order interactions are naturally incorporated into the model to reduce the potential modeling bias and to produce high predictive performance, which enables TDboost to serve as a benchmark model in scoring insurance policies, guiding pricing practice, and facilitating marketing efforts. Second, in contrast to other nonparametric statistical learning methods, TDboost can provide interpretable results, by means of the partial dependence plots, and relative importance of the predictors. Feature selection is performed as an integral part of the procedure. In addition, TDboost handles the predictor and response variables of any type without the need for transformation, and it is highly robust to outliers. Missing values in the predictors are managed almost without loss of information (Elith et al., 2008). All these properties make TDboost an attractive tool for insurance premium modeling.

The remainder of this paper is organized as follows. We briefly review the gradient boosting algorithm and the Tweedie compound Poisson model in Section 2 and Section 3, respectively. We present the main methodology development with implementation details in Section 4. In Section 5, we use simulation to show the high predictive accuracy of TDboost. As an application, we apply TDboost to analyze an auto insurance claim data in Section 6.

2 Gradient Boosting

To keep the paper self-contained, we briefly explain the general procedures for the gradient boosting. Let $\mathbf{x} = (x_1, \ldots, x_p)^\intercal$ be the p-dimensional predictor variables and y be the one-dimensional response variable. The goal is to estimate the optimal prediction function $\tilde{F}(\cdot)$ that maps $\mathbf{x}$ to y by minimizing the expected value of a loss function $\Psi(\cdot, \cdot)$ over the function class $\mathcal{F}$:

$$\tilde{F}(\cdot) = \underset{F(\cdot)\in\mathcal{F}}{\arg\min} E_{y,\mathbf{x}}[\Psi(y, F(\mathbf{x}))],$$

where Ψ is assumed to be differentiable with respect to F. Given the observed data $\{y_i, \mathbf{x}_i\}_{i=1}^n$, estimation of $\tilde{F}(\cdot)$ can be done by minimizing the empirical risk function

$$\min_{F(\cdot)\in\mathcal{F}} R_n(F) =: \min_{F(\cdot)\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \Psi(y_i, F(\mathbf{x}_i)). \tag{1}$$

For the gradient boosting, all candidate functions $F \in \mathcal{F}$ are assumed to be an ensemble of M base learners

$$F(\mathbf{x}) = F^{[0]} + \sum_{m=1}^{M} \beta^{[m]} h(\mathbf{x}; \boldsymbol{\xi}^{[m]}), \tag{2}$$

where $h(\mathbf{x}; \boldsymbol{\xi}^{[m]})$ usually belongs to a class of some simple functions of $\mathbf{x}$ called *base learners* (e.g., regression/decision tree) with the parameter $\boldsymbol{\xi}^{[m]}$ $(m = 1, 2, \cdots, M)$. $F^{[0]}$ is a constant scalar and $\beta^{[m]}$ is the expansion coefficient. Note that differing from the usual structure of an additive model, there is no restriction on the number of predictors to be included in each $h(\cdot)$, and consequently, high-order interactions can be easily considered using this setting.

A forward stagewise algorithm is adopted to approximate the minimizer of (1), which builds up the components $\beta^{[m]} h(\mathbf{x}; \boldsymbol{\xi}^{[m]})$ $(m = 1, 2, \ldots, M)$ sequentially through a gradient-descent-like approach. At each iteration stage m $(m = 1, 2, \ldots)$, suppose the current estimate for $\tilde{F}(\cdot)$ is $\hat{F}^{[m-1]}(\cdot)$. In principle, we want to update $\hat{F}^{[m-1]}(\cdot)$ to $\hat{F}^{[m]}(\cdot)$ along the negative gradient direction of $R_n(F)$. However, we only know the negative gradient at the training data points. Indeed, the negative gradient vector $(u_1^{[m]}, \ldots, u_n^{[m]})$ of $R_n(F)$ with respect to F at $\{F(\mathbf{x}_i) = \hat{F}^{[m-1]}(\mathbf{x}_i)\}_{i=1}^n$ can be written as

$$u_i^{[m]} = -\left.\frac{\partial R_n(F)}{\partial F(\mathbf{x}_i)}\right|_{F(\mathbf{x}_i)=\hat{F}^{[m-1]}(\mathbf{x}_i)},$$

but we cannot directly update $\hat{F}^{[m-1]}(\cdot)$ using $(u_1^{[m]}, \ldots, u_n^{[m]})$, as $u_i^{[m]}$ is only defined at $\mathbf{x}_i$ for $i = 1, \ldots, n$ and cannot make predictions based on new data not represented in the training set.

To solve this problem, the gradient boosting fits the negative gradient vector $(u_1^{[m]}, \ldots, u_n^{[m]})$ (as the working response) to $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ (as the predictor) to find a base learner $h(\mathbf{x}; \boldsymbol{\xi}^{[m]})$. Thus, the fitted $h(\mathbf{x}; \boldsymbol{\xi}^{[m]})$ can be viewed as an approximation of the negative gradient, and it can be evaluated on the entire space of $\mathbf{x}$. The expansion coefficient $\beta^{[m]}$ can then be determined by a line search

$$\beta^{[m]} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \Psi(y_i, \hat{F}^{[m-1]}(\mathbf{x}_i) + \beta h(\mathbf{x}; \boldsymbol{\xi}^{[m]})), \tag{3}$$

which is a reminiscence of the steepest descent method. Consequently, the estimation of

$\tilde{F}(\mathbf{x})$ for the next stage is

$$\hat{F}^{[m]}(\mathbf{x}) := \hat{F}^{[m-1]}(\mathbf{x}) + \nu\beta^{[m]}h(\mathbf{x};\boldsymbol{\xi}^{[m]}), \tag{4}$$

where $0 < \nu \leq 1$ is the shrinkage factor (Friedman, 2001) that controls the update step size. A small ν imposes more shrinkage while $\nu = 1$ gives complete negative gradient steps. Friedman (2001) has found that the shrinkage factor reduces over-fitting and improves the predictive accuracy.

3 Compound Poisson Distribution and Tweedie Model

In this section, we briefly introduce the compound Poisson distribution and Tweedie model, which is necessary for our methodology development. Let N be a Poisson random variable denoted by Pois(λ), and let $\tilde{Z}_d$'s be i.i.d. gamma random variables denoted by Gamma(α, γ) with mean $\alpha\gamma$ and variance $\alpha\gamma^2$. Assume N is independent of $\tilde{Z}_d$'s. Define a random variable Z by

$$Z = \begin{cases} 0 & \text{if } N = 0 \\ \tilde{Z}_1 + \tilde{Z}_2 + \cdots + \tilde{Z}_N & \text{if } N = 1, 2, \ldots \end{cases}. \tag{5}$$

Thus Z is the Poisson sum of independent Gamma random variables. The resulting distribution of Z is referred to as the compound Poisson distribution (Feller, 1968; Bar-Lev and Stramer, 1987; Jørgensen and de Souza, 1994; Smyth and Jørgensen, 2002), which is known to be closely connected to exponential dispersion models (EDM) (Jørgensen, 1987, 1997). Note that the distribution of Z has a probability mass at zero: $Pr(Z = 0) = \exp(-\lambda)$. Then based on that Z conditional on $N = j$ is Gamma($j\alpha, \gamma$), the distribution function of Z can be written as

$$\begin{aligned} f_Z(z|\lambda,\alpha,\gamma) &= Pr(N=0)d_0(z) + \sum_{j=1}^{\infty} Pr(N=j) f_{Z|N=j}(z) \\ &= \exp(-\lambda)d_0(z) + \sum_{j=1}^{\infty} \frac{\lambda^j e^{-\lambda}}{j!} \frac{z^{j\alpha-1}e^{-z/\gamma}}{\gamma^{j\alpha}\Gamma(j\alpha)}, \end{aligned}$$

where d_0 is the Dirac delta function at zero and $f_{Z|N=j}$ is the conditional density of Z given $N = j$. This gives the cumulant generating function of Z

$$\log M_Z(t) = \lambda\{(1-\gamma t)^{-\alpha} - 1\}. \tag{6}$$

Smyth (1996) pointed out that the compound Poisson distribution belongs to a special class of EDMs known as Tweedie models (Tweedie, 1984). The EDMs are defined by the form

$$f_Z(z|\theta,\phi) = a(z,\phi)\exp\left\{\frac{z\theta - \kappa(\theta)}{\phi}\right\}, \tag{7}$$

where $a(\cdot)$ is a normalizing function, $\kappa(\cdot)$ is called the cumulant function, and both $a(\cdot)$ and $\kappa(\cdot)$ are known. The parameter θ is in $\mathbb{R}$ and the dispersion parameter ϕ is in $\mathbb{R}^+$. EDMs have the property that the mean $E(Z) \equiv \mu = \dot{\kappa}(\theta)$ and the variance $\mathrm{Var}(Z) = \phi\ddot{\kappa}(\theta)$, where $\dot{\kappa}(\theta)$ and $\ddot{\kappa}(\theta)$ are the first and second derivatives of $\kappa(\theta)$, respectively. The cumulant generating function of EDMs is

$$\log M_Z(t) = \frac{1}{\phi}\{\kappa(\theta + t\phi) - \kappa(\theta)\}. \tag{8}$$

Tweedie models are special cases of the EDMs characterized by power mean-variance relationship $\mathrm{Var}(Z) = \phi\mu^\rho$ for some index parameter ρ. Such mean-variance relation gives

$$\theta = \begin{cases} \frac{\mu^{1-\rho}}{1-\rho}, & \rho \neq 1 \\ \log\mu, & \rho = 1 \end{cases}, \qquad \kappa(\theta) = \begin{cases} \frac{\mu^{2-\rho}}{2-\rho}, & \rho \neq 2 \\ \log\mu, & \rho = 2 \end{cases}. \tag{9}$$

One can show that the compound Poisson distribution belongs to the class of Tweedie models. Indeed, if we replace the parameters (λ, α, γ) in the cumulant function (6) by

$$\lambda = \frac{1}{\phi}\frac{\mu^{2-\rho}}{2-\rho}, \qquad \alpha = \frac{2-\rho}{\rho-1}, \qquad \gamma = \phi(\rho-1)\mu^{\rho-1}, \tag{10}$$

the cumulant function of the compound Poisson model has the form of a Tweedie model with $1 < \rho < 2$ and $\mu > 0$. As a result, for the rest of this paper, we only consider the model (5), and simply refer to (5) as the Tweedie model (or Tweedie compound Poisson model), denoted by $\mathrm{Tw}(\mu, \phi, \rho)$, where $1 < \rho < 2$ and $\mu > 0$.

It is straightforward to show that the log-likelihood of the Tweedie model is

$$\log f_Z(z|\mu,\phi,\rho) = \frac{1}{\phi}\left(z\frac{\mu^{1-\rho}}{1-\rho} - \frac{\mu^{2-\rho}}{2-\rho}\right) + \log a(z,\phi,\rho), \tag{11}$$

where the normalizing function $a(\cdot)$ can be written as

$$a(z,\phi,\rho) = \begin{cases} \frac{1}{z}\sum_{t=1}^{\infty} W_t(z,\phi,\rho) = \frac{1}{z}\sum_{t=1}^{\infty} \frac{z^{t\alpha}}{(\rho-1)^{t\alpha}\phi^{t(1+\alpha)}(2-\rho)^t t!\Gamma(t\alpha)} & \text{for } z>0 \\ 1 & \text{for } z=0 \end{cases},$$

and $\alpha = (2-\rho)/(\rho-1)$ and $\sum_{t=1}^{\infty} W_t$ is an example of Wright's generalized Bessel function (Tweedie, 1984).

One of the desirable properties of Tweedie models is that they are the only EDMs that are *scale invariant* (Jørgensen, 1997, Section 4.1.1): if Z is a Tweedie variable with mean μ and dispersion ϕ, then cZ follows the same distribution with mean $c\mu$ and dispersion $c^{2-\rho}\phi$. This property makes Tweedie distributions a good choice for modeling data with an arbitrary monetary unit.

4 Our proposal

In this section, we propose to integrate the Tweedie model to the tree-based gradient boosting algorithm to predict insurance claim size. Specifically, our discussion focuses on modeling the personal car insurance as an illustrating example (see Section 6 for a real data analysis), since our modeling strategy is easily extended to other lines of non-life insurance business.

Given an auto insurance policy i, let N_i be the number of claims (known as the claim frequency) and $\tilde{Z}_{d_i}$ be the size of each claim observed for $d_i = 1,\ldots,N_i$. Let w_i be the policy duration, that is, the length of time that the policy remains in force. Then $Z_i = \sum_{d_i=1}^{N_i} \tilde{Z}_{d_i}$ is the total claim amount. In the following, we are interested in modeling the ratio between the total claim and the duration $Y_i = Z_i/w_i$, a key quantity known as the pure premium (Ohlsson and Johansson, 2010).

Following the settings of the compound Poisson model, we assume N_i is Poisson distributed, and its mean $\lambda_i w_i$ has a multiplicative relation with the duration w_i, where λ_i is a policy-specific parameter representing the expected claim frequency under unit duration.

Conditional on N_i, assume Z_{d_i}'s ($d_i = 1, \ldots, N_i$) are i.i.d. Gamma(α, γ_i), where γ_i is a policy-specific parameter that determines claim severity, and α is a constant. Furthermore, we assume that under unit duration (i.e., $w_i = 1$), the mean-variance relation of a policy satisfies

$$Var(Y_i^*) = \phi[E(Y_i^*)]^\rho \tag{12}$$

for all policies, where Y_i^* is the pure premium under unit duration, ϕ is a constant, and $\rho = (\alpha + 2)/(\alpha + 1)$. Note that

$$\begin{aligned} \mu_i^* := E(Y_i^*) = E(E(Y_i^*|N_i)) = \lambda_i \alpha \gamma_i, \\ Var(Y_i^*) = E(Var(Y_i^*|N_i)) + Var(E(Y_i^*|N_i)) = \lambda_i \alpha \gamma_i^2 + \lambda_i \alpha^2 \gamma_i^2. \end{aligned}$$

Similarly, under duration w_i,

$$\begin{aligned} \mu_i := E(Y_i) = \frac{1}{w_i} E(Z_i) = \lambda_i \alpha \gamma_i, \\ Var(Y_i) = \frac{1}{w_i^2} Var(Z_i) = (\lambda_i \alpha \gamma_i^2 + \lambda_i \alpha^2 \gamma_i^2)/w_i. \end{aligned}$$

As a result, we can obtain the mean-variance relation for the pure premium Y_i that

$$Var(Y_i) = \frac{1}{w_i} Var(Y_i^*) = \frac{\phi}{w_i} (\mu_i^*)^\rho = \frac{\phi}{w_i} \mu_i^\rho, \tag{13}$$

where the second equation follows by (12). Consequently, the scale invariant property of Tweedie distribution and (13) implies that

$$Y_i \sim \mathrm{Tw}(\mu_i, \phi/w_i, \rho).$$

Under the aforementioned settings, consider a portfolio of policies $\{(y_i, \mathbf{x}_i, w_i)\}_{i=1}^n$ from n independent insurance contracts, where for the ith contract, y_i is the policy pure premium, $\mathbf{x}_i$ is a vector of explanatory variables that characterize the policyholder and the risk being insured (e.g. house, vehicle), and w_i is the duration. We then assume that the expected

pure premium μ_i is determined by a predictor function $F : \mathbb{R}^p \to \mathbb{R}$ of $\mathbf{x}_i$:

$$\log\{\mu_i\} = \log\{E(Y_i|\mathbf{x}_i)\} = F(\mathbf{x}_i). \tag{14}$$

In this paper, we do not impose a linear or other parametric form restriction on $F(\cdot)$. Given the flexibility of $F(\cdot)$, we call such setting as the boosted Tweedie model (as opposed to the Tweedie GLM). Given $\{(y_i, \mathbf{x}_i, w_i)\}_{i=1}^n$, the log-likelihood function can be written as

$$\begin{aligned} \ell(F(\cdot), \phi, \rho|\{y_i, \mathbf{x}_i, w_i\}_{i=1}^n) &= \sum_{i=1}^n \log f_Y(y_i|\mu_i, \phi/w_i, \rho), \\ &= \sum_{i=1}^n \frac{w_i}{\phi}\left(y_i\frac{\mu_i^{1-\rho}}{1-\rho} - \frac{\mu_i^{2-\rho}}{2-\rho}\right) + \log a(y_i, \phi/w_i, \rho). \end{aligned} \tag{15}$$

4.1 Estimating $F(\cdot)$ via TDboost

We estimate the predictor function $F(\cdot)$ by integrating the boosted Tweedie model into the tree-based gradient boosting algorithm. To develop the idea, we assume that ϕ and ρ are given for the time being. The joint estimation of $F(\cdot)$, ϕ and ρ will be studied in Section 4.2.

Given ρ and ϕ, we replace the general objective function in (1) by the negative log-likelihood derived in (15), and target the minimizer function $F^*(\cdot)$ over a class $\mathcal{F}$ of base learner functions in the form of (2). That is, we intend to estimate

$$F^*(\mathbf{x}) = \underset{F\in\mathcal{F}}{\operatorname{argmin}} \big\{ -\ell(F(\cdot), \phi, \rho|\{y_i, \mathbf{x}_i, w_i\}_{i=1}^n)\big\} = \underset{F\in\mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^n \Psi(y_i, F(\mathbf{x}_i)|\rho), \tag{16}$$

where

$$\Psi(y_i, F(\mathbf{x}_i)|\rho) = w_i\left\{ -\frac{y_i \exp[(1-\rho)F(\mathbf{x}_i)]}{1-\rho} + \frac{\exp[(2-\rho)F(\mathbf{x}_i)]}{2-\rho}\right\}.$$

Note that in contrast to (16), the function class targeted by Tweedie GLM (Smyth, 1996) is restricted to a collection of linear functions of $\mathbf{x}$.

We propose to apply the forward stagewise algorithm described in Section 2 for solving (16). The initial estimate of $F^*(\cdot)$ is chosen as a constant function that minimizes the

negative log-likelihood:

$$\begin{aligned}\hat{F}^{[0]} &= \underset{\eta}{\operatorname{argmin}} \sum_{i=1}^{n} \Psi(y_i, \eta \mid \rho) \\ &= \log\left(\frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i}\right).\end{aligned}$$

This corresponds to the best estimate of F without any covariates. Let $\hat{F}^{[m-1]}$ be the current estimate before the mth iteration. At the mth step, we fit a base learner $h(\mathbf{x}; \boldsymbol{\xi}^{[m]})$ via

$$\widehat{\boldsymbol{\xi}}^{[m]} = \underset{\boldsymbol{\xi}^{[m]}}{\operatorname{argmin}} \sum_{i=1}^{n} [u_i^{[m]} - h(\mathbf{x}_i; \boldsymbol{\xi}^{[m]})]^2, \tag{17}$$

where $(u_1^{[m]}, \ldots, u_n^{[m]})$ is the current negative gradient of $\Psi(\cdot \mid \rho)$, i.e.,

$$\begin{aligned} u_i^{[m]} &= -\frac{\partial \Psi(y_i, F(\mathbf{x}_i) \mid \rho)}{\partial F(\mathbf{x}_i)}\bigg|_{F(\mathbf{x}_i)=\hat{F}^{[m-1]}(\mathbf{x}_i)} && (18) \\ &= w_i\big\{ - y_i \exp[(1-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)] + \exp[(2-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)]\big\}, && (19)\end{aligned}$$

and use an L-terminal node regression tree

$$h(\mathbf{x}; \boldsymbol{\xi}^{[m]}) = \sum_{l=1}^{L} u_l^{[m]} I(\mathbf{x} \in R_l^{[m]}) \tag{20}$$

with parameters $\boldsymbol{\xi}^{[m]} = \{R_l^{[m]}, u_l^{[m]}\}_{l=1}^{L}$ as the base learner. To find $R_l^{[m]}$ and $u_l^{[m]}$, we use a fast top-down "best-fit" algorithm with a least squares splitting criterion (Friedman et al., 2000) to find the splitting variables and corresponding split locations that determine the fitted terminal regions $\{\widehat{R}_l^{[m]}\}_{l=1}^{L}$. Note that estimating the $R_l^{[m]}$ entails estimating the $u_l^{[m]}$ as the mean falling in each region:

$$\bar{u}_l^{[m]} = \text{mean}_{i:\mathbf{x}_i \in \widehat{R}_l^{[m]}}(u_i^{[m]}) \qquad l = 1, \ldots, L.$$

Once the base learner $h(\mathbf{x}; \boldsymbol{\xi}^{[m]})$ has been estimated, the optimal value of the expansion

coefficient $\beta^{[m]}$ is determined by a line search

$$\begin{aligned} \beta^{[m]} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \Psi(y_i, \hat{F}^{[m-1]}(\mathbf{x}_i) + \beta h(\mathbf{x}_i; \widehat{\boldsymbol{\xi}}^{[m]}) \mid \rho) \\ &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \Psi(y_i, \hat{F}^{[m-1]}(\mathbf{x}_i) + \beta \sum_{l=1}^{L} \bar{u}_l^{[m]} I(\mathbf{x}_i \in \widehat{R}_l^{[m]}) \mid \rho). \end{aligned} \tag{21}$$

The regression tree (20) predicts a constant value $\bar{u}_l^{[m]}$ within each region $\widehat{R}_l^{[m]}$, so we can solve (21) by a separate line search performed within each respective region $\widehat{R}_l^{[m]}$. The problem (21) reduces to finding a best constant $\eta_l^{[m]}$ to improve the current estimate in each region $\widehat{R}_l^{[m]}$ based on the following criterion:

$$\hat{\eta}_l^{[m]} = \underset{\eta}{\operatorname{argmin}} \sum_{i:\mathbf{x}_i \in \widehat{R}_l^{[m]}} \Psi(y_i, \hat{F}^{[m-1]}(\mathbf{x}_i) + \eta \mid \rho), \qquad l = 1, \ldots, L, \tag{22}$$

where the solution is given by

$$\hat{\eta}_l^{[m]} = \log \left\{ \frac{\sum_{i:\mathbf{x}_i \in \widehat{R}_l^{[m]}} w_i y_i \exp[(1-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)]}{\sum_{i:\mathbf{x}_i \in \widehat{R}_l^{[m]}} w_i \exp[(2-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)]} \right\}, \qquad l = 1, \ldots, L. \tag{23}$$

Having found the parameters $\{\hat{\eta}_l^{[m]}\}_{l=1}^{L}$, we then update the current estimate $\hat{F}^{[m-1]}(\mathbf{x})$ in each corresponding region

$$\hat{F}^{[m]}(\mathbf{x}) = \hat{F}^{[m-1]}(\mathbf{x}) + \nu \hat{\eta}_l^{[m]} I(\mathbf{x} \in \widehat{R}_l^{[m]}), \qquad l = 1, \ldots, L, \tag{24}$$

where $0 < \nu \leq 1$ is the shrinkage factor. Following (Friedman, 2001), we set $\nu = 0.005$ in our implementation. More discussion on the choice of tuning parameters are in Section 4.4.

In summary, the complete TDBoost algorithm is shown in Algorithm 1. The boosting step is repeated M times and we report $\hat{F}^{[M]}(\mathbf{x})$ as the final estimate.

4.2 Estimating (ρ, ϕ) via profile likelihood

Following Smyth (1996) and Dunn and Smyth (2005), we use the profile likelihood to estimate the dispersion ϕ and the index parameter ρ, which jointly determine the mean-variance

Algorithm 1 TDboost

1. Initialize $\hat{F}^{[0]}$

$$\hat{F}^{[0]} = \log\left(\frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i}\right).$$

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

2.(a) Compute the negative gradient $\{u_i^{[m]}\}_{i=1}^{n}$

$$u_i^{[m]} = w_i\big\{-y_i \exp[(1-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)] + \exp[(2-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)]\big\} \qquad i = 1, \ldots, n.$$

2.(b) Fit the negative gradient vector $\{u_i^{[m]}\}_{i=1}^{n}$ to $\mathbf{x}_1, \ldots, \mathbf{x}_n$ by an L-terminal node regression tree, giving us the partitions $\{\widehat{R}_l^{[m]}\}_{l=1}^{L}$.

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

$$\hat{\eta}_l^{[m]} = \log\left\{\frac{\sum_{i:\mathbf{x}_i \in \widehat{R}_l^{[m]}} w_i y_i \exp[(1-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)]}{\sum_{i:\mathbf{x}_i \in \widehat{R}_l^{[m]}} w_i \exp[(2-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)]}\right\}.$$

2.(d) Update $\hat{F}^{[m]}(\mathbf{x})$ for each region $\widehat{R}_l^{[m]}$, $l = 1, 2, \ldots, L$

$$\hat{F}^{[m]}(\mathbf{x}) = \hat{F}^{[m-1]}(\mathbf{x}) + \nu\hat{\eta}_l^{[m]} I(\mathbf{x} \in \widehat{R}_l^{[m]}) \qquad l = 1, 2, \ldots, L.$$

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

relation $Var(Y_i) = \phi\mu_i^{\rho}/w_i$ of the pure premium. We exploit the fact that in Tweedie models the estimation of μ depends only on ρ: given a fixed ρ, the mean estimate $\mu^*(\rho)$ can be solved in (16) without knowing ϕ. Then conditional on this ρ and the corresponding $\mu^*(\rho)$, we maximize the log-likelihood function with respect to ϕ by

$$\phi^*(\rho) = \underset{\phi}{\operatorname{argmax}}\big\{\ell(\mu^*(\rho), \phi, \rho)\big\}, \tag{25}$$

which is a univariate optimization problem that can be solved using a combination of golden section search and successive parabolic interpolation (Brent, 2013). In such a way, we have determined the corresponding $(\mu^*(\rho), \phi^*(\rho))$ for each fixed ρ. Then we acquire the estimate of ρ by maximizing the profile likelihood with respect to 50 equally spaced values $\{\rho_1, \ldots, \rho_{50}\}$ on $(0, 1)$:

$$\rho^* = \underset{\rho\in\{\rho_1,\ldots,\rho_{50}\}}{\operatorname{argmax}} \big\{\ell(\mu^*(\rho), \phi^*(\rho), \rho)\big\}. \tag{26}$$

Finally, we apply ρ^* in (16) and (25) to obtain the corresponding estimates $\mu^*(\rho^*)$ and $\phi^*(\rho^*)$.

There are some computational issues, which must be taken care of when evaluating the log-likelihood functions in (25) and (26): since in general there are no closed forms for Tweedie densities, in likelihood evaluation one must deal with an infinite summation in the normalizing function $a(y, \phi, \rho) = \frac{1}{y}\sum_{t=1}^{\infty} W_t$. For numerical evaluation of Tweedie densities, Dunn and Smyth (2005) proposed a series expansions approach, which sums an infinite series arising from a Taylor expansion of the characteristic function. Alternatively, Dunn and Smyth (2008) developed a Fourier inversion approach, which consists of an inversion of the characteristic function based on numerical integration methods for oscillating functions. These two numerical methods turn out to be complementary since each has advantages under a certain situation: when only considering the case $1 < \rho < 2$, the series approach performs very well for small y but gradually loses computational efficiency as y increases, whereas the inversion approach performs very well for large y but gradually fails to provide accurate results as y decreases. Hence the inversion approach is preferred for large y and the series approach for small y. Dunn and Smyth (2008) provided a simple guideline to choose between the two methods. In this paper we use their R package "tweedie" (available at `http://cran.r-project.org/web/packages/tweedie/index.html`) for evaluating Tweedie densities in our profile likelihood computation. For further details regarding their algorithms, the reader

may refer to Dunn and Smyth (2005, 2008).

4.3 Model interpretation

Compared to other nonparametric statistical learning methods such as neural networks and kernel machines, our new estimator provides interpretable results. In this section, we discuss some ways for model interpretation after fitting the boosted Tweedie model.

4.3.1 Marginal effects of predictors

The main effects and interaction effects of the variables in the boosted Tweedie model can be extracted easily. In our estimate we can control the order of interactions by choosing the tree size L (the number of terminal nodes) and the number p of predictors. A tree with L terminal nodes produces a function approximation of p predictors with interaction order of at most $\min(L-1, p)$. For example, a stump ($L = 2$) produces an additive TDboost model with only the main effects of the predictors, since it is a function based on a single splitting variable in each tree. Setting $L = 3$ allows both main effects and second order interactions.

Following Friedman (2001) we use the so-called partial dependence plots to visualize the main effects and interaction effects. Given the training data $\{y_i, \mathbf{x}_i\}_{i=1}^n$, with a p-dimensional input vector $\mathbf{x} = [x_1, x_2, \ldots, x_p]'$, let $\mathbf{z}_s$ be a subset of size s, such that $\mathbf{z}_s = \{z_1, \ldots, z_s\} \subset \{x_1, \ldots, x_p\}$. For example, to study the main effect of the variable j, we set the subset $\mathbf{z}_s = \{z_j\}$, and to study the second order interaction of variables i and j, we set $\mathbf{z}_s = \{z_i, z_j\}$. Let $\mathbf{z}_{\setminus s}$ be the complement set of $\mathbf{z}_s$, such that $\mathbf{z}_{\setminus s} \cup \mathbf{z}_s = \{x_1, \ldots, x_p\}$. Let the prediction $\hat{F}(\mathbf{z}_s|\mathbf{z}_{\setminus s})$ be a function of the subset $\mathbf{z}_s$ conditioned on specific values of $\mathbf{z}_{\setminus s}$. The partial dependence of $\hat{F}(\mathbf{x})$ on $\mathbf{z}_s$ then can be formulated as $\hat{F}(\mathbf{z}_s|\mathbf{z}_{\setminus s})$ averaged over the marginal density of the complement subset $\mathbf{z}_{\setminus s}$

$$\hat{F}_s(\mathbf{z}_s) = \int \hat{F}(\mathbf{z}_s|\mathbf{z}_{\setminus s}) p_{\setminus s}(\mathbf{z}_{\setminus s}) d\mathbf{z}_{\setminus s}, \tag{27}$$

where $p_{\setminus s}(\mathbf{z}_{\setminus s}) = \int p(\mathbf{x}) d\mathbf{z}_s$ is the marginal density of $\mathbf{z}_{\setminus s}$. We estimate (27) by

$$\bar{F}_s(\mathbf{z}_s) = \frac{1}{n} \sum_{i=1}^{n} \hat{F}(\mathbf{z}_s|\mathbf{z}_{\setminus s,i}), \tag{28}$$

where $\{\mathbf{z}_{\setminus s,i}\}_{i=1}^{n}$ are evaluated at the training data. We then plot $\bar{F}_s(\mathbf{z}_s)$ against $\mathbf{z}_s$. We have included the partial dependence plot function in our R package "TDboost". We will demonstrate this functionality in Section 6.

4.3.2 Variable importance

In many applications identifying relevant predictors of the model in the context of tree-based ensemble methods is of interest. The TDboost model defines a variable importance measure for each candidate predictor X_j in the set $X = \{X_1, \ldots, X_p\}$ in terms of prediction/explanation of the response Y. The major advantage of this variable selection procedure, as compared to univariate screening methods, is that the approach considers the impact of each individual predictor as well as multivariate interactions among predictors simultaneously.

We start by defining the variable importance (VI henceforth) measure in the context of a single tree. First introduced by Breiman et al. (1984), the VI measure $\mathcal{I}_{X_j}(T_m)$ of the variable X_j in a single tree T_m is defined as the total heterogeneity reduction of the response variable Y produced by X_j, which can be estimated by adding up all the decreases in the squared error reductions $\hat{\delta}_l$ obtained in all $L-1$ internal nodes when X_j is chosen as the splitting variable. Denote $v(X_j) = l$ the event that X_j is selected as the splitting variable in the internal node l, and let $I_{jl} = I(v(X_j) = l)$. Then

$$\mathcal{I}_{X_j}(T_m) = \sum_{l=1}^{L-1} \hat{\delta}_l I_{jl}, \tag{29}$$

where $\hat{\delta}_l$ is defined as the squared error difference between the constant fit and the two sub-region fits (the sub-region fits are achieved by splitting the region associated with the internal node l into the left and right regions). Friedman (2001) extended the VI measure $\mathcal{I}_{X_j}$ for the boosting model with a combination of M regression trees, by averaging (29) over $\{T_1, \ldots, T_M\}$:

$$\mathcal{I}_{X_j} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_{X_j}(T_m). \tag{30}$$

Despite of the wide use of the VI measure, Breiman et al. (1984), White and Liu (1994) and Kononenko (1995) among others have pointed out that the VI measures (29) and (30)

are biased: even if X_j is a non-informative variable to Y (not correlated to Y), X_j may still be selected as a splitting variable, hence the VI measure of X_j is non-zero by Equation (30). Following Sandri and Zuccolotto (2008) and Sandri and Zuccolotto (2010) to avoid the variable selection bias, we compute an adjusted VI measure for each explanatory variable by permutating each X_j:

(1) For $s = 1, \ldots, S$, repeat steps (2)–(4).

(2) Generate a matrix $\mathbf{z}^s$ by randomly permutating (without replacement) the n rows of the design matrix $\mathbf{x}$, while keeping the order of columns unchanged.

(3) Create an $n \times 2p$ matrix $\tilde{\mathbf{x}}^s = [\mathbf{x}, \mathbf{z}^s]$ by binding $\mathbf{z}^s$ with $\mathbf{x}$ matrix by column.

(4) Use the data $\{y, \tilde{\mathbf{x}}^s\}$ to fit the model, and compute VI measures $\mathcal{I}^s_{X_j}$ for X_j and $\mathcal{I}^s_{Z^s_j}$ for Z^s_j, where Z^s_j (jth column of Z^s) is the pseudo-predictor corresponding to X_j.

(5) Compute the VI measure $\overline{\mathcal{I}}_{X_j}$ as the average of $\mathcal{I}^s_{X_j}$ and the baseline $\overline{\mathcal{I}}_{Z_j}$ as the average of $\mathcal{I}^s_{Z^s_j}$

$$\overline{\mathcal{I}}_{X_j} = \frac{1}{S}\sum_{s=1}^{S} \mathcal{I}^s_{X_j} \qquad \overline{\mathcal{I}}_{Z_j} = \frac{1}{S}\sum_{s=1}^{S} \mathcal{I}^s_{Z^s_j}. \tag{31}$$

(6) Report the adjusted VI measure as $\mathcal{I}^{\text{adj}}_{X_j} = \overline{\mathcal{I}}_{X_j} - \overline{\mathcal{I}}_{Z_j}$ for the variable X_j.

The basic idea of the above algorithm is the following: the permutation breaks the association between the response variable Y and each pseudo-predictor Z^s_j, but still preserves the association between Z^s_j and Z^s_k ($k \neq j$); since Z^s_j is re-shuffled from X_j, Z^s_j has the same number of possible splits as the corresponding predictor X_j and has approximately the same probability of being selected in split nodes. Therefore, $\overline{\mathcal{I}}_{Z_j}$ can be viewed as a bias approximation of the importance of X_j.

4.4 Implementation

We have implemented our proposed method in an R package "TDboost", which is publicly available from the Comprehensive R Archive Network at `http://cran.r-project.org/web/packages/TDboost/index.html`. Here, we discuss the choice of three meta parameters

in Algorithm 1: L (the size of the trees), ν (the shrinkage factor) and M (the number of boosting steps).

To avoid over-fitting and improve out-of-sample predictions, the boosting procedure can be regularized by limiting the number of boosting iterations M (early stopping; Zhang and Yu, 2005) and the shrinkage factor ν. Empirical evidence (Friedman, 2001; Bühlmann and Hothorn, 2007; Ridgeway, 2007; Elith et al., 2008) showed that the predictive accuracy is almost always better with a smaller shrinkage factor at the cost of more computing time. However, smaller values of ν usually requires a larger number of boosting iterations M and hence induces more computing time (Friedman, 2001). We choose a "sufficiently small" $\nu = 0.005$ throughout and determine M by the data.

The value L should reflect the true interaction order in the underlying model, but we almost never have such prior knowledge. Therefore we choose the optimal M and L using K-fold cross validation, starting with a fixed value of L. The data are split into K roughly equal-sized folds. Let an index function $\pi(i) : \{1, \ldots, n\} \mapsto \{1, \ldots, K\}$ indicate the fold to which observation i is allocated. Each time, we remove the kth fold of the data ($k = 1, 2, \ldots, K$), and train the model using the remaining $K - 1$ folds. Denoting by $\hat{F}_{-k}^{[M]}(\mathbf{x})$ the resulting model, we compute the validation loss by predicting on each kth fold of the data removed:

$$\mathrm{CV}(M, L) = \frac{1}{n} \sum_{i=1}^{n} \Psi(y_i, \hat{F}_{-\pi(i)}^{[M]}(\mathbf{x}_i; L) \mid \rho). \tag{32}$$

We select the optimal M at which the minimum validation loss is reached

$$\widehat{M}_L = \underset{M}{\operatorname{argmin}} \, \mathrm{CV}(M, L).$$

If we need to select L too, then we repeat the whole process for several L (e.g. $L = 2, 3, 4, 5$) and choose the one with the smallest minimum generalization error

$$\widehat{L} = \underset{L}{\operatorname{argmin}} \, \mathrm{CV}(L, \widehat{M}_L).$$

For a given ν, fitting trees with higher L leads to smaller M being required to reach the minimum error.

5 Simulation Studies

In this section, we compare TDboost with the Tweedie GLM model (TGLM: Jørgensen and de Souza, 1994) and the Tweedie GAM model in terms of the function estimation performance. The Tweedie GAM model is proposed by Wood (2001), which is based on a penalized regression spline approach with automatic smoothness selection. There is an R package "MGCV" accompanying the work, available at `http://cran.r-project.org/web/packages/mgcv/index.html`. In all numerical examples below using the TDboost model, five-fold cross validation is adopted for selecting the optimal (M, L) pair, while the shrinkage factor ν is set to its default value of 0.005.

5.1 Case I

In this simulation study, we demonstrate that TDboost is well suited to fit target functions that are non-linear or involve complex interactions. We consider two true target functions:

- **Model 1** (Discontinuous function): The target function is discontinuous as defined by $F(x) = 0.5I(x > 0.5)$. We assume $x \sim \text{Unif}(0, 1)$, and $y \sim \text{Tw}(\mu, \phi, \rho)$ with $\rho = 1.5$ and $\phi = 0.5$.

- **Model 2** (Complex interaction): The target function has two hills and two valleys.

$$F(x_1, x_2) = e^{-5(1-x_1)^2+x_2^2} + e^{-5x_1^2+(1-x_2)^2},$$

  which corresponds to a common scenario where the effect of one variable changes depending on the effect of another. We assume $x_1, x_2 \sim \text{Unif}(0, 1)$, and $y \sim \text{Tw}(\mu, \phi, \rho)$ with $\rho = 1.5$ and $\phi = 0.5$.

We generate $n = 1000$ observations for training and $n' = 1000$ for testing, and fit the training data using TDboost, MGCV, and TGLM. Since the true target functions are known, we consider the mean absolute deviation (MAD) as performance criteria,

$$\text{MAD} = \frac{1}{n'} \sum_{i=1}^{n'} |F(\mathbf{x}_i) - \hat{F}(\mathbf{x}_i)|,$$

Model	TGLM	MGCV	TDboost
1	0.1102 (0.0006)	0.0752 (0.0016)	0.0595 (0.0021)
2	0.3516 (0.0009)	0.2511 (0.0004)	0.1034 (0.0008)

Table 1: The averaged MADs and the corresponding standard errors based on 100 independent replications.

where both the true predictor function $F(\mathbf{x}_i)$ and the predicted function $\hat{F}(\mathbf{x}_i)$ are evaluated on the test set. The resulting MADs on the testing data are reported in Table 1, which are averaged over 100 independent replications. The fitted functions from Model 2 are plotted in Figure 2. In both cases, we find that TDboost outperforms TGLM and MGCV in terms of the ability to recover the true functions and gives the smallest prediction errors.

5.2 Case II

The idea is to see the performance of the TDboost estimator and MGCV estimator on a variety of very complicated, randomly generated predictor functions, and study how training sample sizes, distribution settings and other characteristics of problems affect final performance of the two methods. We use the "random function generator" (RFG) model by Friedman (2001) in our simulation. The true target function F is randomly generated as a linear expansion of functions $\{g_k\}_{k=1}^{20}$:

$$F(\mathbf{x}) = \sum_{k=1}^{20} b_k g_k(\mathbf{z}_k). \tag{33}$$

Here each coefficient b_k is a uniform random variable from $\mathrm{U}[-1, 1]$. Each $g_k(\mathbf{z}_k)$ is a function of $\mathbf{z}_k$, where $\mathbf{z}_k$ is defined as a p_k-sized subset of the ten-dimensional variable $\mathbf{x}$ in the form

$$\mathbf{z}_k = \{x_{\psi_k(j)}\}_{j=1}^{p_k}, \tag{34}$$

where each ψ_k is an independent permutation of the integers $\{1, \ldots, p\}$. The size p_k is randomly selected by $\min(\lfloor 2.5 + r_k \rfloor, p)$, where r_k is generated from an exponential distribution with mean 2. Hence the expected order of interactions presented in each $g_k(\mathbf{z}_k)$ is between

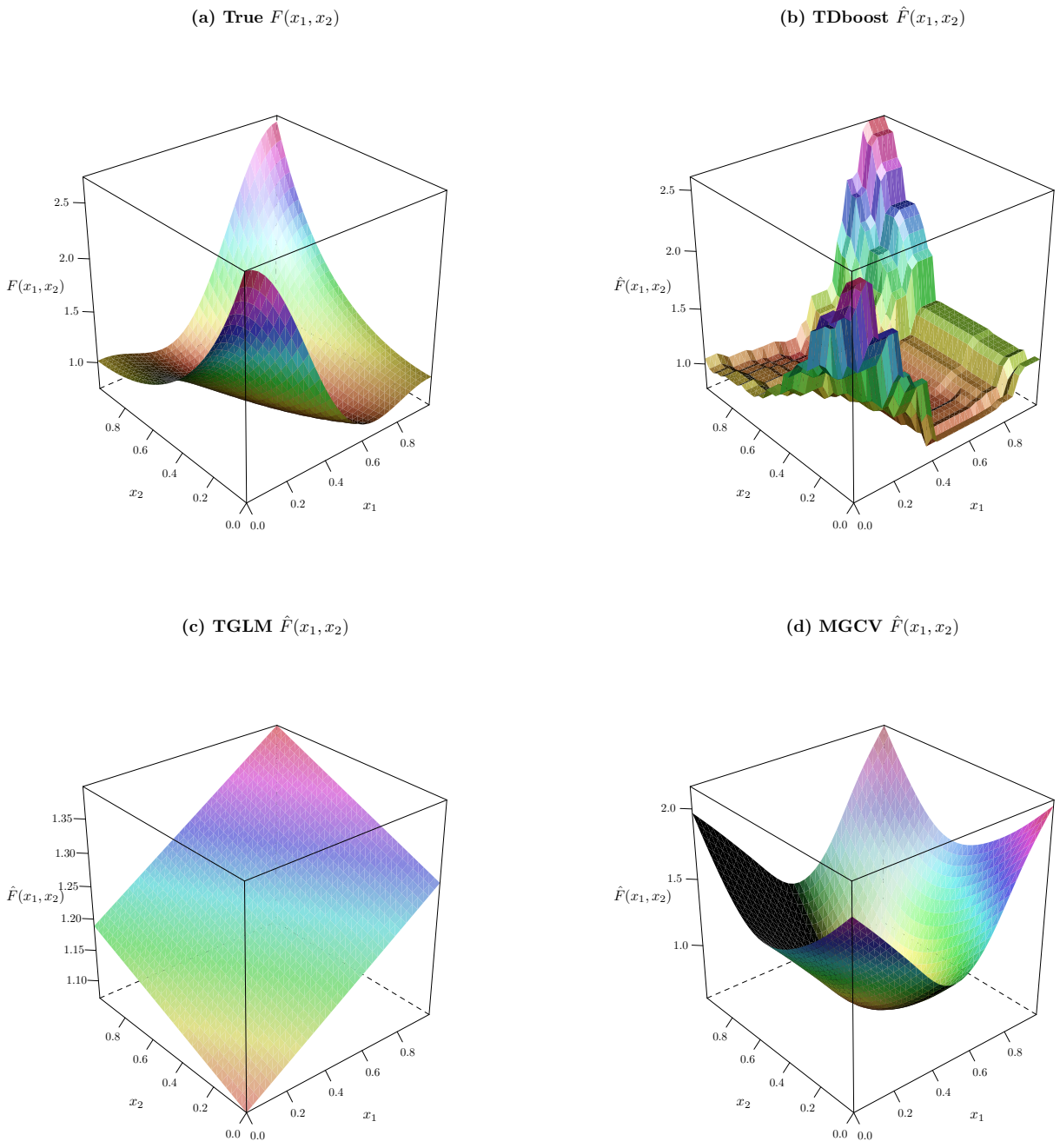

Figure 2: Fitted curves that recover the target function defined in Model 2. The top left figure shows the true target function. The top right, bottom left, and bottom right figures show the predictions on the testing data from TDboost, TGLM, and MGCV, respectively.

four and five. Each function $g_k(\mathbf{z}_k)$ is a p_k-dimensional Gaussian function:

$$g_k(\mathbf{z}_k) = \exp\Big\{ -\frac{1}{2}(\mathbf{z}_k - \mathbf{u}_k)^T \mathbf{V}_k (\mathbf{z}_k - \mathbf{u}_k) \Big\}, \tag{35}$$

where each mean vector $\mathbf{u}_k$ is randomly generated from $\mathrm{N}(0, \mathbf{I}_{p_k})$. The $p_k \times p_k$ covariance matrix $\mathbf{V}_k$ is defined by

$$\mathbf{V}_k = \mathbf{U}_k \mathbf{D}_k \mathbf{U}_k^T, \tag{36}$$

where $\mathbf{U}_k$ is a random orthonormal matrix, $\mathbf{D}_k = diag\{d_k[1], \ldots, d_k[p_k]\}$, and the square root of each diagonal element $\sqrt{d_k[j]}$ is a uniform random variable from $\mathrm{Unif}[0.1, 2.0]$. We generate data $\{y_i, \mathbf{x}_i\}_{i=1}^n$ according to

$$y_i \sim \mathrm{Tw}(\mu_i, \phi, \rho), \quad \mathbf{x}_i \sim \mathrm{N}(0, \mathbf{I}_p), \quad i = 1, \ldots, n, \tag{37}$$

where $\mu_i = \exp\{F(\mathbf{x}_i)\}$.

Setting I: when the index is known

Firstly, we study the situation that the true index parameter ρ is known when fitting models. We generate data according to the RFG model with index parameter $\tilde{\rho} = 1.5$ and the dispersion parameter $\tilde{\phi} = 1$ in the true model. We set the number of predictors to be $p = 10$ and generate $n \in \{1000, 2000, 5000\}$ observations as training sets, on which both MGCV and TDboost are fitted with ρ specified to be the true value 1.5. An additional test set of $n' = 5000$ observations was generated for evaluating the performance of the final estimate.

Figure 3 shows simulation results for comparing the estimation performance of MGCV and TDboost, when varying the training sample size. The empirical distributions of the MADs shown as box-plots are based on 100 independent replications. We can see that in all of the cases, TDboost outperforms MGCV in terms of prediction accuracy.

We also test estimation performance on μ when the index parameter ρ is misspecified, that is, we use a guess value ρ differing from the true value $\tilde{\rho}$ when fitting the TDboost model. Because μ is statistically orthogonal to ϕ and ρ, meaning that the off-diagonal elements of the Fisher information matrix are zero (Jørgensen, 1997), we expect $\hat{\mu}$ will vary very slowly as ρ changes. Indeed, using the previous simulation data with the true value $\tilde{\rho} = 1.5$ and

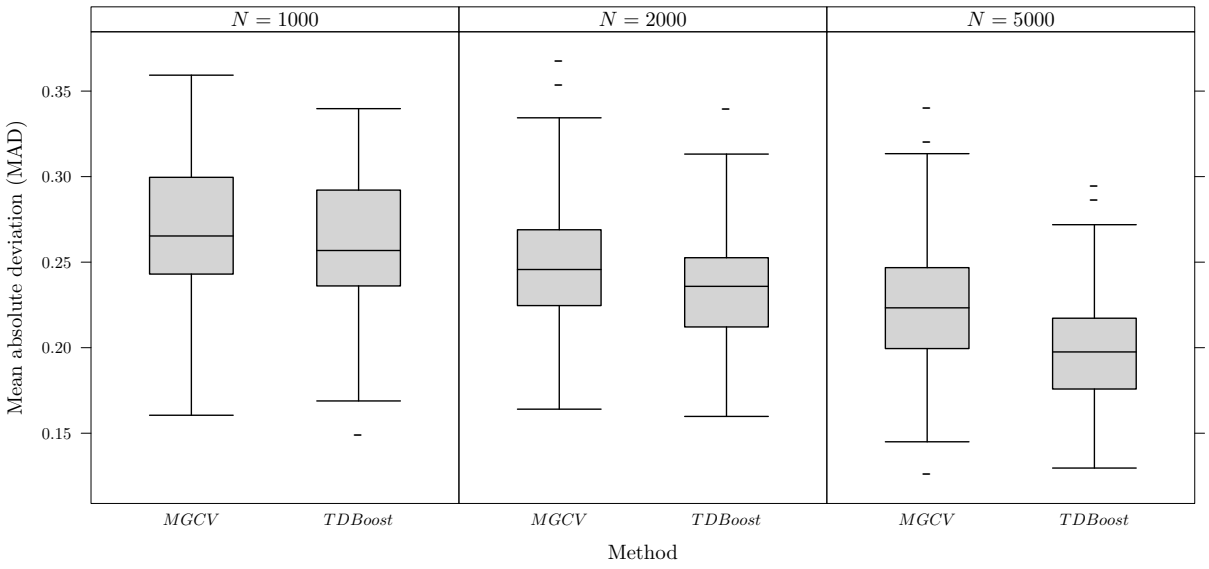

Figure 3: Simulation results for Setting I: compare the estimation performance of **MGCV** and **TDboost** when varying the training sample size and the dispersion parameter in the true model. Box-plots display empirical distributions of the MADs based on 100 independent replications.

$\tilde{\phi} = 1$, we fitted TDboost models with nine guess values of $\rho \in \{1.1, 1.2, \ldots, 1.9\}$. The resulting MADs are displayed in Figure 4, which shows the choice of the value ρ has almost no significant effect on estimation accuracy of μ.

Setting II: using the estimated index

Next we study the situation that the true index parameter ρ is unknown, and we use the estimated ρ obtained from the profile likelihood procedure discussed in Section 4.2 for fitting the model. The same data generation scheme is adopted as in Setting I, except now both MGCV and TDboost are fitted with ρ estimated by maximizing the profile likelihood. Figure 5 shows simulation results for comparing the estimation performance of MGCV and TDboost in such setting. We can see that the results have no significant difference to the results of Setting I: TDboost still outperforms MGCV in terms of prediction accuracy when using the estimated ρ instead of the true value.

Lastly, we demonstrate our results from the estimation of the dispersion ϕ and the index ρ by using the profile likelihood. A total number of 200 sets of training samples are randomly

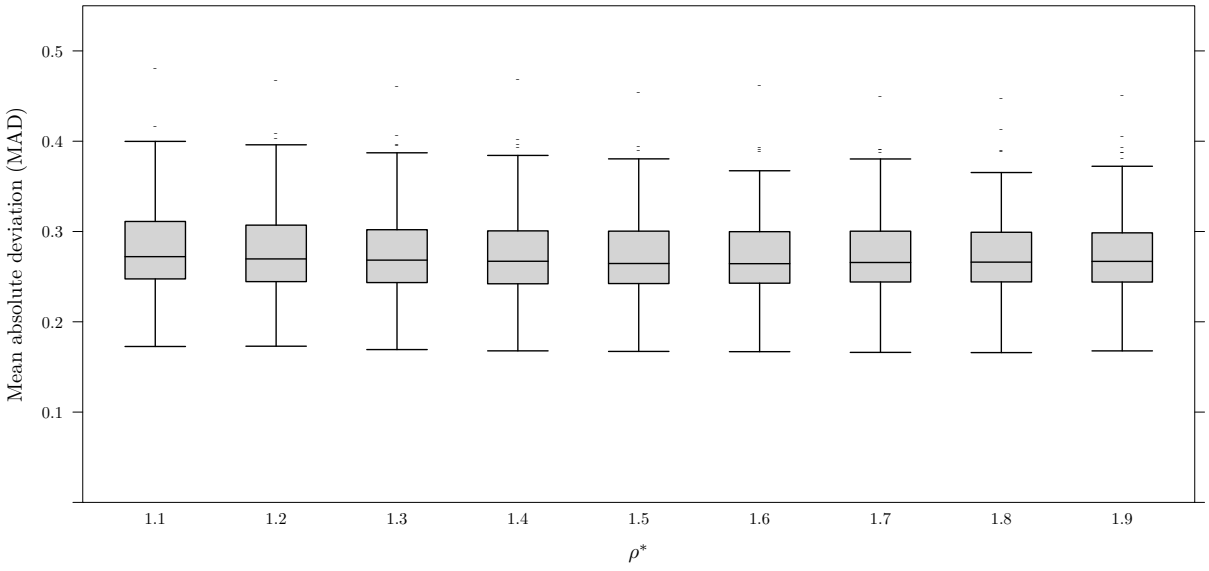

Figure 4: Simulation results for Setting I when the index is misspecified: the estimation performance of **TDboost** when varying the value of the index parameter $\rho \in \{1.1, 1.2, \ldots, 1.9\}$. In the true model $\tilde{\rho} = 1.5$ and $\tilde{\phi} = 1$. Box-plots show empirical distributions of the MADs based on 200 independent replications.

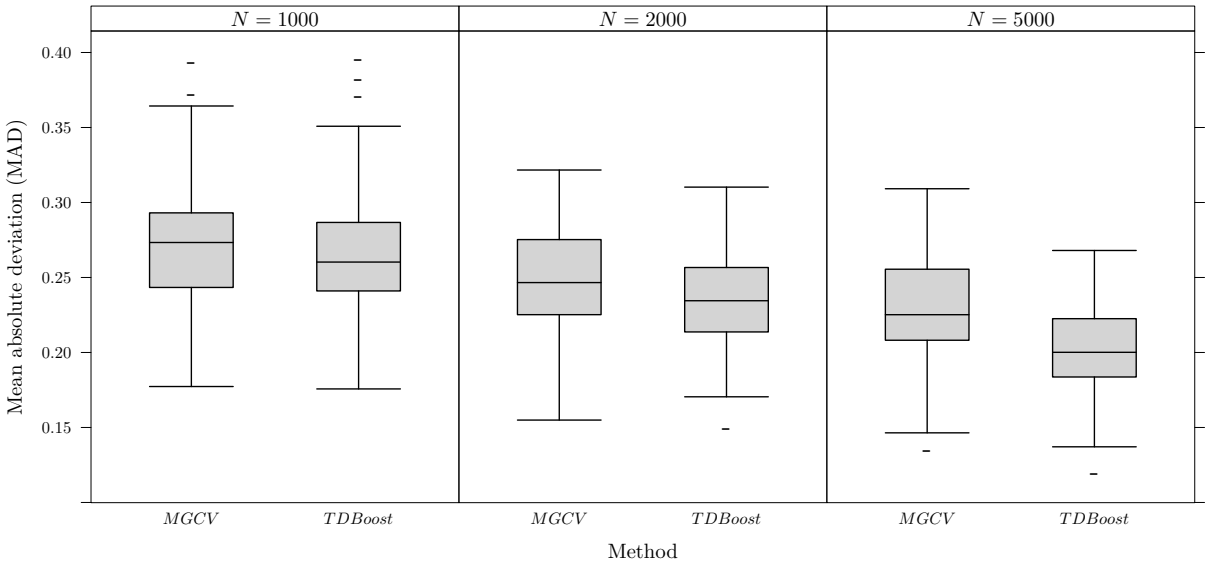

Figure 5: Simulation results for Setting II: compare the estimation performance of **MGCV** and **TDboost** when varying the training sample size and the dispersion parameter in the true model. Box-plots display empirical distributions of the MADs based on 100 independent replications.

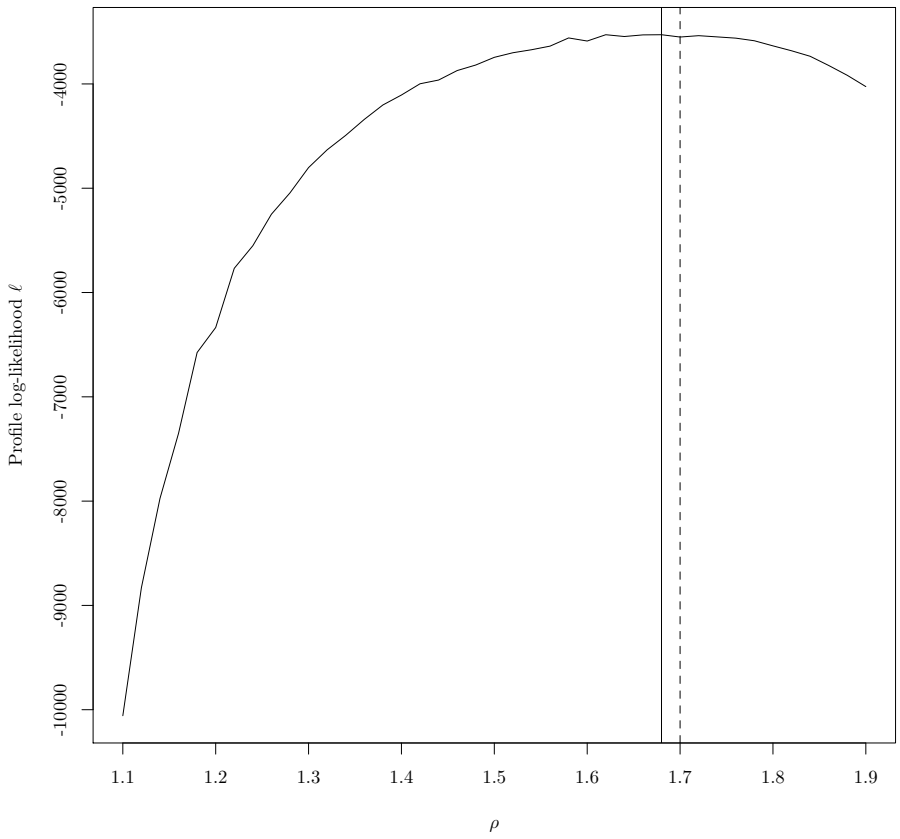

Figure 6: The curve represents the profile likelihood function of ρ from a single run. The dotted line shows the true value $\rho = 1.7$. The solid line shows the estimated value $\rho^* = 1.68$ corresponding to the maximum likelihood. The associated estimated dispersion is ϕ^* =1.89.

generated from a true model according to the setting (37) with $\phi = 2$ and $\rho = 1.7$, each sample having 2000 observations. We fit the TDboost model on each sample and compute the estimates ϕ^* at each of the 50 equally spaced values $\{\rho_1, \ldots, \rho_{50}\}$ on $(1, 2)$. The $(\rho_j, \phi^*(\rho_j))$ corresponding to the maximal profile likelihood is the estimate of (ρ, ϕ). The estimation process is repeated 200 times. The estimated indices have mean $\overline{\rho^*} = 1.68$ and standard error $SE(\rho^*) = 0.026$, so the true value $\rho = 1.7$ is within $\overline{\rho^*} \pm SE(\rho^*)$. The estimated dispersions have mean $\overline{\phi^*} = 1.82$ and standard error $SE(\phi^*) = 0.12$. Figure 6 shows the profile likelihood function of ρ for a single run.

Total Claim Amount	% obs.	% of total sum	Mean	Median
0	61.1	0	0	0
$(0, 10000]$	29.6	36.0	4902	4842
$(10000, 50000]$	9.1	61.5	27144	27679
> 50000	0.2	2.5	52157	51986

Table 2: Description of the individual total claim amount in the last five years.

6 Application: Automobile Claims

We consider an auto insurance claim dataset as analyzed in Yip and Yau (2005) and Zhang and Yu (2005). The data set contains 10,296 driver vehicle records, each record including an individual driver's total claim amount (z_i) in the last five years ($w_i = 5$) and 17 characteristics $x_i = (x_{i,1}, \ldots, x_{i,17})$ for the driver and the insured vehicle. We want to predict the expected pure premium based on x_i. Table 2 and Table 3 summarize the data set. The histogram of the total claim amounts in Figure 1 shows that the empirical distribution of these values is highly skewed. We find that approximately 61.1% of policyholders had no claims, and approximately 29.6% of the policy holders had a positive claim amount up to 10,000 dollars. Note that only 9.3% of the policyholders had a high claim amount above 10,000 dollars, but the sum of their claim amount made up to 64% of the overall sum.

We separate the entire dataset into a training set and a testing set with equal size. Then the TDboost model is fitted on the training set and tuned with five-fold cross validation. For comparison, we also fit TGLM and MGCV, both of which are fitted using all the explanatory variables. In MGCV, the numerical variables AGE, BLUEBOOK, HOMEKIDS, INCOME, KIDSDRIV, MVR_PTS, NPOLICY, RETAINED and TRAVTIME are modeled by smooth terms represented using penalized regression splines. We find the appropriate smoothness for each applicable model term using Generalized Cross Validation (GCV) (Craven and Wahba, 1978; Wahba, 1990). For the TDboost model, it is not necessary to carry out data transformation, since the tree-based boosting method can automatically handle different types of data. For other models, we use logarithmic transformation on two variables, i.e. log(BLUEBOOK), log(INCOME+10), and scale all the numerical variables except for HOMEKIDS, KIDSDRIV, MVR_PTS and NPOLICY to have mean 0 and standard deviation 1. We also create dummy variables for the categorical variables with more

ID	Variable	Type	Description
1	AGE	N	Driver's age
2	BLUEBOOK	N	Value of vehicle
3	HOMEKIDS	N	Number of children
4	INCOME	N	Annual income
5	KIDSDRIV	N	Number of driving children
6	MVR_PTS	N	Motor vehicle record points
7	NPOLICY	N	Number of policies
8	RETAINED	N	Number of years as a customer
9	TRAVTIME	N	Distance to work
10	AREA	C	Home/work area: Rural, Urban
11	CAR_USE	C	Vehicle use: Commercial, Private
12	CAR_TYPE	C	Type of vehicle: Panel Truck, Pickup, Sedan, Sports Car, SUV, Van
13	GENDER	C	Driver's gender: F, M
14	JOBCLASS	C	Unknown, Blue Collar, Clerical, Doctor, Home Maker, Lawyer, Manager, Professional, Student
15	MAX_EDUC	C	Education level: High School or Below, Bachelors, High School, Masters, PhD
16	MARRIED	C	Married or not: Yes, No
17	REVOKED	C	Whether license revoked in past 7 years: Yes, No

Table 3: Explanatory variables in the claim history data set. Type N stands for numerical variable, Type C stands for categorical variable.

	AGE	INCOME	HOMEKIDS	BLUEBOOK	KIDSDRIV
Min.	16.00	0	0.0000	1500	0.0000
1st Qu.	39.00	27620	0.0000	9200	0.0000
Median	45.00	53564	0.0000	14405	0.0000
Mean	44.84	61610	0.7199	15666	0.1694
3rd Qu.	51.00	86214	1.0000	20900	0.0000
Max.	81.00	367030	5.0000	69740	4.0000

	NPOLICY	RETAINED	TRAVTIME	MVR_PTS
Min.	1.000	1.000	5.00	0.000
1st Qu.	1.000	1.000	22.00	0.000
Median	1.000	4.000	33.00	1.000
Mean	1.695	5.328	33.42	1.709
3rd Qu.	2.000	7.000	44.00	3.000
Max.	9.000	25.000	142.00	13.000

Table 4: Descriptive statistics for the continuous variables in the claim history data set in Section 6.

AREA	MARRIED	REVOKED	GENDER
Rural: 20.2%	No: 39.9%	No: 87.8%	F: 53.8%
Urban: 79.8%	Yes: 60.1%	Yes: 12.2%	M: 46.2%

CAR_USE	MAX_EDUC	CAR_TYPE	JOBCLASS
Private: 63.2%	<High School: 14.6%	Panel Truck: 8.3%	Blue Collar: 22.2%
Commercial: 36.8%	Bachelors: 27.3%	Pickup: 17.3%	Clerical: 15.5%
	High School: 28.7%	Sedan: 26.2%	Professional: 13.6%
	Masters: 20.2%	Sports Car: 11.4%	Manager: 12.2%
	PhD: 9.2%	SUV: 27.9%	Lawyer: 10.0%
		Van: 8.9%	Student: 8.7%
			(Other): 17.8%

Table 5: Descriptive statistics for the categorical variables in the claim history data set in Section 6.

	Model		
Parameter	TGLM	MGCV	TDboost
Index ρ^*	1.37	1.34	1.34
Dispersion ϕ^*	2.31	2.15	2.05

Table 6: The estimated ρ and ϕ of the model **TGLM**, **MGCV** and **TDboost** using the profile likelihood method.

than two levels (CAR_TYPE, JOBCLASS and MAX_EDUC). For all models, we use the profile likelihood method to estimate the dispersion ϕ and the index ρ, which are in turn used in fitting the final models. The estimated values of ϕ and ρ are reported in Table 6. We see that the estimated value of the dispersion parameter in TGLM is greater than those in MGCV and TDBoost.

To examine the performance of TGLM, MGCV and TDboost, after fitting on the training set, we predict the pure premium $P(\mathbf{x}) = \hat{\mu}(\mathbf{x})$ by applying each model on the independent held-out testing set. However, attention must be paid when measuring the differences between predicted premiums $P(\mathbf{x})$ and real losses y on the testing data. The mean squared loss or mean absolute loss are not appropriate here because the losses have high proportions of zeros and are highly right skewed. Therefore an alternative statistical measure – the ordered Lorenz curve and the associated Gini index – proposed by Frees et al. (2011) are used for capturing the discrepancy between the premium and loss distributions. By calculating the

Gini index, the performance of different predictive models can be compared. Here we only briefly explain the idea of the ordered Lorenz curve (Frees et al., 2011, 2013). Let $B(\mathbf{x})$ be the "base premium", which is calculated using the existing premium prediction model, and let $P(\mathbf{x})$ be the "competing premium" calculated using an alternative premium prediction model. In the ordered Lorenz curve, the distribution of losses and the distribution of premiums are sorted based on the relative premium $R(\mathbf{x}) = P(\mathbf{x})/B(\mathbf{x})$. The ordered premium distribution is

$$\hat{D}_P(s) = \frac{\sum_{i=1}^{n} B(\mathbf{x}_i) I(R(\mathbf{x}_i) \leq s)}{\sum_{i=1}^{n} B(\mathbf{x}_i)},$$

and the ordered loss distribution is

$$\hat{D}_L(s) = \frac{\sum_{i=1}^{n} y_i I(R(\mathbf{x}_i) \leq s)}{\sum_{i=1}^{n} y_i}.$$

Two empirical distributions are based on the same sort order, which makes it possible to compare the premium and loss distributions for the same policyholder group. The ordered Lorenz curve is the graph of $(\hat{D}_P(s), \hat{D}_L(s))$. When the percentage of losses equals the percentage of premiums for the insurer, the curve results in a 45 degree line, known as "the line of equality". Twice the area between the ordered Lorenz curve and the line of equality measures the discrepancy between the premium and loss distributions, and is defined as the Gini index. Curves below the line of equality indicate that, given knowledge of the relative premium, an insurer could identify the profitable contracts, whose premiums are greater than losses. Therefore, a larger Gini index (hence a larger area between the line of equality and the curve below) would imply a more favorable model.

Following Frees et al. (2013), we successively specify the prediction from each model as the base premium $B(\mathbf{x})$ and use predictions from the remaining models as the competing premium $P(\mathbf{x})$ to compute the Gini indices. The entire procedure of the data splitting and Gini index computation are repeated 20 times, and a matrix of the averaged Gini indices and standard errors is reported in Table 7. To pick the "best" model, we use a "minimax" strategy (Frees et al., 2013) to select the base premium model that are least vulnerable to competing premium models; that is, we select the model that provides the smallest of the maximal Gini indices, taken over competing premiums. We find that the maximal Gini index

	Competing Premium		
Base Premium	TGLM	MGCV	TDboost
TGLM	0	6.432 (0.313)	14.324 (0.415)
MGCV	8.052 (0.551)	0	14.020 (0.497)
TDboost	4.677 (0.418)	2.818 (0.456)	0

Table 7: The averaged Gini indices and standard errors in the auto insurance claim data example based on 20 random splits.

is 14.324 when using $B(\mathbf{x}) = \hat{\mu}^{\text{TGLM}}(\mathbf{x})$ as the base premium, 14.020 when $B(\mathbf{x}) = \hat{\mu}^{\text{MGCV}}(\mathbf{x})$, and 4.677 when $B(\mathbf{x}) = \hat{\mu}^{\text{TDboost}}(\mathbf{x})$. Therefore, TDboost has the smallest maximum Gini index at 4.677, hence is the least vulnerable to alternative scores. Figure 7 also shows that when TGLM (or MGCV) is selected as the base premium, the area between the line of equality and the ordered Lorenz curve is larger when choosing TDboost as the competing premium, indicating again that the TDboost model represents the most favorable choice.

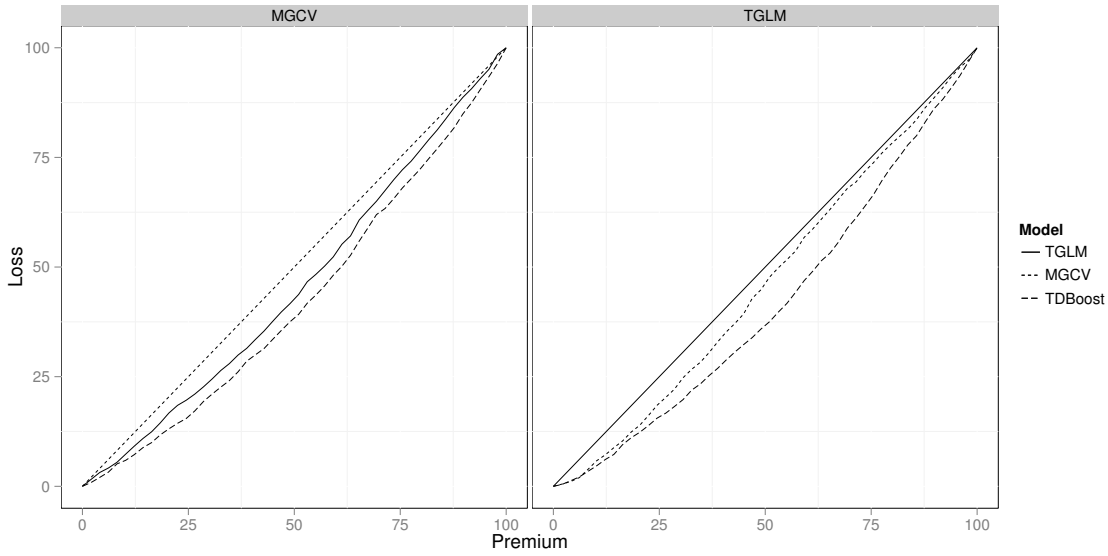

Figure 7: The ordered Lorenz curves for the auto insurance claim data.

Next, we focus on the analysis using the TDboost model. There are several explanatory variables significantly related to the pure premium. The VI measure and the baseline value of each explanatory variable are shown in Figure 8. We find that REVOKED, MVR_PTS, AREA and INCOME have high VI measure scores (the vertical line), and their scores all surpass the corresponding baselines (the horizontal line-length), indicating that the impor-

tance of those explanatory variables is real. We also find the variables AGE, CAR_TYPE, JOBCLASS, NPOLICY, MARRIED, KIDSDRIV, MAX_EDUC and CAR_USE have larger-than-baseline VI measure scores, but the absolute scales are much less than aforementioned four variables. On the other hand, although the VI measure score of, e.g., BLUEBOOK is quite large, it does not significantly surpass the baseline importance.

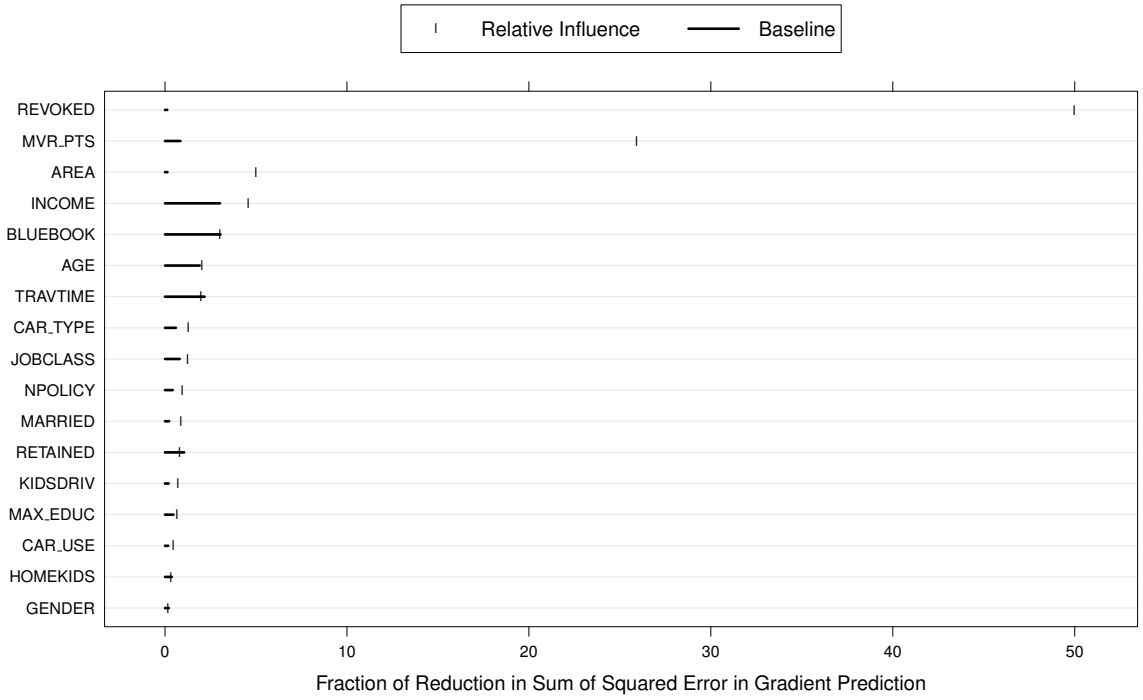

Figure 8: The variable importance measures and baselines of 17 explanatory variables for modeling the pure premium.

We now use the partial dependence plots to visualize the fitted model. Figure 9 shows the main effects of four important explanatory variables on the pure premium. We clearly see that the strong nonlinear effects exist in predictors INCOME and MVR_PTS: for the policyholders with the annual income below 163 (in $1000), their pure premium is negatively associated with the income; after the income passes 163, the pure premium starts to gradually increase with the income until the pure premium curve reaches a plateau when the income passes 237; Additionally, the pure premium is positively associated with motor vehicle record points MVR_PTS, but the pure premium curve reaches a plateau when MVR_PTS exceeds five. On the other hand, the partial dependence plots suggest that a policyholder who lives in the urban area (AREA="URBAN") or with driver's license revoked (REVOKED="YES")

typically has relatively high pure premium.

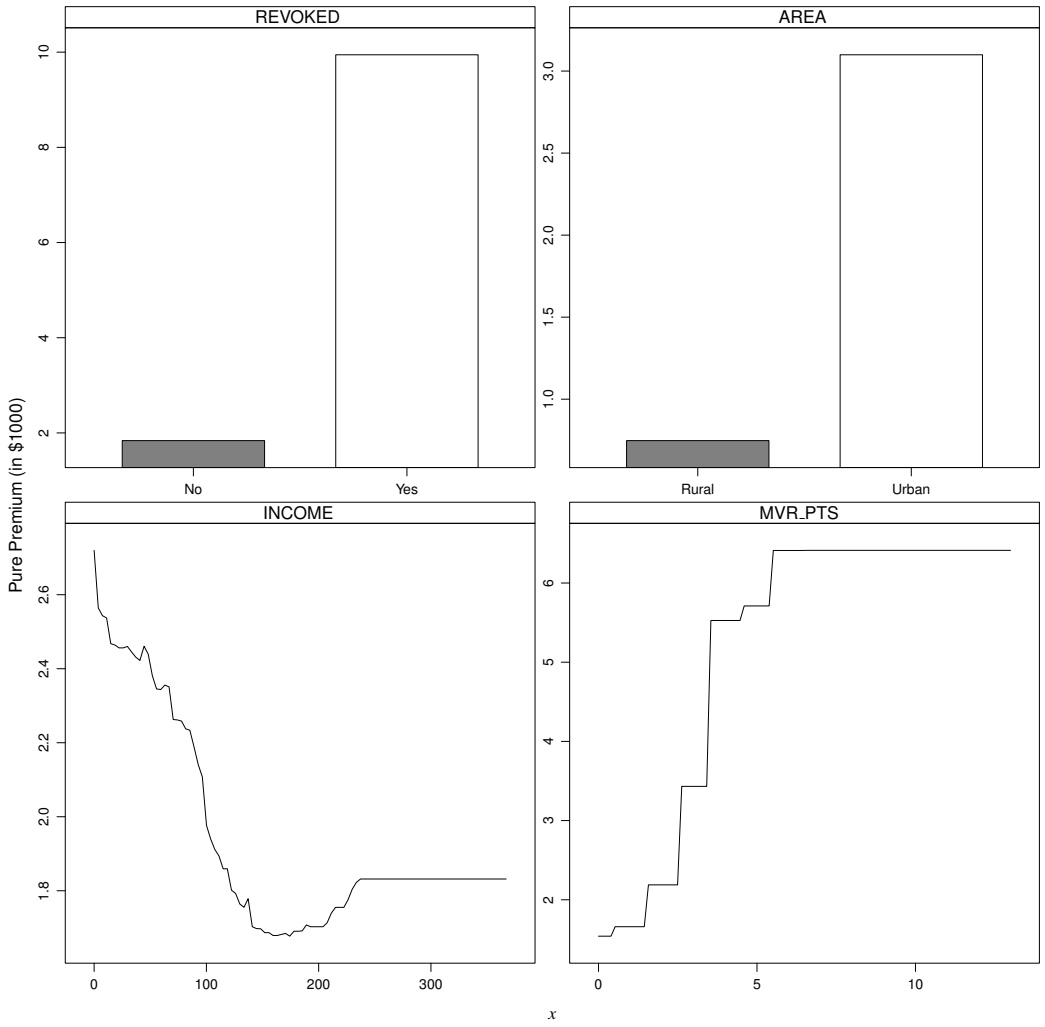

Figure 9: Marginal effects of four most significant explanatory variables on the pure premium.

In our model, the data-driven choice for the tree size is $L = 7$, which means that our model includes higher order interactions. In Figure 10, we visualize the effects of four important second order interactions using the joint partial dependence plots. These four interactions are AREA $\times$ MVR_PTS, AREA $\times$ REVOKED, AREA $\times$ NPOLICY and INCOME $\times$ MVR_PTS. The first three interactions all involve the variable AREA: we can see that the marginal effects of MVR_PTS, REVOKED and NPOLICY on the pure premium are greater for the policyholders living in the urban area (AREA="URBAN") than those living in the rural area (AREA="RURAL"). Also, a strong INCOME $\times$ MVR_PTS interaction suggests

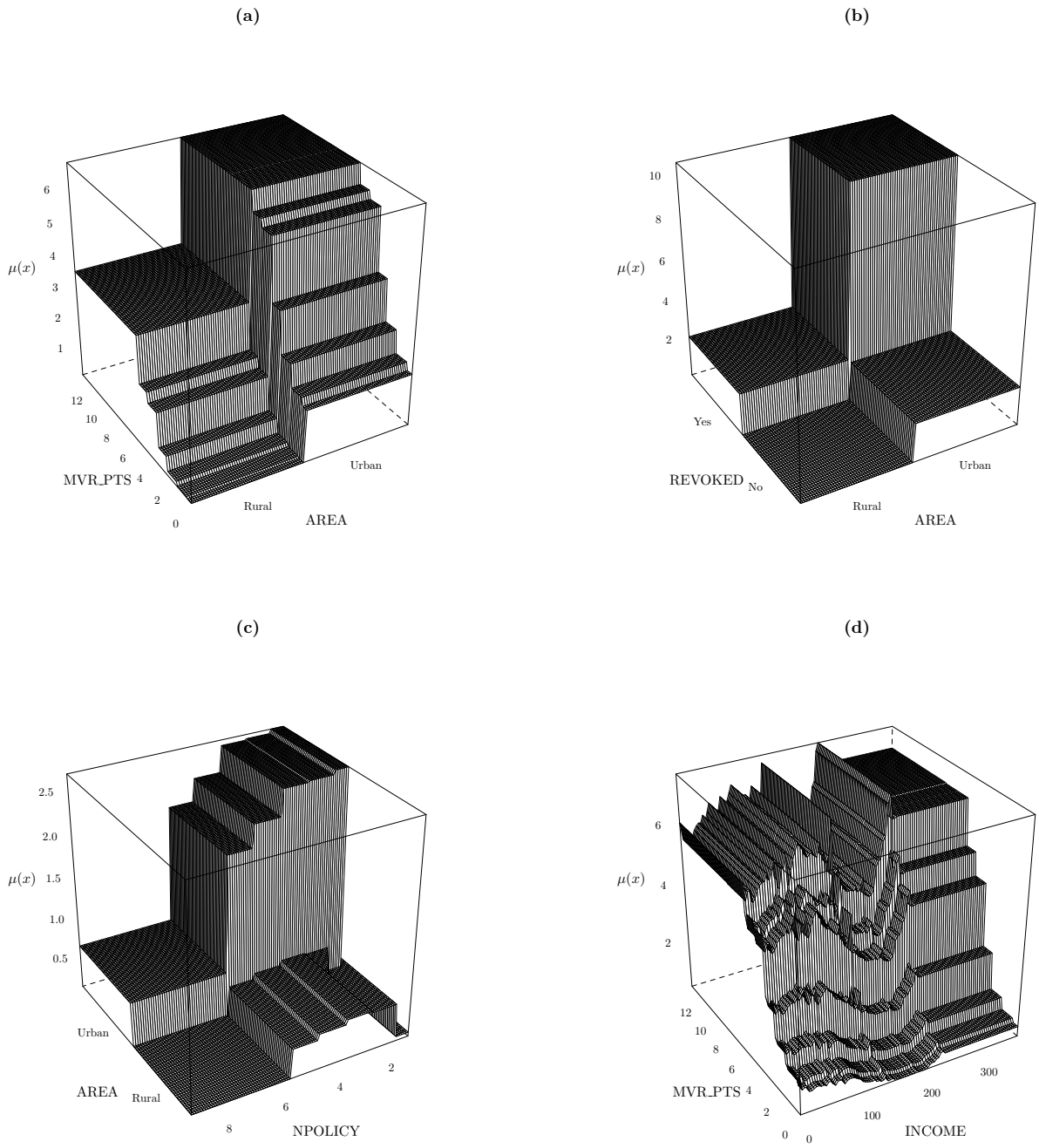

Figure 10: Four strong pairwise interactions.

that when MVR_PTS value is low, income values of the policyholders do not have a strong marginal effect on the expected pure premium; but when MVR_PTS value is high, the partial dependence of the pure premium on the income becomes stronger.

7 Conclusions

The need for nonlinear risk factors as well as risk factor interactions for modeling insurance claim sizes is well-recognized by actuarial practitioners, but practical tools to study them are very limited. In this paper, relying on neither the linear assumption nor a pre-specified interaction structure, a flexible tree-based gradient boosting method is designed for the Tweedie model. We implement the proposed method in a user-friendly R package "TDboost" that can make accurate insurance premium predictions for complex data sets and serve as a convenient tool for actuarial practitioners to investigate the nonlinear and interaction effects. In the context of personal auto insurance, we implicitly use the policy duration as a volume measure (or exposure), and demonstrate the favorable prediction performance of TDboost for the pure premium. In cases that exposure measures other than duration are used, which is common in commercial insurance, we can extend the TDboost method to the corresponding claim size by simply replacing the duration with any chosen exposure measure.

We also want to point out that TDboost can be an important complement to the traditional GLM model in insurance rating. Even under the strict circumstances that the regulators demand the final model to have a GLM structure, our approach can still be quite helpful due to its ability to extract additional information such as non-monotonicity/nonlinearity and important interaction. In Appendix A, we provide an additional real data analysis to demonstrate that our method can provide insights into the structure of interaction terms. After integrating the obtained information about the interaction terms into the original GLM model, we can much enhance the overall accuracy of the insurance premium prediction while maintaining a GLM model structure.

References

Anstey, K. J., Wood, J., Lord, S., and Walker, J. G. (2005), "Cognitive, sensory and physical factors enabling driving safety in older adults," *Clinical psychology review*, 25, 45–65.

Bar-Lev, S. K. and Stramer, O. (1987), "Characterizations of natural exponential families with power variance functions by zero regression properties," *Probability theory and related fields*, 76, 509–522.

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

— (1999), "Prediction games and arcing algorithms," *Neural Computation*, 11, 1493–1517.

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

Brent, R. P. (2013), *Algorithms for minimization without derivatives*, Courier Dover Publications.

Bühlmann, P. and Hothorn, T. (2007), "Boosting algorithms: Regularization, prediction and model fitting," *Statistical Science*, 22, 477–505.

Chiappori, P.-A. and Salanie, B. (2000), "Testing for Asymmetric Information in Insurance Markets," *The Journal of Political Economy*, 108, 56–78.

Craven, P. and Wahba, G. (1978), "Smoothing noisy data with spline functions," *Numerische Mathematik*, 31, 377–403.

Dionne, G., Gouriéroux, C., and Vanasse, C. (2001), "Testing for evidence of adverse selection in the automobile insurance market: A comment," *Journal of Political Economy*, 109, 444–453.

Dunn, P. K. and Smyth, G. K. (2005), "Series evaluation of Tweedie exponential dispersion model densities," *Statistics and Computing*, 15, 267–280.

— (2008), "Evaluation of Tweedie exponential dispersion model densities by Fourier inversion," *Statistics and Computing*, 18, 73–86.

Elith, J., Leathwick, J. R., and Hastie, T. (2008), "A working guide to boosted regression trees," *Journal of Animal Ecology*, 77, 802–813.

Feller, W. (1968), *An Introduction to Probability Theory and its Applications*, Wiley\ & Sons, New York.

Frees, E. W., Meyers, G., and Cummings, A. D. (2011), "Summarizing insurance scores using a Gini index," *Journal of the American Statistical Association*, 106.

Frees, E. W. J., Meyers, G., and Cummings, A. D. (2013), "Insurance ratemaking and a Gini index," *Journal of Risk and Insurance*.

Freund, Y. and 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.

— (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, 1189–1232.

Friedman, J., Hastie, T., and Tibshirani, R. (2000), "Additive logistic regression: A statistical view of boosting (With discussion and a rejoinder by the authors)," *The Annals of Statistics*, 28, 337–407.

Friedman, J. H. (2002), "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, 38, 367–378.

Haberman, S. and Renshaw, A. E. (1996), "Generalized linear models and actuarial science," *Statistician*, 45, 407–436.

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The elements of statistical learning: Data mining, inference, and prediction. Second Edition.*, Springer Series in Statistics, Springer.

Hastie, T. J. and Tibshirani, R. J. (1990), *Generalized additive models*, vol. 43, CRC Press.

Jørgensen, B. (1987), "Exponential dispersion models," *Journal of the Royal Statistical Society. Series B (Methodological)*, 127–162.

— (1997), *The theory of dispersion models*, vol. 76, CRC Press.

Jørgensen, B. and de Souza, M. C. (1994), "Fitting Tweedie's compound Poisson model to insurance claims data," *Scandinavian Actuarial Journal*, 1994, 69–93.

Kononenko, I. (1995), "On biases in estimating multi-valued attributes," in *Proceedings of the 14th international joint conference on Artificial intelligence-Volume 2*, Morgan Kaufmann Publishers Inc., pp. 1034–1040.

McCartt, A. T., Shabanova, V. I., and Leaf, W. A. (2003), "Driving experience, crashes and traffic citations of teenage beginning drivers," *Accident Analysis & Prevention*, 35, 311–320.

Mildenhall, S. J. (1999), "A systematic relationship between minimum bias and generalized linear models," in *Proceedings of the Casualty Actuarial Society*, vol. 86, pp. 393–487.

Murphy, K. P., Brockman, M. J., and Lee, P. K. (2000), "Using generalized linear models to build dynamic pricing systems," in *Casualty Actuarial Society Forum, Winter*, pp. 107–139.

Nelder, J. and Wedderburn, R. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society. Series A (General)*, 135, 370–384.

Ohlsson, E. and Johansson, B. (2010), *Non-life insurance pricing with generalized linear models*, Springer.

Owsley, C., Ball, K., Sloane, M. E., Roenker, D. L., and Bruni, J. R. (1991), "Visual/cognitive correlates of vehicle accidents in older drivers." *Psychology and aging*, 6, 403.

Peters, G. W., Shevchenko, P. V., and Wüthrich, M. V. (2008), "Model risk in claims reserving within TweedieÕs compound Poisson models," *ASTIN Bulletin, to appear.*

Quijano Xacur, O. A. et al. (2011), "Property and Casualty Premiums based on Tweedie Families of Generalized Linear Models," Ph.D. thesis, Concordia University.

Renshaw, A. E. (1994), "Modelling the claims process in the presence of covariates," *Astin Bulletin*, 24, 265–285.

Ridgeway, G. (2007), "Generalized Boosted Regression Models," *R package manual*.

Sandri, M. and Zuccolotto, P. (2008), "A bias correction algorithm for the Gini variable importance measure in classification trees," *Journal of Computational and Graphical Statistics*, 17.

— (2010), "Analysis and correction of bias in Total Decrease in Node Impurity measures for tree-based algorithms," *Statistics and Computing*, 20, 393–407.

Showers, V. E. and Shotick, J. A. (1994), "The effects of household characteristics on demand for insurance: A tobit analysis," *Journal of Risk and Insurance*, 492–502.

Smyth, G. and Jørgensen, B. (2002), "Fitting Tweedie's Compound Poisson Model to Insurance Claims Data: Dispersion Modelling," *ASTIN Bulletin*, 32, 143–157.

Smyth, G. K. (1996), "Regression analysis of quantity data with exact zeros," in *Proceedings of the second Australia–Japan workshop on stochastic models in engineering, technology and management*, Citeseer, pp. 572–580.

Tweedie, M. (1984), "An index which distinguishes between some important exponential families," in *Statistics: Applications and New Directions: Proc. Indian Statistical Institute Golden Jubilee International Conference*, pp. 579–604.

Van de Ven, W. and van Praag, B. M. (1981), "Risk aversion and deductibles in private health insurance: application of an adjusted tobit model to family health care expenditures," *Health, economics, and health economics*, 125–48.

Wahba, G. (1990), *Spline models for observational data*, vol. 59, SIAM.

White, A. P. and Liu, W. Z. (1994), "Technical note: Bias in information-based measures in decision tree induction," *Machine Learning*, 15, 321–329.

Wood, S. (2001), "mgcv: GAMs and generalized ridge regression for R," *R News*, 1, 20–25.

— (2006), *Generalized additive models: an introduction with R*, CRC press.

Yip, K. C. and Yau, K. K. (2005), "On modeling claim frequency data in general insurance with extra zeros," *Insurance: Mathematics and Economics*, 36, 153–163.

Zhang, T. and Yu, B. (2005), "Boosting with early stopping: Convergence and consistency," *The Annals of Statistics*, 1538–1579.

Zhang, W. (2011), "cplm: Monte Carlo EM algorithms and Bayesian methods for fitting Tweedie compound Poisson linear models," *R package*, `http://cran.r-project.org/web/packages/cplm/index.html`.