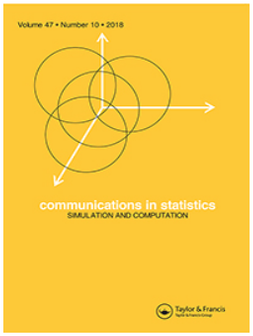

Taylor & Francis
Taylor & Francis Group

Communications in Statistics - Simulation and Computation

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/lssp20

Tweedie gradient boosting for extremely unbalanced zero-inflated data

He Zhou , Wei Qian & Yi Yang

To cite this article: He Zhou , Wei Qian & Yi Yang (2020): Tweedie gradient boosting for extremely unbalanced zero-inflated data, Communications in Statistics - Simulation and Computation, DOI: 10.1080/03610918.2020.1772302

To link to this article: https://doi.org/10.1080/03610918.2020.1772302

Published online: 11 Jul 2020.

COMMUNICATIONS IN STATISTICS - SIMULATION AND COMPUTATION®
https://doi.org/10.1080/03610918.2020.1772302

Tweedie gradient boosting for extremely unbalanced zero-inflated data

He Zhou[a], Wei Qian[b], and Yi Yang[c]

[a]School of Statistics, University of Minnesota, Minneapolis, Minnesota, USA; [b]Department of Applied Economics and Statistics, University of Delaware, Newark, Delaware, USA; [c]Department of Mathematics and Statistics, McGill University, Montreal, Quebec, Canada

ABSTRACT

Tweedie's compound Poisson model is a popular method to model insurance claims with probability mass at zero and nonnegative, highly right-skewed distribution. In particular, it is not uncommon to have extremely unbalanced data with excessively large proportion of zero claims, and even traditional Tweedie model may not be satisfactory for fitting the data. In this paper, we propose a boosting-assisted zero-inflated Tweedie model, called EMTboost, that allows zero probability mass to exceed a traditional model. We makes a nonparametric assumption on its Tweedie model component, that unlike a linear model, is able to capture nonlinearities, discontinuities, and complex higher order interactions among predictors. A specialized Expectation-Maximization algorithm is developed that integrates a blockwise coordinate descent strategy and a gradient tree-boosting algorithm to estimate key model parameters. We use extensive simulation and data analysis on synthetic zero-inflated auto-insurance claim data to illustrate our method's prediction performance.

ARTICLE HISTORY
Received 19 November 2019
Accepted 17 May 2020

KEYWORDS
Claim frequency and severity; EM algorithm; gradient boosting; Zero-inflated insurance claims data

1. Introduction

Setting premium for policyholders is one of the most important problems in insurance business, and it is crucial to predict the size of actual but unforeseeable claims. For typical portfolios in property and casualty insurance business, the policy claim for a covered risk usually has a highly right-skewed continuous distribution for positive claims, while having a probability mass at zero when a claim does not occur. This phenomenon poses unique challenges for data analysis as the data cannot be transformed to normality by power transformation and special treatment on zero claims is often required. In particular, Jørgensen and Paes De Souza (1994) and Smyth and Jørgensen (2002) used generalized linear models (GLM; Nelder and Wedderburn 1972) with a Tweedie distributed outcome, assuming Poisson arrival of claims and Gamma distributed amount for individual claims, to simultaneously model frequency and severity of insurance claims. Although Tweedie GLM has been widely used in actuarial studies (e.g., Mildenhall 1999; Murphy, Brockman, and Lee 2000; Sandri and Zuccolotto 2008), its structure of the logarithmic mean is restricted to a linear form, which can be too rigid for some applications. Yang, Qian, and Zou (2018) proposed the sparse penalized Tweedie GLM model and Fontaine et al. (2019) extended it to the multi-task sparse learning case. Zhang (2013) modeled the nonlinearity by adding splines to capture nonlinearity in claim data, and generalized additive models (GAM; Hastie and Tibshirani 1990; Wood 2006)

CONTACT Yi Yang yi.yang6@mcgill.ca Department of Mathematics and Statistics, McGill University, Montreal, QC, Canada H3A 0G4.

can also model nonlinearity by estimating smooth functions. The structures of these models have to be determined *a priori* by specifying spline degrees, main effects and interactions to be used in the model fitting. More flexibly, Yang, Qian, and Zou (2018) proposed a nonparametric Tweedie model to identify important predictors and their interactions.

Despite the popularity of the Tweedie model under linear or nonlinear logarithmic mean assumptions, it remains under-studied for problems of modeling extremely unbalanced (zero-inflated) claim data. However, it is well-known that the percentage of zeros in insurance claim data can often be well over 90%, posing challenges even for traditional Tweedie model. In statistics literature, there are two general approaches to handle data sets with excess zeros: the "Hurdle-at-zero" models and the "zero-inflated" models. The Hurdle models (e.g., Cragg 1971; Mullahy 1986) use a truncated-at-zero strategy, whose examples include truncated Poisson and truncated negative-binomial models. On the other hand, "zero-inflated" models typically use a mixture model strategy, whose examples include zero-inflated Poisson regression and zero-inflated negative binomial regression (e.g., Lambert 1992; Hall 2000; Frees, Lee, and Yang 2016), among many notable others.

In this paper, we aim to tackle the extremely unbalanced insurance data problem with excessive zeros by developing a zero-inflated nonparametric Tweedie compound Poisson model. To our knowledge, no existing work systematically studied the zero-inflated Tweedie model and its computational issues. Under a mixture model framework that subsumes traditional Tweedie model as a special case, we develop an Expectation-Maximization (EM) algorithm that efficiently integrates a blockwise coordinate descent algorithm and a gradient boosting-type algorithm to estimate key parameters. We call our method as EMTboost for brevity.

The EMTboost method assumes a mixture of Tweedie model component and a mass zero component. As one interesting feature, it can simultaneously provide estimation for the zero mass probability as well as the dispersion/power parameters of the Tweedie model component, which are useful information in understanding the zero-inflated nature of claim data under analysis. In addition, we employ boosting techniques to fit the mean of the Tweedie component. This boosting approach is motivated by its proven success for nonparametric regression and classification (Freund and Schapire 1997; Breiman 1998, 1999; Friedman 2001, 2002; Hastie, Tibshirani, and Friedman 2009). By integrating a gradient-boosting algorithm with trees as weak learners, the zero-inflated model can learn nonlinearities, discontinuities and complex higher order interactions of predictors, and potentially reduce modeling bias to produce high predictive performance. Due to the inherent use trees, this approach also naturally handles missing values, outliers and various predictor types.

The rest of the article is organized as follows. Sec. 2 briefly presents the models. The main methodology with implementation details is given in Sec. 3. We use simulations to show performance of EMTboost in Sec. 4, and apply it to analyze an auto-insurance claim data in Sec. 5. Brief concluding remarks are given in Sec. 6.

2. Zero-inflated Tweedie model

To begin with, we give a brief overview of the Tweedie's compound Poisson model, followed by the introduction of the zero-inflated Tweedie model. Let N be a Poisson random variable denoted by Pois(λ) with mean λ, and let $\tilde{Z}_d$'s ($d = 0, 1, ..., N$) 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 compound Poisson random variable Z by

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

Then Z is a Poisson sum of independent Gamma random variables. The compound Poisson distribution (Jørgensen and Paes De Souza 1994; Smyth and Jørgensen 2002) is closely connected to a special class of exponential dispersion models (Jørgensen 1987) known as Tweedie models (Tweedie 1984), whose probability density functions are of the form

$$f_{\text{Tw}}(z|\theta,\phi) = a(z,\phi)\exp\left\{\frac{z\theta-\kappa(\theta)}{\phi}\right\},$$

where $a(\cdot)$ and $\kappa(\cdot)$ are given functions, with $\theta \in \mathbb{R}$ and $\phi \in \mathbb{R}^+$. For Tweedie models, the mean and variance of Z has the property $\mathbb{E}(Z) := \mu = \dot{\kappa}(\theta)$, $\text{Var}(Z) = \phi\ddot{\kappa}(\theta)$, where $\dot{\kappa}(\theta)$ and $\ddot{\kappa}(\theta)$ are the first and second derivatives of $\kappa(\theta)$, respectively. The power mean-variance relationship is $\text{Var}(Z) = \phi\mu^\rho$ for some index parameter $\rho \in (1,2)$, which gives $\theta = \mu^{1-\rho}/(1-\rho)$, $\kappa(\theta) = \mu^{2-\rho}/(2-\rho)$ and $\ddot{\kappa}(\theta) = \mu^\rho$. If we re-parameterize the compound Poisson model by

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

then it will have the form of a Tweedie model $\text{Tw}(\mu,\phi,\rho)$ with the probability density function

$$f_{\text{Tw}}(z|\mu,\phi,\rho) := a(z,\phi,\rho)\exp\left(\frac{1}{\phi}\left(z\frac{\mu^{1-\rho}}{1-\rho} - \frac{\mu^{2-\rho}}{2-\rho}\right)\right) \tag{1}$$

where

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

with $\alpha = (2-\rho)/(\rho-1)$ and $1 < \rho < 2$. When $z > 0$, the sum of infinite series $\sum_{t=1}^{\infty} W_t$ is an example of Weight's generalized Bessel function.

With the formulation above, the Tweedie model has positive probability mass at zero with $\mathbb{P}(Z=0) = \mathbb{P}(N=0) = \exp(-\lambda)$. Despite its popularity in actuarial studies, Tweedie models do not always give ideal performance in cases when the empirical distribution of claim data (e.g., in auto insurance), is extremely unbalanced and has an excessively high proportion of zero claims, which will be illustrated in the numerical exposition. This motivates us to consider a zero-inflated mixture model that combines a Tweedie distribution with probability q and an exact zero mass with probability $1-q$:

$$Y = \begin{cases} Z, & \text{with probability } q, \text{where } Z \sim \text{Tw}(\mu,\phi,\rho), \\ 0, & \text{with probability } 1-q. \end{cases} \tag{2}$$

We denote this zero-inflated Tweedie model by $Y \sim \text{ZIF-Tw}(\mu,\phi,\rho,q)$. The probability density function of Y can be written as

$$f_{\text{ZIF-Tw}}(y|\mu,\phi,\rho,q) := qf_{\text{Tw}}(y|\mu,\phi,\rho) + (1-q)I\{y=0\},$$

so that $\mathbb{P}(Y=0) = q\exp\left(-\frac{1}{\phi}\frac{\mu^{2-\rho}}{2-\rho}\right) + (1-q)$ and $\mathbb{E}(Y) = q\mu$.

3. Methodology

Let $Z_w = \sum_{d=1}^{N_w}\tilde{Z}_d$ be the total claim amount, and the number of claims N_w is Poisson distributed $\text{Pois}(\lambda w)$. Here ω is the duration, i.e., the length of time that the policy remains in force.

Conditional on N_w, assume Z_d's $(d = 1, ..., N)$ are i.i.d. Gamma(α, γ). Let $Y(w) = Z_w/w$ be the total claim amount averaged over the duration w. We can show that $Y(w) \sim \text{Tw}(\mu, \phi/w, \rho)$: first, we can see that $Y(1) \sim \text{Tw}(\mu, \phi, \rho)$ as defined in (1)

$$E(Y(1)) = E(E(Y(1)|N_1)) = \lambda\alpha\gamma,$$
$$\text{Var}(Y(1)) = E(\text{Var}(Y(1)|N_1)) + \text{Var}(E(Y(1)|N_1)) = \lambda\alpha\gamma^2 + \lambda\alpha^2\gamma^2.$$

Therefore

$$E(Y(w)) = \frac{1}{w}E(Z_w) = \frac{1}{w}\lambda w\alpha\gamma = \lambda\alpha\gamma,$$
$$\text{Var}(Y(w)) = \frac{1}{w^2}\text{Var}(Z_w) = (\lambda\alpha\gamma^2 + \lambda\alpha^2\gamma^2)/w.$$

Since the mean-variance relation for $Y(1)$ is $\text{Var}(Y(1)) = \phi[E(Y(1))]^\rho$, we can obtain the mean-variance relation for $Y(w)$

$$\text{Var}(Y(w)) = \frac{1}{w}\text{Var}(Y(1)) = \frac{\phi}{w}(E(Y(1)))^\rho = \frac{\phi}{w}(E(Y(w)))^\rho.$$

By the scale-invariance property of Tweedie distribution, we see that indeed $Y(w) \sim \text{Tw}(\mu, \phi/w, \rho)$.

Now consider a portfolio of polices $\mathbf{D} = \{(y_i, \mathbf{x}_i, \omega_i)\}_{i=1}^n$ from n independent insurance policy contracts, where for the ith contract, y_i is the policy pure premium, $\mathbf{x}_i$ is a p-dimensional vector of explanatory variables that characterize the policyholder and the risk being insured, and ω_i is the policy duration. If we assume that each policy pure premium Y_i under unit duration is an observation from the zero-inflated Tweedie distribution $Y_i \sim \text{ZIF-Tw}(\mu_i, \phi, \rho, q)$. Then we know that $Y_i \sim \text{ZIF-Tw}(\mu_i, \phi/\omega_i, \rho, q)$ as defined in (2). For now we assume that the value of ρ is given and in the end of this section we will discuss the estimation of ρ. Assume μ_i is determined by a regression function $F: \mathbb{R}^p \to \mathbb{R}$ of $\mathbf{x}_i$ through the log link function

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

Let $\boldsymbol{\theta} = (F, \phi, q) \in \mathcal{F} \times \mathbb{R}^+ \times [0, 1]$ denote a collection of parameters to be estimated with $\mathcal{F}$ denoting a class of regression functions (based on tree learners). Our goal is to maximize the log-likelihood function of the mixture model

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \log \mathcal{L}(\boldsymbol{\theta}; \mathbf{D}).$$

where

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{D}) := \prod_{i=1}^n f_{\text{ZIF-Tw}}\left(y_i | \exp(F(\mathbf{x}_i)), \phi/\omega_i, \rho, q\right), \tag{3}$$

but doing so directly is computationally difficult. To efficiently estimate $\boldsymbol{\theta} = (F, \phi, q)$, we propose a gradient-boosting based EM algorithm, referred to as EMTboost henceforth. We first give an outline of the EMTboost algorithm and the details will be discussed further in Secs. 3.1 and 3.2. The basic idea is to first construct a proxy Q-function corresponding to the current iterate by which the target likelihood function (3) is lower bounded (E-step), and then maximize the Q-function to get the next update (M-step) so that 3 can be driven uphill:

E-Step Construction We introduce n independent Bernoulli latent variables $\Pi_1, ..., \Pi_n$ such that for $i = 1, ..., n, P(\Pi_i = 1) = q$, and $\Pi_i = 1$ when y_i is sampled from Tweedie$(\mu_i, \phi/\omega_i, \rho)$ and $\Pi_i = 0$ if y_i is from the exact zero point mass. Denote $\Pi = (\Pi_1, ..., \Pi_n)^\top$. Assume predictors $\mathbf{x}_i$'s are fixed. Given $\Pi_i \in \{0, 1\}$ and $\boldsymbol{\theta}$, the joint-distribution of the complete model for each observation is

$$f(y_i, \Pi_i|\boldsymbol{\theta}) := \left(q \cdot f_{\mathrm{Tw}}(y_i|\exp(F(\mathbf{x}_i)), \phi, \omega_i)\right)^{\Pi_i}((1-q)\cdot I\{y_i=0\})^{1-\Pi_i}.$$

The posterior distribution of each latent variable Π_i is

$$f(\Pi_i|y_i, \boldsymbol{\theta}) = \frac{f(y_i, \Pi_i|\boldsymbol{\theta})}{f(y_i, \Pi_i|\boldsymbol{\theta}) + f(y_i, 1-\Pi_i|\boldsymbol{\theta})}.$$

For the E-step construction, denote $\boldsymbol{\theta}^t = (F^t, \phi^t, q^t)$ the value of $\boldsymbol{\theta}$ during tth iteration of the EMTboost algorithm. The Q-function for each observation is

$$\begin{aligned} Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^t) &:= \mathbb{E}_{\Pi_i \sim f(\Pi_i|y_i, \boldsymbol{\theta}^t)}\left[\log f(y_i, \Pi_i|\boldsymbol{\theta})\right] \\ &= f(\Pi_i = 1|y_i, \boldsymbol{\theta}^t)\log f(y_i, \Pi_i = 1|\boldsymbol{\theta}) + f(\Pi_i = 0|y_i, \boldsymbol{\theta}^t)\log f(y_i, \Pi_i = 0|\boldsymbol{\theta}) \\ &= \delta^t_{1,i}(\boldsymbol{\theta}^t)\log\left(q \cdot f_{\mathrm{Tw}}(y_i|\exp(F(\mathbf{x}_i)), \phi, \omega_i)\right) + \delta^t_{0,i}(\boldsymbol{\theta}^t)\log(1-q)I\{y_i=0\}, \end{aligned}$$

where

$$\delta^t_{1,i}(\boldsymbol{\theta}^t) = f(\Pi_i = 1|y_i, \boldsymbol{\theta}^t) = \begin{cases} 1, & \text{if } y_i > 0; \\ \dfrac{q^t \exp\left(\dfrac{\omega_i}{\phi^t}\left(-\dfrac{\exp\left(F^t(\mathbf{x}_i)(2-\rho)\right)}{2-\rho}\right)\right)}{q^t \exp\left(\dfrac{\omega_i}{\phi^t}\left(-\dfrac{\exp\left(F^t(\mathbf{x}_i)(2-\rho)\right)}{2-\rho}\right)\right) + (1-q^t)}, & \text{if } y_i = 0, \end{cases} \tag{4}$$

$$\delta^t_{0,i}(\boldsymbol{\theta}^t) = f(\Pi_i = 0|y_i, \boldsymbol{\theta}^t) = 1 - \delta^t_{1,i}(\boldsymbol{\theta}^t) \tag{5}$$

Given n observations of data $\mathbf{D} = \{(y_i, \mathbf{x}_i, \omega_i)\}_{i=1}^n$, the Q-function is

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) &= \frac{1}{n}\sum_{i=1}^n Q_i(\boldsymbol{\theta}|\boldsymbol{\theta}^t) \\ &= \frac{1}{n}\sum_{i=1}^n \delta^t_{1,i}(\boldsymbol{\theta}^t)\log\left(q \cdot f_{\mathrm{Tw}}(y_i|\exp(F(\mathbf{x}_i)), \phi, \omega_i)\right) + \delta^t_{0,i}(\boldsymbol{\theta}^t)\log\left((1-q)\cdot I\{y_i=0\}\right) \\ &= \frac{1}{n}\sum_{i=1}^n \delta^t_{1,i}(\boldsymbol{\theta}^t)\log\left\{qa(y_i, \phi/\omega_i, \rho)\exp\left[\frac{\omega_i}{\phi}\left(y_i\frac{\exp((1-\rho)F(\mathbf{x}_i))}{1-\rho} - \frac{\exp((2-\rho)F(\mathbf{x}_i))}{2-\rho}\right)\right]\right\} \\ &+ \frac{1}{n}\sum_{\{i y_i=0\}} \delta^t_{0,i}(\boldsymbol{\theta}^t)\log(1-q) \end{aligned} \tag{6}$$

M-Step Maximization

Given the Q-function (6), we update $\boldsymbol{\theta}^t$ to $\boldsymbol{\theta}^{t+1}$ through maximization of (6) by

$$\boldsymbol{\theta}^{t+1} = \left(F^{t+1}, \phi^{t+1}, q^{t+1}\right) \leftarrow \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t),$$

in which F^{t+1}, ϕ^{t+1} and q^{t+1} are updated successively through blockwise coordinate descent

$$F^{t+1} \leftarrow \arg\max_{F\in\mathcal{F}} Q\left(F\middle|\left(F^t, \phi^t, q^t\right)\right) \tag{7}$$

$$\phi^{t+1} \leftarrow \arg\max_{\phi\in\mathbb{R}^+} Q\left(\phi\middle|\left(F^{t+1}, \phi^t, q^t\right)\right) \tag{8}$$

$$q^{t+1} \leftarrow \arg\max_{q} Q\Big(q\big|\big(F^{t+1}, \phi^{t+1}, q^{t}\big)\Big) \tag{9}$$

Specifically, (7) is equivalent to update

$$F^{t+1} \leftarrow \arg\max_{F\in\mathscr{F}} \sum_{i=1}^{n} \delta_{1,i}^{t}(F^{t}, \phi^{t}, q^{t})\Psi(y_i, F(\mathbf{x}_i), \omega_i), \tag{10}$$

where the risk function Ψ is defined as

$$\Psi(y_i, F(\mathbf{x}_i), \omega_i) = \omega_i\left(y_i\frac{\exp\big(F(\mathbf{x}_i)(1-\rho)\big)}{1-\rho} - \frac{\exp\big(F(\mathbf{x}_i)(2-\rho)\big)}{2-\rho}\right). \tag{11}$$

We use a gradient tree-boosted algorithm to compute (10), and its details are deferred to Sec. 3.1. After updating F^{t+1} we then update ϕ^{t+1} in (8) using

$$\begin{aligned}\phi^{t+1} \leftarrow \arg\max_{\phi\in\mathbb{R}^{+}} \sum_{i=1}^{n}\delta_{1,i}^{t}(F^{t+1}, \phi^{t}, q^{t})\Bigg\{&\log a(y_i, \phi/\omega_i, \rho),\\ &+\frac{\omega_i}{\phi}\left(y_i\frac{\exp\big((1-\rho)F^{t+1}(\mathbf{x}_i)\big)}{1-\rho} - \frac{\exp\big((2-\rho)F^{t+1}(\mathbf{x}_i)\big)}{2-\rho}\right)\Bigg\}.\end{aligned} \tag{12}$$

Conditional on the updated F^{t+1} and q^{t}, maximizing the log-likelihood function with respect to ϕ in (12) is a univariate optimization problem that can be solved by using a combination of golden section search and successive parabolic interpolation (Brent 2013).

After updating F^{t+1} and ϕ^{t+1}, we can use a simple formula to update q^{t+1} for (9)

$$q^{t+1} \leftarrow \frac{1}{n}\sum_{i=1}^{n}\delta_{1,i}^{t}\big(F^{t+1}, \phi^{t+1}, q^{t}\big). \tag{13}$$

We repeat the above E-step and M-step iteratively until convergence. In summary, the complete EMTboost algorithm is shown in Algorithm 1.

Algorithm 1. EMTboost Algorithm

Input: Dataset $\mathbf{D} = \{(y_i, \mathbf{x}_i, \omega_i)\}_{i=1}^{n}$ and the index parameter ρ.

Output: Estimates $\hat{\boldsymbol{\theta}} = (\hat{F}, \hat{\phi}, \hat{q})$.

1 Initialize $\boldsymbol{\theta}^{0} = (F^{0}, \phi^{0}, \rho^{0})$. Compute the index set $\mathcal{I} = \{i: y_i = 0\}$ and initialize $\{\delta_{0,i}^{0}, \delta_{1,i}^{0}\}_{i\in\mathcal{I}}$ by setting $\delta_{1i} = 1, \delta_{0i} = 0$ for $i \notin \mathcal{I}$.

2 **for** $t = 0, 1, 2, \ldots, T$ **do**

3 **E-step:** Update $\{\delta_{0,i}^{t}, \delta_{1,i}^{t}\}_{i\in\mathcal{I}}$ by (4) and (5).

4 **M-step:** Update $\boldsymbol{\theta}^{t+1} = (F^{t+1}, \phi^{t+1}, q^{t+1})$ by using (10) that calls Algorithm 3, (12) and (13).

$$F^{t+1} \leftarrow \arg\max_{F\in\mathcal{F}} Q\Big(F\big|\big(F^{t}, \phi^{t}, q^{t}\big)\Big)$$

$$\phi^{t+1} \leftarrow \arg\max_{\phi\in\mathbb{R}^{+}} Q\Big(\phi\big|\big(F^{t+1}, \phi^{t}, q^{t}\big)\Big)$$

$$q^{t+1} \leftarrow \arg\max_{q} Q\Big(q\big|\big(F^{t+1}, \phi^{t+1}, q^{t}\big)\Big)$$

5 end

6 Return $\hat{\boldsymbol{\theta}} = (\hat{F}, \hat{\phi}, \hat{q}) = (F^{T}, \phi^{T}, q^{T})$.

So far we only assume that the value of ρ is known when estimating $\boldsymbol{\theta} = (F, \phi, q)$. Next we give a profile likelihood method to jointly estimate $(\boldsymbol{\theta}, \rho) = (F, \phi, q, \rho)$ when ρ is unknown. Following Dunn and Smyth (2005),we pick a sequence of K equally-spaced candidate values $\{\rho_1, ..., \rho_K\}$ on the interval (1, 2), and for each fixed ρ_k, $k = 1, ..., K$, we apply Algorithm 1 to maximize the log-likelihood function (3) with respect to $\boldsymbol{\theta}_{\rho_k} = (F_{\rho_k}, \phi_{\rho_k}, q_{\rho_k})$, which gives the corresponding estimators $\hat{\boldsymbol{\theta}}_{\rho_k} = (\hat{F}_{\rho_k}, \hat{\phi}_{\rho_k}, \hat{q}_{\rho_k})$ and the log-likelihood function $\mathcal{L}(\hat{\boldsymbol{\theta}}_{\rho_k}; \mathbf{D}, \rho_k)$. Then from the sequence $\{\rho_1, ..., \rho_K\}$ we choose the optimal $\hat{\rho}$ as the maximizer of $\mathcal{L}$.

$$\hat{\rho} = \arg \max_{\rho \in \{\rho_1, ..., \rho_K\}} \left\{ \mathcal{L}\left(\hat{\boldsymbol{\theta}}_{\rho}; \mathbf{D}, \rho\right) \right\}.$$

We then obtain the corresponding estimator $\hat{\boldsymbol{\theta}}_{\hat{\rho}} = (\hat{F}_{\hat{\rho}}, \hat{\phi}_{\hat{\rho}}, \hat{q}_{\hat{\rho}})$. This profile likelihood algorithm is shown in Algorithm 2.

Algorithm 2. Profile Likelihood for EMTboost

Input: Dataset $\mathbf{D} = \{(y_i, \mathbf{x}_i, \omega_i)\}_{i=1}^n$.

Output: Estimates $\hat{\boldsymbol{\theta}}_{\hat{\rho}} = (\hat{F}_{\hat{\rho}}, \hat{\phi}_{\hat{\rho}}, \hat{q}_{\hat{\rho}})$.

1 Pick a sequence of K equally-spaced candidate values $\{\rho_1, ..., \rho_K\}$ on the interval (1, 2).
for $k = 1, ..., K$ **do**
2 Set $\rho = \rho_k$.
3 Call Algorithm 1 to compute $\hat{\boldsymbol{\theta}}_{\rho_k} = (\hat{F}_{\rho_k}, \hat{\phi}_{\rho_k}, \hat{q}_{\rho_k})$ and the corresponding log-likelihood function $\mathcal{L}(\hat{\boldsymbol{\theta}}_{\rho_k}; \mathbf{D}, \rho_k)$.
4 **end**
5 Compute the optimal $\hat{\rho}$

$$\hat{\rho} = \arg \max_{\rho \in \{\rho_1, ..., \rho_K\}} \left\{ \mathcal{L}\left(\hat{\boldsymbol{\theta}}_{\rho}; \mathbf{D}, \rho\right) \right\}.$$

6 Return the final estimator $\hat{\boldsymbol{\theta}}_{\hat{\rho}} = (\hat{F}_{\hat{\rho}}, \hat{\phi}_{\hat{\rho}}, \hat{q}_{\hat{\rho}})$.

3.1. Estimating $F(\cdot)$ via tree-based gradient boosting

To minimize the weighted sum of the risk function (11), we employ the tree-based gradient boosting algorithm to recover the predictor function $F(\cdot)$:

$$\tilde{F}(\cdot) = \arg \min_{F(\cdot) \in \mathcal{F}} \sum_{i=1}^{n} \delta_{1,i} \Psi(y_i, F(\mathbf{x}_i), \omega_i),$$

Note that the objective function does not depend on ϕ. To solve the gradient-tree boosting, each candidate function $F \in \mathcal{F}$ is assumed to be an ensemble of L-terminal nodes regression trees, as base learners:

$$\begin{aligned} F(\mathbf{x}) &= F^{[0]} + \sum_{m=1}^{M} \beta^{[m]} h\left(\mathbf{x}; \xi^{[m]}\right), \\ &= F^{[0]} + \sum_{m=1}^{M} \beta^{[m]} \left\{ \sum_{l=1}^{L} u_l^{[m]} I(\mathbf{x} \in R_l^{[m]}) \right\} \end{aligned}$$

where $F^{[0]}$ is a constant scalar, $\beta^{[m]}$ is the expansion coefficient and $h(\mathbf{x}; \xi^{[m]})$ is the mth base learner, characterized by the parameter $\xi^{[m]} = \{R_l^{[m]}, u_l^{[m]}\}_{l=1}^{L}$, with $R_l^{[m]}$ being the disjoint regions

representing the terminal nodes of the tree, and constants $u_l^{[m]}$ being the values assigned to each region.

The constant $\hat{F}^{[0]}$ is chosen as the 1-terminal tree that minimizes the negative log-likelihood. A forward stagewise algorithm (Friedman 2001) builds up the components $\beta^{[m]}h(\mathbf{x};\boldsymbol{\xi}^{[m]})$ sequentially through a gradient-descent-like approach with $m = 1, 2, ..., M$. At iteration stage m, suppose that the current estimation for $\tilde{F}(\cdot)$ is $\hat{F}^{[m-1]}(\cdot)$. To update from $\hat{F}^{[m-1]}(\cdot)$ to $\hat{F}^{[m]}(\cdot)$, the gradient-tree boosting method fits the mth regression tree $h(\mathbf{x};\boldsymbol{\xi}^{[m]})$ to the negative gradient vector by least-squares function minimization:

$$\hat{\boldsymbol{\xi}}^{[m]} = \arg\min_{\boldsymbol{\xi}^{[m]}} \sum_{i=1}^{n} \left[g_i^{[m]} - h\left(\mathbf{x}_i;\boldsymbol{\xi}^{[m]}\right) \right]^2,$$

where $(g_1^{[m]}, ..., g_n^{[m]})^\top$ is the current negative gradient vector of Ψ with respect to (w.r.t.) $\hat{F}^{[m-1]}$:

$$g_i^{[m]} = -\frac{\partial \Psi\left(y_i, F(\mathbf{x}_i), \omega_i\right)}{\partial F(\mathbf{x}_i)}\bigg|_{F(\mathbf{x}_i)=\hat{F}^{[m-1]}(\mathbf{x}_i)}.$$

When fitting this regression trees, first use a fast top-down "best-fit" algorithm with a least-squares splitting criterion (Friedman, Hastie, and Tibshirani 2000) to find the splitting variables and the corresponding splitting locations that determine the terminal regions $\{\hat{R}_l^{[m]}\}_{l=1}^L$, then estimate the terminal-node values $\{\hat{u}_l^{[m]}\}_{l=1}^L$. This fitted regression tree $h(\mathbf{x};\{\hat{u}_l^{[m]}, \hat{R}_l^{[m]}\}_{l=1}^L)$ can be viewed as a tree-constrained approximation of the unconstrained negative gradient. Due to the disjoint nature of the regions produced by regression trees, finding the expansion coefficient $\beta^{[m]}$ can be reduced to solving L optimal constants $\eta_l^{[m]}$ within each region $\hat{R}_l^{[m]}$. And the estimation of $\tilde{F}$ for the next stage becomes

$$\hat{F}^{[m]} = \hat{F}^{[m-1]} + \nu \sum_{l=1}^{L} \hat{\eta}_l^{[m]} I(\mathbf{x} \in \hat{R}_l^{[m]}), \tag{14}$$

where $0 \le \nu \le 1$ is the shrinkage factor 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 overfitting and improve the predictive accuracy. The complete algorithm is shown in Algorithm 3.

Algorithm 3. TDboost Algorithm

Input: Dataset $\mathbf{D} = \{(y_i, \mathbf{x}_i, \omega_i)\}_{i=1}^n$ and the index parameter ρ.

Output: Estimates $\hat{F}$.

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

2 **for** $m = 0, 1, 2, ..., M$ **do**

3 Compute the negative gradient vector $(g_1^{[m]}, ..., g_n^{[m]})^\top$

$$g_i^{[m]} = \omega_i \Big\{ -y_i \exp\left[(1-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)\right] + \exp\left[(2-\rho)\hat{F}^{[m-1]}(\mathbf{x}_i)\right] \Big\}, i = 1, ..., n.$$

4 Fit the negative gradient vector to $(\mathbf{x}_1, ..., \mathbf{x}_n)^\top$ by an L-terminal node regression tree, giving the partition $\{\hat{R}_l^{[m]}\}_{l=1}^L$.

5 Compute the optimal terminal node predictions $\eta_l^{[m]}$ for each region $\hat{R}_l^{[m]}, l = 1, 2, ..., L$

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

6 Update $\hat{F}^{[m]}$ for each region $\hat{R}_l^{[m]}$ by (14).
7 **end**
8 Return $\hat{F} = \hat{F}^{[M]}$.

3.2. Implementation details

Next we give a data-driven method to find initial values for parameter estimation. The idea is that we approximately view the latent variables as $\Pi_i \approx I\{y_i \neq 0\}$. That is, we treat all zeros as if they are all from the exact zero mass portion, which can be reasonable for extremely unbalanced zero-inflated data. If the latent variables were known, it is straightforward to find the MLE solution of a constant mean model:

$$\boldsymbol{\theta}^0 = \arg\max_{\boldsymbol{\theta}\in\Theta} \log \mathcal{L}\left(\boldsymbol{\theta}; \mathbf{D}, \tilde{\Pi}\right),$$

where $\Theta = \mathcal{C} \times \mathbb{R}^+ \times [0, 1]$, $\mathcal{C} = \{F \equiv \eta \mid \eta \in \mathbb{R}\}$, and η is a constant scalar. We then find initial values successively as follows:

Initialize F^0 by

$$\begin{aligned} F^0 &= \arg\min_{\eta\in\mathbb{R}} \sum_{i=1}^n I\{y_i \neq 0\} \cdot \Psi(y_i, \eta, \omega_i) \\ &= \log\left[\frac{\sum_{i=1}^n I\{y_i \neq 0\} \cdot y_i \cdot \omega_i}{\sum_{i=1}^n I\{y_i \neq 0\} \cdot \omega_i}\right]. \end{aligned}$$

Initialize ϕ^0 by

$$\phi^0 = \arg\min_{\phi\in\mathbb{R}^+} \sum_{i=1}^n I\{y_i \neq 0\}\left(\log a(y_i, \phi/\omega_i, \rho) + \frac{\omega_i}{\phi}\left(y_i \frac{\exp\left(F^0(1-\rho)\right)}{1-\rho} - \frac{\exp\left(F^0(2-\rho)\right)}{2-\rho}\right)\right),$$

Initialize q^0 by

$$q^0 = \frac{1}{n}\sum_{i=1}^n I\{y_i \neq 0\}.$$

Given $\boldsymbol{\theta}^0$ obtained above, we can then initialize $(\delta_{1,i}, \delta_{0,i})$ by Equations (4) and (5), giving $(\delta_{0,i}^0, \delta_{1,i}^0)$.

As a last note, when implementing EMTboost algorithm, for more stable computation, we may want to avoid that the probability q converges to 1 (or 0). In such case, we can add a regularization term $r\log(1-q)$ on q so that each M-step in Q-function (6) becomes

$$\mathcal{P}Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^t\right) = Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^t\right) + \underbrace{r\log(1-q)}_{\text{regularization term}}, \tag{15}$$

where $r \in \mathbb{R}^+$ is a non-negative regularization parameter. Apparently, when maximizing the penalized log-likelihood function (15), larger q will be penalized more. We establish the EM algorithm similar as before, and only need to modify the Maximization step of (13) w.r.t. q:

$$q_{\mathcal{P}}^{t+1} = \frac{\frac{1}{n}\sum_{i=1}^{n}\delta_{1,i}^{t}}{r+1}, \tag{16}$$

pulling the original update $q^{t+1} = \frac{1}{n}\sum_{i=1}^{n}\delta_{1,i}^{t}$ toward 0 by fraction $r+1$. Alternatively, if in some cases, we want to avoid that the EMTboost model degrades to an exact zero mass, the regularization term can be chosen as $r\log(1-|1-2q|)$. The updating step with respect to q becomes a soft thresholding update with the threshold r:

$$q_{\mathcal{P}'}^{t+1} = \frac{1}{2} - \frac{\mathcal{S}_r\left(1 - \frac{2}{n}\sum_{i=1}^{n}\delta_{1,i}^{t}\right)}{2(r+1)}$$

where $\mathcal{S}_r(\cdot)$ is the soft thresholding function with $\mathcal{S}_r(x) = \text{sign}(x)(|x|-r)_+$. We apply these penalized EMTboost methods to the real data application in Appendix C.1.

4. Simulation studies

We have implemented our proposed method in R, the source code is publicly available on GitHub at https://github.com/emeryyi/EMTboost.git. In this section, we compare the EMTboost model (Sec. 3) with a regular Tweedie boosting model (i.e. $q \equiv 1$; TDboost) and the Gradient Tree-Boosted Tobit model (Grabit; Sigrist and Hirnschall 2019) in terms of the function estimation performance. The Grabit model extends the Tobit model (Tobin 1958) using gradient-tree boosting algorithm. We here present two simulation studies in which zero-inflated data are generated from zero-inflated Tweedie model (Case 1 in Sec. 4.2) and zero-inflated Tobit model (Case 2 in Sec. 4.3). An additional simulation result (Case 3) in which data are generated from a Tweedie model is put in Appendix B.

Fitting Grabit, TDboost and EMTboost models to these data sets, we get the final predictor function $\hat{F}(\cdot)$ and parameter estimators. Then we make a prediction about the pure premium by applying the predictor functions on an independent held-out testing set to find estimated expectation: $\hat{\mu}(\mathbf{x}) = \mathbb{E}(y|\mathbf{x})$. For the three competing models, the predicted pure premium is given by equations

$$\begin{aligned}
\hat{\mu}^{\text{Grabit}}(\mathbf{x}) &= \varphi\big(-\hat{F}_{\text{Grabit}}(\mathbf{x})\big) + \hat{F}_{\text{Grabit}}(\mathbf{x})\big(1-\Phi\big(-\hat{F}_{\text{Grabit}}(\mathbf{x})\big)\big),\\
\hat{\mu}^{\text{TDboost}}(\mathbf{x}) &= \exp\big(\hat{F}_{\text{TDboost}}(\mathbf{x})\big),\\
\hat{\mu}^{\text{EMTboost}}(\mathbf{x}) &= \exp\big(\hat{F}_{\text{EMTboost}}(\mathbf{x})\big),
\end{aligned}$$

where $\varphi(\cdot)$ is the probability density function of the standard normal distribution and $\Phi(\cdot)$ is its cumulative distribution function. The predicted pure premium of the Grabit model is derived in detail in Appendix A. As the true model is known in simulation settings, we can compare the difference between the predicted premiums and the expected true losses. For the zero-inflated Tobit model in case 2, the expected true loss is given by $\mathbb{E}_{\text{ZIF-Tobit}}\big[y|F(\mathbf{x})\big] = q[\varphi(-F(\mathbf{x})) + F(\mathbf{x})(1-\Phi(-F(\mathbf{x})))]$, with q being the probability that response y comes from the Tobit model and $F(\mathbf{x})$ the true target function. For our zero-inflated Tweedie mode, the expected true loss is $\mathbb{E}_{\text{ZIF-Tw}}\big[y|F(\mathbf{x})\big] = q\exp(F(\mathbf{x}))$.

4.1. Measurement of prediction accuracy

Given a portfolio of policies $\mathbf{D} = \{(y_i, \mathbf{x_i}, \omega_i)\}_{i=1}^n$, y_i is the claim cost for the ith policy and $\hat{y}_i$ is denoted as the predicted claim cost. We consider the following three measurements of prediction accuracy of $\{\hat{y}_i\}_{i=1}^n$.

GiniIndex(Gini[a]) Gini index is a well-accepted tool to evaluate the performance of predictions. There exists many variants of Gini index and one variant we use is denoted by Gini^a: for a sequence of numbers $\{s_1, ..., s_n\}$, let $R(s_i) \in \{1, ..., n\}$ be the rank of s_i in the sequence in an increasing order. To break the ties when calculating the order, we use the **LAST** tie-breaking method, i.e., we set $R(s_i) > R(s_j)$ if $s_i = s_j$, $i < j$. Then the normalized Gini index is referred to as:

$$\text{Gini}^a = \frac{\frac{\sum_{i=1}^n y_i R(\hat{y}_i)}{\sum_{i=1}^n y_i} - \sum_{i=1}^n \frac{n-i+1}{n}}{\frac{\sum_{i=1}^n y_i R(y_i)}{\sum_{i=1}^n y_i} - \sum_{i=1}^n \frac{n-i+1}{n}}.$$

Note that this criterion only depends on the rank of the predictions and larger Gini^a index means better prediction performance.

GiniIndex(Gini[b]) We exploit a popular alternative–the ordered Lorentz curve and the associated Gini index (denoted by Gini^b; Frees, Meyers, and Cummings 2011, 2014) to capture the discrepancy between the expected premium $P(\mathbf{x}) = \hat{\mu}(\mathbf{x})$ and the true losses y. We successively specify the prediction from each model as the base premium and use predictions from the remaining models as the competing premium to compute the Gini^b indices. Let $B(\mathbf{x})$ be the "base premium" and $P(\mathbf{x})$ be the "competing premium". In the ordered Lorentz 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

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

and the ordered loss distribution is

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

Then the ordered Lorentz curve is the graph of $(\hat{D}_P(s), \hat{D}_L(s))$. Twice the area between the ordered Lorentz curve and the line of equality measures the discrepancy between the premium and loss distributions, and is defined as the Gini^b index.

MeanAbsoluteDeviation(**MAD**) Mean Absolute Deviation with respect to the true losses $\{y_i\}_{i=1}^n$ is defined as $\frac{1}{n}\sum_{i=1}^n |y_i - \hat{y}_i|$. In the following simulation studies, we can directly compute the mean absolute deviation between the predicted losses $\{\hat{y}_i\}_{i=1}^n$ and the expected true losses $\{\mathbb{E}[y_i|\mathbf{x}_i]\}_{i=1}^n$ to obtain $\frac{1}{n}\sum_{i=1}^n |\mathbb{E}[y_i|\mathbf{x}_i] - \hat{y}_i|$, while in the real data study, we can only compute the MAD against true losses $\{y_i\}_{i=1}^n$.

4.2. Case 1

In this simulation case, we generate data from the zero-inflated Tweedie models with two different target functions: one with two interactions and the other generated from Friedman's (2001) "random function generator" (RFG) model. We fit the training data using Grabit, TDboost, and

Table 1. Simulation results for case 1.1 with MADs.

	Competing models				
			EMTboost		
q	TDboost	Grabit	$\rho = 1.5$	$\rho = 1.7$	Tuned ρ
1.00	0.597 (.013)	0.746 (.029)	0.594 (.016)	0.598 (.012)	0.598 (.015)
0.85	0.565 (.015)	0.761 (.032)	0.554 (.017)	0.555 (.017)	0.562 (.016)
0.75	0.561 (.018)	0.706 (.026)	0.489 (.010)	0.485 (.011)	0.503 (.010)
0.50	0.454 (.024)	0.674 (.044)	0.365 (.012)	0.375 (.014)	0.361 (.012)
0.25	0.301 (.013)	0.382 (.019)	0.240 (.010)	0.242 (.011)	0.237 (.010)
0.10	0.135 (.005)	0.169 (.009)	0.122 (.004)	0.124 (.004)	0.124 (.004)

EMTboost. In all numerical studies, five-fold cross-validation is adopted to select the optimal ensemble size M and regression tree size L, while the shrinkage factor ν is set as 0.001.

4.2.1. Two interactions function (case 1.1)

In this simulation study, we demonstrate the performance of EMTboost to recover the mixed data distribution that involves exact zero mass, and the robustness of our model in terms of premium prediction accuracy when the index parameter ρ is misspecified. We consider the true target function with two hills and two valleys:

$$F(x_1, x_2) = e^{-5(1-x_1)^2+x_2^2} + e^{-5x_1^2+(1-x_2)^2}, \tag{17}$$

which corresponds to a common scenario where the effect of one variable changes depending on the effect of the other. The response Y follows a zero-inflated Tweedie distribution ZIF-Tw(μ, ϕ, ρ, q) with Tweedie portion probability q:

$$Y \sim \begin{cases} Z, & \text{with probability } q, Z \sim \text{Tw}(\mu, \phi, \rho), \\ 0, & \text{with probability } 1-q, \end{cases}$$

where

$$\mu = \exp\left(F(x_1, x_2)\right), x_1, x_2 \overset{\text{ind.}}{\sim} \text{Unif}(0, 1),$$

with $\phi = 1, \rho = 1.5$ and q chosen from a decreasing sequence of values: $q \in \{1,\ 0.85,\ 0.75,\ 0.50,\ 0.25,\ 0.10\}$.

We generate $n = 500$ observations $\{\mathbf{x}_i, y_i\}_{i=1}^n$ for training and $n' = 1200$ for testing, and fit the training data using Grabit, TDboost and EMTboost models. The true target functions are known, and we use MAD (against expected true premium) and Gini[a] index as performance criteria.

When fitting EMTboost, we design three scenarios to illustrate the robustness of our method w.r.t. ρ. In the first scenario, set $\rho = 1.5$, which is the true value. In the second scenario, set $\rho = 1.7$, which is misspecified. In the last scenario, we use the profile likelihood method to estimate ρ.

The resulting MADs and Gini[a] indices of the three competing models on the held-out testing data are reported in Tables 1 and 2, which are averaged over 20 independent replications for each q. Boxplots of MADs comparing Grabit, TDboost and EMTboost (with estimated ρ) are shown in Figure 1. In all three scenarios, EMTboost outperforms Grabit and TDboost in terms of the ability to recover the expected true premium by giving smallest MADs and largest Gini[a] indices, especially when zeros inflate: $q \in \{0.5, 0.25, 0.1\}$. The prediction performance of EMTboost when $\rho = 1.7$ is not much worse than that when $\rho = 1.5$, showing that the choice of ρ has relatively small effect on estimation accuracy.

Table 2. Simulation results for case 1.1 with Gini[a] indices.

	Competing models				
			EMTboost		
q	TDboost	Grabit	$\rho = 1.5$	$\rho = 1.7$	Tuned ρ
1.00	0.480 (.008)	0.449 (.011)	0.481 (.006)	0.481 (.006)	0.481 (.006)
0.85	0.393 (.008)	0.354 (.009)	0.397 (.007)	0.397 (.007)	0.397 (.007)
0.75	0.343 (.009)	0.300 (.020)	0.363 (.008)	0.365 (.007)	0.361 (.008)
0.50	0.242 (.012)	0.186 (.016)	0.289 (.011)	0.288 (.012)	0.292 (.011)
0.25	0.172 (.016)	0.116 (.020)	0.219 (.016)	0.215 (.017)	0.217 (.015)
0.10	0.085 (.028)	0.107 (.023)	0.137 (.027)	0.122 (.028)	0.136 (.025)

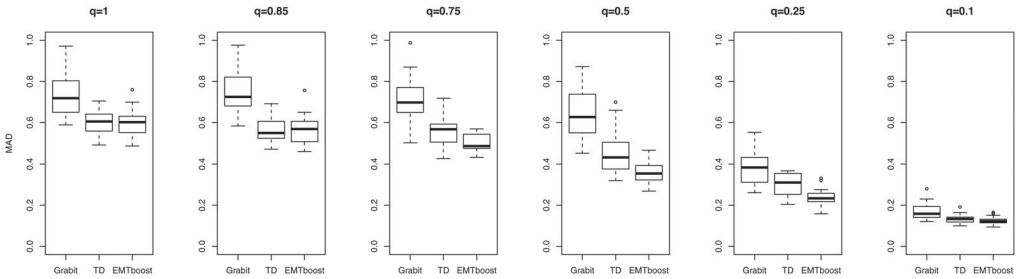

Figure 1. Simulation results for case 1.1: comparing MADs of Grabit, TDboost and EMTboost with decreasing q. Boxplots display empirical distributions of MADs based on 20 independent replications.

4.2.2. Random function generator (case 1.2)

In this case, we compare the performance of the three competing models in various complicated and randomly generated predictor functions. We use the RFG model whose true target function F is randomly generated as a linear expansion of functions $\{g_k\}_{k=1}^{20}$:

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

Here, each coefficient b_k is a uniform random variable from Unif$[-1, 1]$. 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 p-dimensional variable $\mathbf{x}$ in the form

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

where each ψ_k is an independent permutation of the integers $\{1, ..., p\}$. 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 interaction presented in each $g_k(\mathbf{z}_k)$ is between four and five. Each function $g_k(\mathbf{z}_k)$ is a p_k-dimensional Gaussian function:

$$g_k(\mathbf{z}_k) = \exp\left\{-\frac{1}{2}(\mathbf{z}_k - \mathbf{u}_k)^{\mathrm{T}}\mathbf{V}_k(\mathbf{z}_k - \mathbf{u}_k)\right\},$$

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

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

where $\mathbf{U}_k$ is a random orthonormal matrix, $\mathbf{D}_k = \mathrm{diag}\{d_k[1], ..., d_k[p_k]\}$, and the square root of each diagonal element $\sqrt{d_k[j]}$ is a uniform random variable from Unif$[0.1, 2.0]$. We generate data $\{y_i, \mathbf{x}_i\}_{i=1}^{n}$ from zero-inflated Tweedie distribution where $\mathbf{x}_i \sim \mathrm{N}(0, \mathbf{I}_p)$, $\mu_i = \exp\{F(\mathbf{x}_i)\}, i = 1, ..., n$.

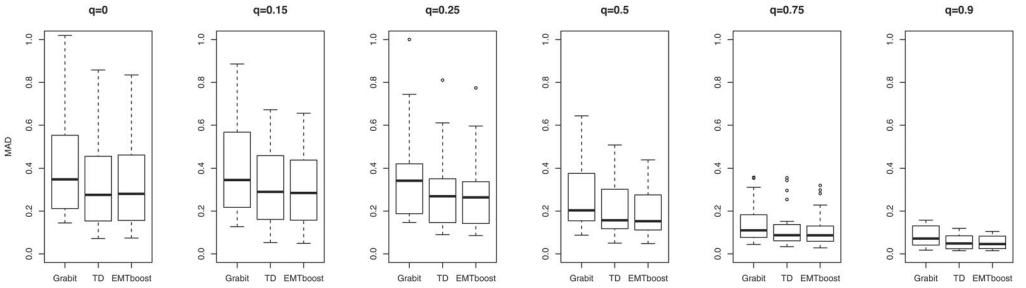

Figure 2. Simulation results for case 1.2: comparing MADs of Grabit, TDboost and EMTboost with decreasing q. Boxplots display empirical distributions of the MADs based on 20 independent replications.

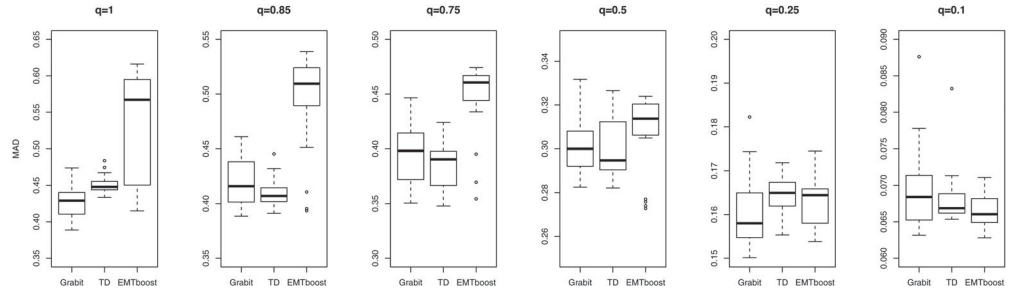

Figure 3. Simulation results for case 2.1: comparing the MADs of Grabit, TDboost and EMTboost with decreasing q. Boxplots display empirical distributions of MADs based on 20 independent replications.

We randomly generate 20 sets of samples with $\phi = 1$ and $\rho = 1.5$, each sample having 2000 observations, 1000 for training and 1000 for testing. When fitting EMTboost for each $q \in \{1, 0.85, 0.75, 0.5, 0.25, 0.1\}$, the estimates of Tweedie portion probability have mean $\overline{q^*} = 0.96, 0.79, 0.71, 0.53, 0.28, 0.13$. Figure 2 shows simulation results comparing the MADs of Grabit, TDboost and EMTboost. We can see, in all the cases, EMTboost outperforms Grabit and becomes very competitive compared to TDboost when q decreases.

4.3. Case 2

In this simulation case, we generate data from the zero-inflated Tobit models with two target functions similar to that of case 1. For all three gradient-tree boosting models, five-fold cross-validation is adopted for developing trees. Profile likelihood method is used again.

4.3.1. Two interactions function (case 2.1)

In this simulation study, we compare the performance of three models in terms of MADs. Consider the data generated from the zero-inflated Tobit model where the true target function is given by

$$F(x_1, x_2) = 2\cos\left(2.4\pi(|x_1|^3 + |x_2|^3)^{0.5}\right).$$

Conditional on covariates $\mathbf{X} = (X_1, X_2)$, the latent variable Y^* follows a Gaussian distribution:

$$Y^* = F(X_1, X_2) + \epsilon, \;\; X_k \text{ i.i.d.} \sim \text{Unif}(-1, 1), \;\; k = 1, 2, \;\; \epsilon \sim N(0, 1).$$

The Tobit response Y_{Tobit} can be expressed as $Y_{\text{Tobit}} = \max(Y^*, 0)$, and we generate the zero-inflated Tobit data using the Tobit response:

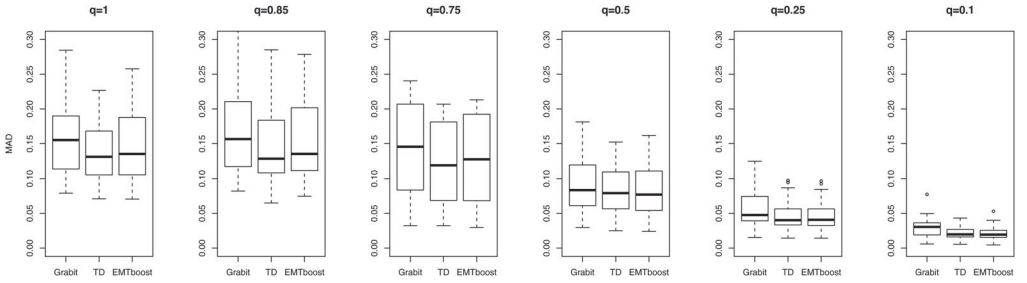

Figure 4. Simulation results for case 2.2: comparing the MADs of Grabit, TDboost and EMTboost when decreasing q. Boxplots display empirical distributions of MADs based on 20 independent replications.

$$Y \sim \begin{cases} Y_{\text{Tobit}}, & \text{with probability } q, \\ 0, & \text{with probability } 1-q. \end{cases} \tag{18}$$

where q takes value from the sequence $\{1, 0.85, 0.75, 0.5, 0.25, 0.1\}$.

We generate $n = 500$ observations for training and $n' = 4500$ for testing. Figure 3 shows simulation results when comparing MADs of Grabit, TDboost and EMTboost based on 20 independent replications. We can see from the first boxplot that when $q = 1$, zero-inflated Tobit distribution degenerates to a Tobit distribution, and not surprisingly, Grabit outperforms EMTboost in MADs. As q decreases, meaning the proportion of zeros increases, the prediction performance of EMTboost gets improved. When the exact zero mass probability is $1 - q = 0.9$, the averaged MADs of the three models are $\overline{\text{MAD}}_{\text{Grabit}} = 0.0697, \overline{\text{MAD}}_{\text{TDboost}} = 0.0681$, $\overline{\text{MAD}}_{\text{EMTboost}} = 0.0664$, with EMTboost performing the best.

4.3.2. Random function generator (case 2.2)

We again use the RFG model in this simulation. The true target function F is randomly generated as given in Section 4.2.2. The latent variable Y^* follows

$$Y^* = F(\mathbf{x}_i) + \epsilon, \;\; \mathbf{x}_i \sim \text{N}(0, \mathbf{I}_p), \;\; \epsilon \sim \text{N}(0,1), i = 1, ..., n.$$

We set $Y_{\text{Tobit}} = \max(Y^*, 0)$ and generate the data following the zero-inflated Tobit model (18) with Tobit portion probability $q \in \{1, 0.85, 0.75, 0.5, 0.25, 0.1\}$. We randomly generate 20 sets of sample from the zero-inflated Tobit model for each q, and each sample contains 2000 observations, 1000 for training and 1000 for testing. Figure 4 shows MADs of Grabit, TDboost and EMTboost as Boxplots. Interestingly, for all the q's, TDboost and EMTboost outperform Grabit even though the true model is a Tobit model. The MAD of EMTboost becomes better when q decreases, and is competitive with that of TDboost when $q = 0.1$: the averaged MADs of the three models are $\overline{\text{MAD}}_{\text{Grabit}} = 0.0825$, $\overline{\text{MAD}}_{\text{TDboost}} = 0.0565, \overline{\text{MAD}}_{\text{EMTboost}} = 0.0564$. As for the averaged Gini^a indices, EMTboost performs the best when $q = 0.1$: $\overline{\text{Gini}^a}_{\text{Grabit}} = 0.0463$, $\overline{\text{Gini}^a}_{\text{TDboost}} = 0.0816, \overline{\text{Gini}^a}_{\text{EMTboost}} = 0.1070$.

5. Application: automobile claims

5.1. Data set

We consider the auto-insurance claim data set as analyzed in Yip and Yau (2005) and Zhang and Yu (2005). The data set contains 10,296 driver vehicle records, each including an individual driver's total claim amount (z_i) and 17 characteristics $\mathbf{x}_i = (x_{i,1}, ..., x_{i,17})$ for the driver and insured vehicle. We want to predict the expected pure premium based on $\mathbf{x}_i$. The description statistics of the data are provided in Yang, Qian, and Zou (2018). Approximately 61.1% of policyholders had

Table 3. Real – Zero percentage w.r.t. λ.

λ	1	0.75	0.50	0.25	0.15	0.10	0.05
Zero Percentage	61.1%	67.7%	75.9%	86.3%	91.3%	94.0%	96.9%

no claims, and 29.6% of the policyholders had a positive claim amount up to $10,000. Only 9.3% of the policy-holders had a high claim amount above $10,000, but the sum of their claim amount made up to 64% of the overall sum. We use this original data set to synthesize the often more realistic scenarios with extremely unbalanced zero-inflated data sets.

Specifically, we randomly under-sample (without replacement) from the nonzero-claim data with certain fraction λ to increase the percentage of the zero-claim data. For example, if we set the under-sampling fraction as $\lambda = 0.15$, then the percentage of the non-claim policyholders will become approximately $61.1/(61.1 + 38.9\lambda) = 91.28\%$. We choose a decreasing sequence of under-sampling fractions $\lambda \in \{1, 0.75, 0.5, 0.25, 0.15, 0.1\}$. For each λ, we randomly under-sample the positive-loss data without replacement and combine these nonzero-loss data with the zero-loss data to generate a new data set. Then we separate this new data set into two sets uniformly for training and testing. The corresponding percentages of zero-loss data among the new data set w.r.t. different λ are presented in Table 3. The Grabit, TDboost and EMTboost models are fitted on the training set and their estimators are obtained with five-fold cross-validation.

5.2. Performance comparison

To compare the performance of Grabit, TDboost and EMTboost models, we predict the pure premium $P(\mathbf{x})$ by applying each model on the held-out testing set. Since the losses are highly right-skewed, we use the ordered Lorentz curve and the associated Gini^b index described in Sec. 4.1 to capture the discrepancy between the expected premiums and true losses.

The entire procedure of under-sampling, data separating and Gini^b index computation are repeated 20 times for each λ. A sequence of matrices of the averaged Gini^b indices and standard errors w.r.t. each under-sampling fraction λ are presented in Table 4. We then follow the "minimax" strategy (Frees, Meyers, and Cummings 2014) to pick the "best" base premium model that is least vulnerable to the competing premium models. For example, when $\lambda = 0.15$, the maximal Gini^b index is 40.381 when using $B(\mathbf{x}) = \hat{\mu}^{\text{Grabit}}(\mathbf{x})$ as the base premium, 36.735 when $B(\mathbf{x}) = \hat{\mu}^{\text{TDboost}}(\mathbf{x})$, and -22.674 when $B(\mathbf{x}) = \hat{\mu}^{\text{EMTboost}}(\mathbf{x})$. Therefore, EMTboost has the smallest maximum Gini^b index at -22.674, hence having the best performance. Figure 5 also shows that when Grabit (or TDboost) is selected as the base premium, EMTboost represents the most favorable choice.

After computing the Gini^b index matrix and using the "minimax" strategy to choose the best candidate model, we count the frequency, out of 20 replications, of each model chosen as the best model and record the ratio of their frequencies. The results w.r.t each λ are demonstrated in Figure 6. From Table 4 and Figure 6, we find that when λ decreases, the performance of EMTboost gradually outperforms that of TDboost in terms of averaged Gini^b indices and the corresponding model-selection ratios. In particular, TDboost outperforms EMTboost when $\lambda = 1, 0.75, 0.5$, and EMTboost outperforms TDboost when $\lambda = 0.25, 0.15, 0.1, 0.05$. When $\lambda = 0.25, 0.15, 0.1$, EMTboost has the largest model-selection ratio among the three.

We also find that TDboost and EMTboost both outperform Grabit when $\lambda = 1$, 0.75, 0.5, 0.25, 0.15, 0.1, but Grabit becomes the best when $\lambda = 0.05$; interestingly, if we compare MAD results in Table 5, the prediction error of EMTboost becomes the smallest for each λ. This inconsistent results between the criteria MAD and Gini^b index when $\lambda = 0.05$ can be explained by the different learning characteristics of the EMTboost methods and the Grabit methods. To see it more clearly, we compute the MADs on the positive-loss dataset, denoted by MAD$^+$, and zero-loss dataset,

Table 4. Grabit, TDboost and EMTboost Gini^b indices.

	Competing premium		
Base Premium	Grabit	TDboost	EMTboost
		$\lambda = 1$	
Grabit	0	11.459 (0.417)	11.402 (0.386)
TDboost	**6.638** (0.409)	0	0.377 (0.355)
EMTboost	7.103 (0.357)	2.773 (0.414)	0
		$\lambda = 0.75$	
Grabit	0	14.955 (0.413)	15.162 (0.435)
TDboost	**5.466** (0.425)	0	1.848 (0.504)
EMTboost	6.152 (0.385)	2.622 (0.560)	0
		$\lambda = 0.50$	
Grabit	0	25.047 (1.539)	25.621 (1.492)
TDboost	3.516 (0.963)	0	**4.056** (0.651)
EMTboost	5.702 (0.698)	2.525 (0.501)	0
		$\lambda = 0.25$	
Grabit	0	51.502 (1.062)	51.581 (1.005)
TDboost	−18.248 (2.445)	0	20.035 (2.414)
EMTboost	1.283 (2.593)	**3.929** (2.544)	0
		$\lambda = 0.15$	
Grabit	0	37.290 (2.505)	40.381 (1.822)
TDboost	−23.569 (2.607)	0	36.735 (3.188)
EMTboost	**−22.674** (1.975)	−22.926 (2.604)	0
		$\lambda = 0.10$	
Grabit	0	−1.189 (5.828)	16.721 (5.120)
TDboost	14.581 (6.587)	0	35.298 (3.026)
EMTboost	**−2.742** (4.884)	−20.080 (3.572)	0
		$\lambda = 0.05$	
Grabit	0	−16.851 (2.662)	**−8.652** (3.059)
TDboost	42.493 (3.792)	0	27.754 (3.784)
EMTboost	32.169 (3.767)	−13.448 (3.551)	0

The "best" base premium models are emphasized.

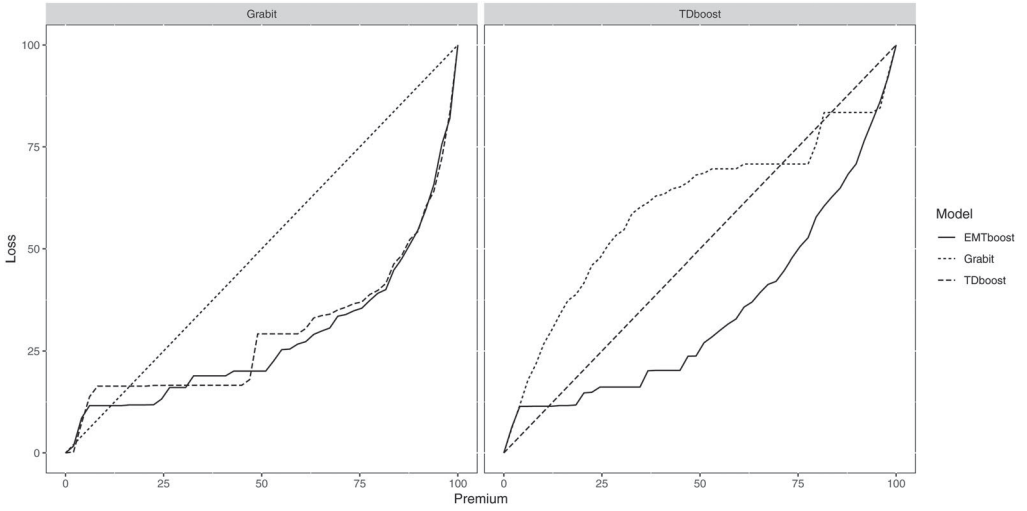

Figure 5. The ordered Lorentz curves for the synthetic data on a single replication when $\lambda = 0.15$. Grabit (or TDboost) is set as the base premium and the EMTboost is the competing premium. The ordered Lorentz curve of EMTboost is below the line of equality when choosing Grabit or TDboost as the base premium.

denoted by MAD^0 separately, and compute the Gini^a indices on the nonzero dataset, denoted by Gini^{a+}. When $\lambda = 0.05$, EMTboost obtains the smallest averaged MAD on zero-loss dataset ($\overline{\text{MAD}^0}_{\text{EMTboost}} = 0.146 < \overline{\text{MAD}^0}_{\text{TDboost}} = 0.269 < \overline{\text{MAD}^0}_{\text{Grabit}} = 0.422$), while Grabit obtains the

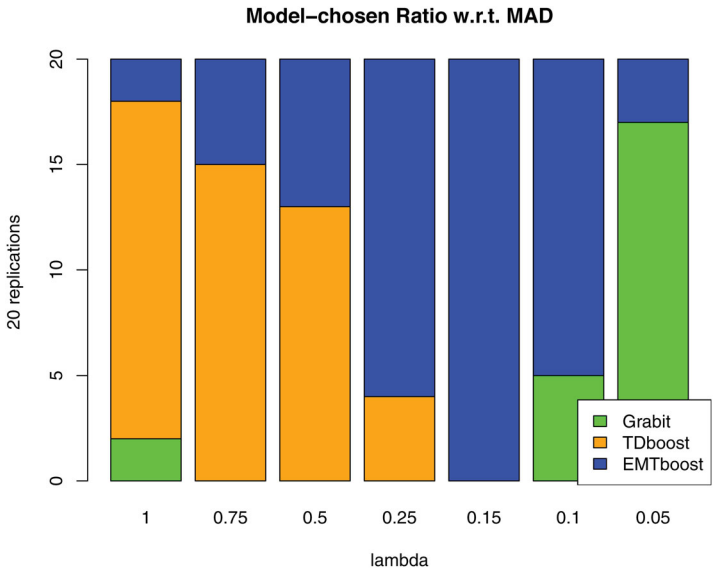

Figure 6. Bar chart of model-selection ratio among Grabit, TDboost and EMTboost w.r.t. Gini^b indices under 20 independent replications when λ increases.

Table 5. Comparing Grabit, TDboost and EMTboost with MADs.

	Competing models		
λ	Grabit	TDboost	EMTboost
1.00	4.248 (.014)	4.129 (.012)	4.067 (.012)
0.75	3.879 (.017)	3.679 (.011)	3.622 (.012)
0.50	3.345 (.026)	2.994 (.017)	2.928 (.016)
0.25	2.439 (.014)	1.945 (.021)	1.766 (.014)
0.15	1.720 (.015)	1.489 (.023)	1.309 (.019)
0.10	1.265 (.011)	1.100 (.015)	0.986 (.014)
0.05	0.714 (.012)	0.578 (.015)	0.402 (.009)

smallest averaged MAD on positive-loss dataset ($\overline{\text{MAD}^+}_{\text{EMTboost}} = 10.406 > \overline{\text{MAD}^+}_{\text{TDboost}} = 10.297 > \overline{\text{MAD}^+}_{\text{Grabit}} = 9.927$). The MAD^0 performance shows that EMTboost captures the zero information much better than TDboost and Grabit. The somewhat worse MAD$^+$ performance of EMTboost when $\lambda = 0.05$ can be explained by the deficiency of the nonzero data points (only about 100 nonzeros comparing with over 3000 zeros); if we fix the nonzero sample size with under-sampling fraction $\lambda = 0.2$, and at the same time, over-sample the zero-loss part with over-sampling fraction $\eta = 3$ to obtain about 96% zero proportion, then the averaged Gini^b results summarized in Table C3 in Appendix C.2 indeed show that EMTboost remains to perform competitively compared with the other methods under this large zero proportion setting.

6. Concluding remarks

We have proposed and studied the EMTboost model to handle very unbalanced claim data with excessive proportion of zeros. Our proposal overcomes the difficulties that traditional Tweedie model have when handling these common data scenarios, and at the same time, preserves the flexibility of nonparametric models to accommodate complicated nonlinear and high-order interaction relations. We also expect that our zero-inflated Tweedie approach can be naturally extended to high-dimensional linear settings (Qian, Yang, and Zou 2016). It remains interesting

to develop extended approaches to subject-specific zero-inflation settings, and provide formal procedures that can conveniently test if zero-inflated Tweedie model is necessary in data analysis compared to its simplified alternatives under both parametric and nonparametric frameworks.

Acknowledgment

We sincerely thank the Editor, the Associate Editor and anonymous reviewers for their valuable comments.

Funding

Yang's research is partially supported by NSERC RGPIN-2016-05174 and FRQ-NT NC-205972. Qian's research is partially supported by NSF DMS-1916376 and JMPC Faculty Fellowship.

Appendices for "Tweedie gradient boosting for extremely unbalanced zero-inflated data"

Appendix A: Tobit model (truncated normal distribution)

Suppose the latent variable Y^* follows, conditional on covariate $\mathbf{x}$, a Gaussian distribution:

$$Y^*|\mathbf{x} \sim N(\mu(\mathbf{x}), \sigma^2)$$

This latent variable Y^* is observed only when it lies in an interval $[y_l, y_u]$. Otherwise, one observes y_l or y_u depending on whether the latent variable is below the lower threshold y_l or above the upper threshold y_u, respectively. Denoting Y as the observed variable, we can express it as:

$$Y = \begin{cases} y_l, & \text{if } Y^* \le y_l, \\ Y^*, & \text{if } y_l < Y^* < y_u, \\ y_u, & \text{if } y_u \le Y^*. \end{cases}$$

The density of Y is given by:

$$\begin{aligned} f_{\text{Tobit}}(y; \mu(\mathbf{x}), \sigma) = \Phi\left(\frac{y_l - \mu(\mathbf{x})}{\sigma}\right)\mathbf{I}_{y_l}(y) + \left(1 - \Phi\left(\frac{y_u - \mu(\mathbf{x})}{\sigma}\right)\right)\mathbf{I}_{y_u}(y). \\ + \frac{1}{\sigma}\varphi\left(\frac{y - \mu(\mathbf{x})}{\sigma}\right)\mathbf{I}\{y_l < y < y_u\} \end{aligned}$$

Then the expectation of $Y|\mathbf{x}$ is given by:

$$\begin{aligned} \mathbb{E}_\sigma[y|\mathbf{x}] &= \int_{-\infty}^{+\infty} y f_{\text{Tobit}}(y; \mu(\mathbf{x}), \sigma) dy \\ &= y_l\Phi(\alpha) + \int_{y_l}^{y_u} y\frac{1}{\sigma}\varphi\left(\frac{y - \mu(\mathbf{x})}{\sigma}\right)dy + y_u(1 - \Phi(\beta)) \\ &= y_l\Phi(\alpha) + \int_{\alpha}^{\beta} (s\sigma + \mu(\mathbf{x}))\varphi(s)ds + y_u(1 - \Phi(\beta)) \\ &= y_l\Phi(\alpha) + \sigma(\varphi(\alpha) - \varphi(\beta)) + \mu(\mathbf{x})(\Phi(\beta) - \Phi(\alpha)) + y_u(1 - \Phi(\beta)), \end{aligned}$$

where $\alpha = \frac{y_l - \mu(\mathbf{x})}{\sigma}, \beta = \frac{y_u - \mu(\mathbf{x})}{\sigma}$. And

$$\phi(\xi) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\xi^2\right)$$

is the probability density function of the standard normal distribution and $\Phi(\cdot)$ is its cumulative distribution function:

$$\Phi(y) = \int_{-\infty}^{y} \varphi(\xi)d\xi$$

In simulation 2, the latent variable is truncated by 0 from below, i.e., $y_l = 0, y_u = \infty$. So we have $\varphi(\beta) = 0, \Phi(\beta) = 1$. We also set the variance of the Gaussian distribution as $\sigma = 1$. Then the expectation of $Y|\mathbf{x}$ is given by:

$$\mathbb{E}_{\sigma=1}[y|\mathbf{x}] = \varphi(-\mu(\mathbf{x})) + \mu(\mathbf{x})(1 - \Phi(-\mu(\mathbf{x}))).$$

Appendix B: case 3

In this simulation study, we demonstrate that our EMTboost model can fit the non-zero-inflated dataset well. We consider the data generated from the Tweedie model, with the true target function (17). We generate response Y from the Tweedie distribution $\mathrm{Tw}(\mu, \phi, \rho)$, with $\mu = \exp(F(x_1, x_2)), x_1, x_2 \sim \mathrm{Unif}(0,1)$ and the index parameter $\rho = 1.5$. We find that when the dispersion parameter ϕ takes large value in $\mathbb{R}^+$, Tweedie's zero mass probability $\mathbb{P}(Y_{\mathrm{Tw}} = 0) = \exp\left(-\frac{1}{\phi}\frac{\mu^{2-\rho}}{2-\rho}\right)$ gets closer to 1. So we choose three large dispersion values $\phi \in \{20, 30, 50\}$.

We generate $n = 500$ observations $\{\mathbf{x}_i, y_i\}_{i=1}^n$ for training and $n' = 1000$ for testing, and fit the training data using Grabit, TDboost and EMTboost models. For all three models, five-fold cross-validation is adopted and the shrinkage factor ν is set as 0.001. The profile likelihood method is used.

The discrepancy between the predicted loss and the expected true loss in criteria MAD and Gini^b index are shown in Tables B1 and B2, which are averaged over 20 independent replications for each ϕ. Table B1 shows that EMTboost obtains the smallest MAD. In terms of Gini^b indices, EMTboost is also chosen as the "best" model for each ϕ.

In this setting, $\mathbb{P}(Y_{\mathrm{Tw}} = 0) \approx 0.83, 0.88, 0.91$, which means that all the customers are generally very likely to have no claim. The assumption of our EMTboost model with a general exact zero mass probability coincides with this data structure. Its zero mass probability estimation $1 - \hat{q}$ is 0.863, 0.909 and 0.943 respectively, showing that EMTboost learns this zero part of information quite well. As a result, EMTboost performs no worse than TDboost, which is based on the true model assumption.

Table B1. Simulation results for case 3 with MADs.

	Competing models		
ϕ	Grabit	TDboost	EMTboost
20	2.499 (.072)	2.465 (.057)	2.449 (.054)
30	2.442 (.068)	2.492 (.060)	2.442 (.060)
50	2.553 (.091)	2.560 (.082)	2.544 (.080)

Table B2. Simulation results for case 3 with Gini^b indices.

	Competing premium		
Base premium	Grabit	TDboost	EMTboost
		$\phi = 20$	
Grabit	0	8.205 (3.492)	8.052 (3.471)
TDboost	4.487 (2.462)	0	4.104 (2.283)
EMTboost	**3.153** (1.982)	2.347 (1.656)	0
		$\phi = 30$	
Grabit	0	4.273 (3.447)	3.244 (3.411)
TDboost	4.017 (3.257)	0	3.454 (3.191)
EMTboost	**1.404** (2.730)	0.846 (2.403)	0
		$\phi = 50$	
Grabit	0	5.452 (4.806)	10.362 (4.801)
TDboost	3.479 (3.654)	0	2.476 (3.899)
EMTboost	−0.836 (2.785)	**1.031** (3.354)	0

The "best" base premium models are emphasized based on the matrices of averaged Gini^b indices.

Appendix C: real data

Appendix C.1: implemented EMTboost: penalization on q

When fitting the EMTboost model, we want to avoid the situation that the Tweedie portion probability estimation $\hat{q}$ degenerates to 0. So we add a regularization term $r\log(1-q)$ to the Q-function, as Equation (16) in Sec. 3.2. We choose an increasing sequence of penalty parameter $\log_{10}(r) \in \{-2, \ldots, 1\}_{41}$, and use the strategy of "warm start" to improve the computation efficiency, i.e., setting the current solution $(\hat{\mu}(r_l), \hat{\phi}(r_l), \hat{q}(r_l))$ as the initialization for the next solution $(\hat{\mu}(r_{l+1}), \hat{\phi}(r_{l+1}), \hat{q}(r_{l+1}))$.

We use this implemented EMTboost model to run the penalized solution paths with respect to the sequence of penalty parameter r on the extremely zero-inflated training data ($\lambda = 0.05$) in sec. (5.1) under 20 independent replications, and then apply the estimators to the testing data to compute the discrepancy under MAD. We also compute the MADs on zero-loss dataset (MAD^0) and positive-loss dataset (MAD$^+$) separately, and the Gini^a indices

Table C1. Implemented EMTboost ($\lambda = 0.05$) results with MAD s, MAD^{0}s and MAD$^+$s.

log(r)	−2.00	−1.10	−0.65	0.25	0.70
MAD	0.432 (0.047)	0.377 (0.036)	0.351 (0.033)	0.330 (0.032)	0.328 (0.031)
MAD^0	0.114 (0.026)	0.054 (0.011)	0.026 (0.005)	0.004 (0.001)	0.002 (0.000)
MAD$^+$	10.452 (1.017)	10.527 (1.017)	10.560 (1.017)	10.587 (1.017)	10.590 (1.017)

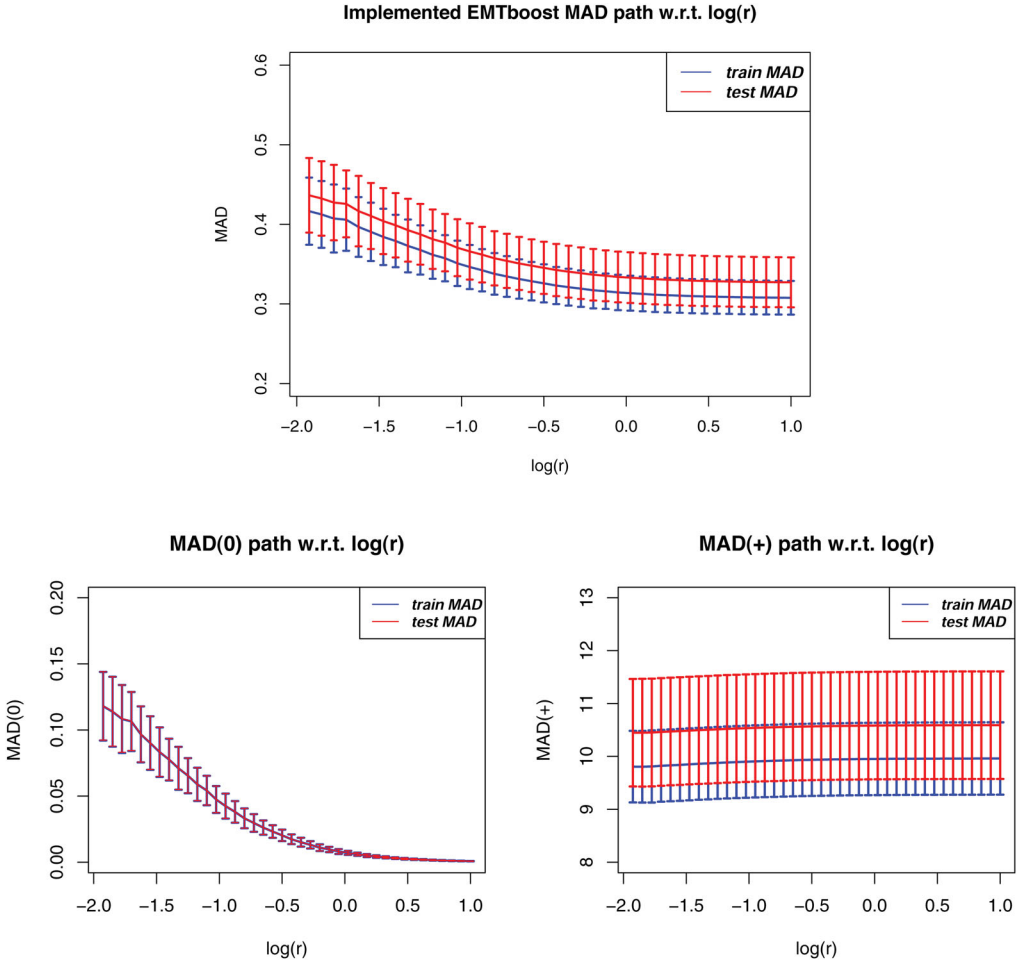

Figure 7. Implemented EMTboost ($\lambda = 0.05$) results: MAD error path with its one standard deviation when increasing regularization parameter r. The blue lines are the training error lines, and the red ones are the testing error lines. Top figure: averaged MAD drops when penalty parameter increases. Bottom left: averaged MAD^0 and its standard deviation drop remarkably and approximate 0 when r increases. Bottom right: averaged MAD$^+$ is flat and increases a little when r increases. All the averaged testing MADs are within one standard deviation of the averaged training MADs.

Table C2. Comparing Grabit, TDboost, EMTboost and implemented EMTboost ($r = 1/6$) with Gini^q indices and MADs.

$\lambda = 0.05$	Grabit	TDboost	EMTboost	Implemented EMTboost
$\hat{q}$	–	–	0.697 (.018)	0.132 (.003)
Gini^q	0.415 (.023)	0.134 (.040)	0.238 (.033)	0.261 (.033)
Gini^{q+}	0.104 (.030)	−0.116 (.039)	−0.164 (.031)	−0.103 (.040)
MAD	0.714 (.012)	0.578 (.015)	0.482 (.013)	0.356 (.008)
MAD^0	0.422 (.009)	0.269 (.012)	0.146 (.008)	0.032 (.002)
MAD$^+$	9.927 (.223)	10.287 (.226)	10.406 (.228)	10.551 (.227)

Table C3. Grabit, TDboost, EMTboost Gini^b indices with $\lambda = 0.2$ and $\eta = 3$.

$\lambda = 0.2$, $\eta = 3$	Competing premium		
Base premium	Grabit	TDboost	EMTboost
Grabit	0	23.872 (3.928)	35.616 (2.655)
TDboost	–4.822 (2.822)	0	27.234 (2.152)
EMTboost	**–5.583** (1.599)	–14.806 (1.681)	0

on positive-loss dataset (Gini^{a+}). Table C1 and Figure 7 shows the implemented EMTboost MAD path w.r.t. the logarithm of a sequence of penalty parameter r.

Table C2 shows the results of Grabit, TDboost, EMTboost, and implemented EMTboost $(r = 1/6)$. Implemented EMTboost performs the best in MAD and MAD^0, and better than EMTboost and TDboost in Gini^a, and Gini^{a+}.

Appendix C.2: Re-sampling: under-sampling fraction $\lambda = 0.2$ and over-sampling fraction $\eta = 3$

We control the nonzero sample size in real application by under-sampling the nonzero-loss data set with fraction $\lambda = 0.2$ and over-sampling the zero-loss data with fraction $\eta = 3$, generating a data set containing 95.9% zeros. Following the training and testing procedure in Sec. 5, Table C3 shows that EMTboost has the smallest of the maximal (averaged) Gini^b indices, thus is chosen as the "best" model.

References

Breiman, L. 1998. Arcing classifier (with discussion and a rejoinder by the author). *The Annals of Statistics* 26: 801–49.

Breiman, L. 1999. Prediction games and arcing algorithms. *Neural Computation* 11 (7):1493–517. doi:10.1162/089976699300016106.

Brent, R. P. 2013. *Algorithms for minimization without derivatives*. Englewood Cliffs, NJ: Prentice-Hall.

Cragg, J. G. 1971. Some statistical models for limited dependent variables with application to the demand for durable goods. *Econometrica* 39 (5):829–44. doi:10.2307/1909582.

Dunn, P. K., and G. K. Smyth. 2005. Series evaluation of Tweedie exponential dispersion model densities. *Statistics and Computing* 15 (4):267–80. doi:10.1007/s11222-005-4070-y.

Fontaine, S., Y. Yang, W. Qian, Y. Gu, and B. Fan. 2019. A unified approach to sparse Tweedie modeling of multisource insurance claim data. *Technometrics*. Accepted Manuscript.

Frees, E., G. Lee, and L. Yang. 2016. Multivariate frequency-severity regression models in insurance. *Risks* 4 (1):4. doi:10.3390/risks4010004.

Frees, E. W., G. Meyers, and A. D. Cummings. 2011. Summarizing insurance scores using a Gini index. *Journal of the American Statistical Association* 106 (495):1085–98. doi:10.1198/jasa.2011.tm10506.

Frees, E. W., G. Meyers, and A. D. Cummings. 2014. Insurance ratemaking and a Gini index. *Journal of Risk and Insurance* 81:335–66.

Freund, Y., and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55 (1):119–39. doi:10.1006/jcss.1997.1504.

Friedman, J., T. Hastie, and R. Tibshirani. 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. [Mismatch] doi:10.1214/aos/1016218223.

Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29 (5):1189–232. doi:10.1214/aos/1013203451.

Friedman, J. H. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38:367–78.

Hall, D. B. 2000. Zero-inflated Poisson and binomial regression with random effects: A case study. *Biometrics* 56 (4):1030–9. doi:10.1111/j.0006-341x.2000.01030.x.

Hastie, T. J., and R. J. Tibshirani. 1990. *Generalized additive models*. Vol. 43. Boca Raton, FL: CRC Press.

Hastie, T. J., R. J. Tibshirani, and J. Friedman. 2009. *The elements of statistical learning: Data mining, inference, and prediction*. Springer Series in Statistics. New York, NY: Springer.

Jørgensen, B. 1987. Exponential dispersion models. *Journal of the Royal Statistical Society: Series B (Methodological)* 49 (2):127–62. doi:10.1111/j.2517-6161.1987.tb01685.x.

Jørgensen, B., and M. C. Paes De Souza. 1994. Fitting Tweedie's compound Poisson model to insurance claims data. *Scandinavian Actuarial Journal* 1994 (1):69–93. doi:10.1080/03461238.1994.10413930.

Lambert, D. 1992. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34 (1):1–14. doi:10.2307/1269547.

Mildenhall, S. J. 1999. A systematic relationship between minimum bias and generalized linear models. In Proceedings of the Casualty Actuarial Society, vol. 86, 393–487.

Mullahy, J. 1986. Specification and testing of some modified count data models. *Journal of Econometrics* 33 (3): 341–65. doi:10.1016/0304-4076(86)90002-3.

Murphy, K. P., M. J. Brockman, and P. K. Lee. 2000. Using generalized linear models to build dynamic pricing systems. In *Casualty Actuarial Society Forum, Winter*, 107–39. Landover, MD: Colortone Press.

Nelder, J., and R. Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)* 135 (3):370–84. doi:10.2307/2344614.

Qian, W., Y. Yang, and H. Zou. 2016. Tweedie's compound Poisson model with grouped elastic net. *Journal of Computational and Graphical Statistics* 25 (2):606–25. doi:10.1080/10618600.2015.1005213.

Sandri, M., and P. Zuccolotto. 2008. A bias correction algorithm for the Gini variable importance measure in classification trees. *Journal of Computational and Graphical Statistics* 17 (3):611–28. doi:10.1198/106186008X344522.

Sigrist, F., and C. Hirnschall. 2019. Grabit: Gradient tree-boosted Tobit models for default prediction. *Journal of Banking & Finance* 102:177–92. doi:10.1016/j.jbankfin.2019.03.004.

Smyth, G. K., and B. Jørgensen. 2002. Fitting Tweedie's compound Poisson model to insurance claims data: Dispersion modelling. *ASTIN Bulletin* 32 (1):143–57. doi:10.2143/AST.32.1.1020.

Tobin, J. 1958. Estimation of relationships for limited dependent variables. *Econometrica* 26 (1):24–36. doi:10.2307/1907382.

Tweedie, M. 1984. An index which distinguishes between some important exponential families. In Statistics: Applications and New Directions: Proceedings of Indian Statistical Institute Golden Jubilee International Conference, 579–604.

Wood, S. N. 2006. *Generalized additive models: An introduction with R*. Boca Raton, FL: CRC Press.

Yang, Y., W. Qian, and H. Zou. 2018. Insurance premium prediction via gradient tree-boosted Tweedie compound Poisson models. *Journal of Business & Economic Statistics* 36 (3):456–15. doi:10.1080/07350015.2016.1200981.

Yip, K. C., and K. K. Yau. 2005. On modeling claim frequency data in general insurance with extra zeros. *Insurance: Mathematics and Economics* 36 (2):153–63. doi:10.1016/j.insmatheco.2004.11.002.

Zhang, T., and B. Yu. 2005. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics* 33 (4):1538–79. doi:10.1214/009053605000000255.

Zhang, Y. 2013. Likelihood-based and bayesian methods for tweedie compound poisson linear mixed models. *Statistics and Computing* 23 (6):743–57. doi:10.1007/s11222-012-9343-7.