Tweedie's Compound Poisson Model With Grouped Elastic Net

Wei Qian, Yi Yang and Hui Zou

Abstract. Tweedie's Compound Poisson model is a popular method to model data with probability mass at zero and non-negative, highly right-skewed distribution. Motivated by wide applications of the Tweedie model in various fields such as actuarial science, we investigate the grouped elastic net method for the Tweedie model in the context of the generalized linear model. To efficiently compute the estimation coefficients, we devise a two-layer algorithm that embeds the blockwise majorization descent method into an iteratively reweighted least square strategy. In together with the strong rule, the proposed algorithm is implemented in an easy-to-use R package `HDtweedie`, and is shown to compute the whole solution path very efficiently. Simulations are conducted to study the variable selection and model fitting performance of various lasso methods for the Tweedie model. The modeling applications in risk segmentation of insurance business are illustrated by analysis of an auto insurance claim dataset. Supplementary materials for this article are available online.

Key Words: coordinate descent, insurance score, IRLS-BMD, lasso, variable selection

1. INTRODUCTION

Tweedie's Compound Poisson model is known to model data with highly right-skewed distribution, which has probability mass at zero and non-negative support. As an example, the histogram of an auto insurance claim data in Figure 1 has a spike at zero and a heavy right tail at the positive range (see section 5 for a description of the data illustrated here). Specifically, the response Y of the Tweedie's Compound Poisson model can be represented as

$$Y = \sum_{i=1}^{N} X_i, \tag{1}$$

where N is a Poisson random variable with mean ξ, and conditional on N, X_i's $(1 \leq i \leq N)$ are i.i.d. Gamma(α, γ) distribution with mean $\alpha\gamma$ and variance $\alpha\gamma^2$. When $N = 0$, $Y = 0$. From now on, we call the distribution of Y the Tweedie distribution or the Tweedie model for simplicity. It is clear that the Tweedie distribution has positive probability mass at zero, since $P(Y = 0) = P(N = 0) = \exp(-\xi)$.

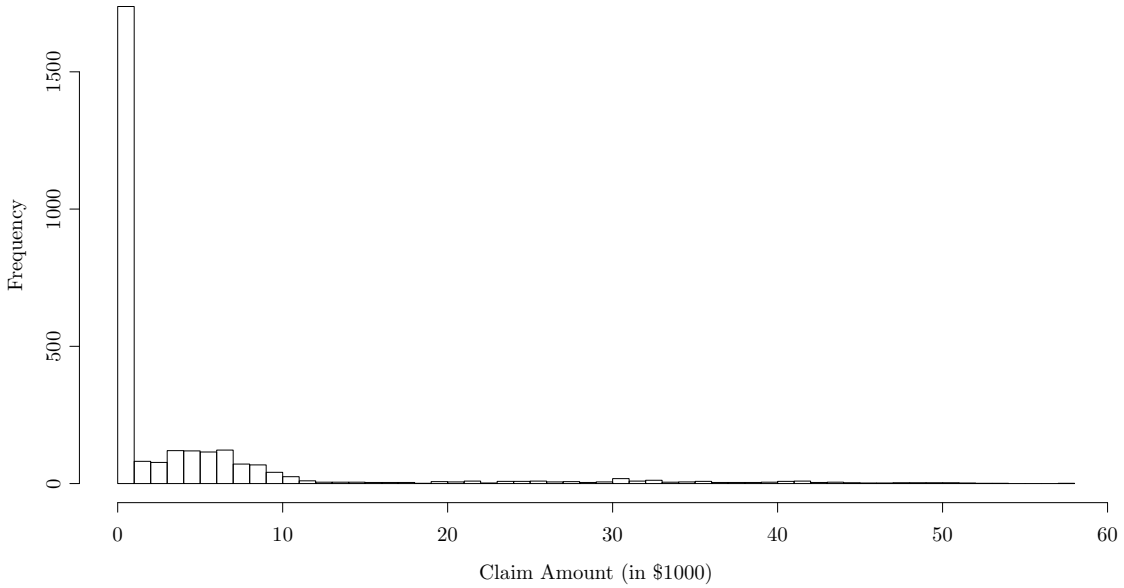

Figure 1: Histogram of an auto insurance claim data.

The Tweedie distribution has attracted applications from diverse fields. For example, in actuarial science, Y refers to the total claim loss of an insurance policy, N is the number of claims, and X_i $(1 \leq i \leq N)$ is the individual loss of the ith claim (e.g., Smyth and Jϕrgensen, 2002; Zhang, 2013*b*). In meteorological studies, Y can be the total weekly precipitation, N is the number of rainfall events and X_i is the precipitation of the ith event (e.g., Dunn, 2004). Also, data with patterns of the Tweedie distribution often arises in ecological studies and political science analysis. A typical example of ecological studies is fishery survey, in which Y is the total biomass of a particular fish species, N is the fish count, and X_i is the weight of the ith fish (e.g., Shono, 2008; Foster and Bravington, 2013). In political science, the dollar outcomes (Y) are often a result of an aggregation of a number of projects or grants (e.g., Lauderdale, 2012). For a broader account of Tweedie model applications and their references, see also Dunn and Smyth (2005).

The Tweedie model is known to be closely connected to the dispersion exponential model (Jϕrgensen, 1987), which has the form

$$f(y|\theta,\phi) = a(y,\phi)\exp(\frac{y\theta-\kappa(\theta)}{\phi}), \tag{2}$$

where $a(\cdot)$ and $\kappa(\cdot)$ are given functions, θ is a parameter in $\mathbb{R}$ and ϕ is the dispersion parameter in $(0, +\infty)$. By the well-known property of exponential family distributions, $\mu := E(Y) = \dot{\kappa}(\theta)$ and $Var(Y) = \phi\ddot{\kappa}(\theta)$, where $\dot{\kappa}(\theta)$ and $\ddot{\kappa}(\theta)$ are the first and second derivatives of $\kappa(\theta)$, respectively. If the mean-variance relation is specified to be $Var(Y) = \phi\mu^{\rho}$, where ρ is the power parameter $(1 < \rho < 2)$, we have $\ddot{\kappa}(\theta) = \mu^{\rho}$, $\theta = \mu^{1-\rho}/(1-\rho)$ and $\kappa(\theta) = \mu^{2-\rho}/(2-\rho)$. Then (2) can be written as

$$f(y|\mu, \rho, \phi) = a(y, \phi) \exp\Big(\frac{1}{\phi}\big(\frac{y\mu^{1-\rho}}{1-\rho} - \frac{\mu^{2-\rho}}{2-\rho}\big)\Big). \tag{3}$$

By comparing the moment generating functions (Smyth, 1996), it is easy to see that models (1) and (3) are equivalent when $\xi = \mu^{2-\rho}/\phi(2-\rho)$, $\alpha = (2-\rho)/(\rho-1)$ and $\gamma = \phi(\rho-1)\mu^{\rho-1}$. Note that $\rho = 1$ corresponds to the Poisson distribution and $\rho = 2$ corresponds to the Gamma distribution. We only consider the case that $1 < \rho < 2$, which is the primary interest of this article, although the derived algorithm in section 2 and section 3 can be applied to the cases of $\rho = 1$ and $\rho = 2$ with some minor modifications (see also section 5 for an application of the Gamma distribution modeling with the grouped elastic net).

One of the most important questions in Tweedie model applications is how to explain the response by predictor variables. For example, in actuarial studies, it is important to understand how the policy holder's characteristics are related to the expected claim loss. In precipitation modeling, the precipitation amount can be associated with the history weather record and other relevant climate measurements. The biomass in fishery studies can be determined by temporospatial factors and other fishery and environmental variables. The dollar outcomes in political science studies can be related to political and demographic variables. In the context of the generalized linear model, we assume that μ is associated with a p-dimensional predictor vector $\mathbf{x} \in \mathbb{R}^p$. Here we use the log link, that is, $\log(\mu) = \beta_0 + \boldsymbol{\beta}^T\mathbf{x}$, where β_0 is the intercept and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the coefficient vector. Such log-linear relation generates a multiplicative structure that is convenient for explanatory analysis and is widely adopted in the aforementioned applications. More arguments for the use of a multiplicative model and the log link in GLM for insurance applications can be found in, e.g., Ohlsson and Johansson (2010, section 1.3) and Murphy et al. (2000). Let $\{(y_i, \mathbf{x}_i), i = 1, 2, \cdots, n\}$ be the i.i.d. observations with sample size n. Further assume that ϕ is the same for all observations.

Then, the *negative* log-likelihood can be written as

$$l(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^{n} v_i \Big(\frac{y_i e^{-(\rho-1)(\beta_0+\boldsymbol{\beta}^T \mathbf{x_i})}}{\rho - 1} + \frac{e^{(2-\rho)(\beta_0+\boldsymbol{\beta}^T \mathbf{x_i})}}{2-\rho} \Big), \tag{4}$$

where v_i's $(1 \leq i \leq n)$ are the observation weights (by default, they are all equal to $1/n$). When the dimension of $\mathbf{x}$ is high, which is common in practice, a model selection technique has to be applied. For example, in insurance industry, it is common practice that hundreds and even thousands of variables are created for insurance policy pricing purposes. However, only a very small proportion of these variables are adopted in the final model.

Among various model selection techniques, the lasso (Tibshirani, 1996) is a very popular method that selects variables by shrinking some coefficient estimates to zero. Specifically, in the classical lasso setting, an L_1 penalty of coefficients is imposed to a negative log-likelihood function or other relevant loss functions. The minimizer of such penalized likelihood function is known to achieve both variable selection and coefficient estimation. The existence of efficient algorithms such as LARS (Efron et al., 2004; see also Osborne et al., 2000) and coordinate descent (Tseng, 2001; Friedman et al., 2010) makes the lasso an attractive competitor to other well-known model selection methods such as the stepwise or subset selection. Since the seminal work of Tibshirani (1996), a variety of lasso-type penalized methods are studied in order to achieve better results in different situations. For example, the adaptive lasso proposed by Zou (2006) imposes different weights on L_1 penalties of different variables, and achieves both variable selection consistency and estimation asymptotic normality. The elastic net proposed by Zou and Hastie (2005) applies L_2 penalties in addition to L_1 penalties, and better handles the situation that some variables are highly correlated with a "group"-like selection phenomenon. Another particularly interesting extension to the lasso is the grouped lasso (Yuan and Lin, 2006). By partitioning the variable coefficients into blocks and imposing a so-called grouped lasso penalty (which may be viewed as an intermediate between L_1 and L_2 penalties), for a given block, the grouped lasso solution either selects all the variables in the block or shrinks all coefficients of the block to zero. Such property of the grouped lasso is particularly important when categorical variables with multiple levels are present (e.g., in ANOVA model, we group dummy variables corresponding to one categorical factor into one block) or when some variables are treated as nonparametric components (e.g.,

in additive model, the nonparametric components is approximated by a linear combination of some basis functions, and we group the basis function terms that correspond to one nonparametric component into one block). The estimation and/or variable selection consistency properties of the grouped lasso estimators for linear models are studied in, e.g., Wang and Leng (2008), Bach (2008), Nardi et al. (2008), Huang and Zhang (2010), and Wei and Huang (2010).

In spite of the important progress in lasso methods and the broad applications of the Tweedie model, as far as we know, no publication is made regarding applications of the Tweedie model variable selection with lasso methods in any of the aforementioned scientific context. This somewhat surprising vacancy may be partially attributable to the lack of awareness in the relevant scientific community and the lack of publicly available software that is efficiently implemented to give the lasso-type solutions for the Tweedie model. The main purpose of this article is to introduce a unified algorithm that can efficiently solve various lasso-type problems for the Tweedie model and use data examples to illustrate its variable selection and model fitting performance. In particular, we choose the grouped lasso and the grouped elastic net as the main theme for the algorithm derivation since their special cases also give regular lasso and elastic net solutions. We also allow different weights for grouped lasso penalties so that the corresponding adaptive versions of the solutions can be generated.

Various algorithms has been studied for the grouped lasso usually under the linear regression and logistic regression settings. Yuan and Lin (2006) show that the solution path of the grouped lasso solution is generally not piecewise linear, which implies that the LARS-type algorithms do not apply for the grouped lasso. Motivated by the shooting algorithm of Fu (1998), they propose a blockwise coordinate descent algorithm for linear regression. However, their algorithm assumes a blockwise orthonormal condition, which is not always desirable in statistical applications. The study of the grouped lasso is extended to the logistic regression by Kim et al. (2006), who propose a gradient descent algorithm to solve the constrained-form problem. Meier et al. (2008) also propose a blockwise coordinate gradient descent (BCGD) algorithm for the grouped lasso of the logistic regression to directly solve the penalized-form problem.

For efficient computation of the grouped elastic net for the Tweedie model, we propose

a new blockwise coordinate descent algorithm that extends from the iteratively reweighted least square (IRLS) strategy (Friedman et al., 2010). A blockwise majorization descent (BMD) method is embedded into the IRLS strategy to solve the penalized weighted least square (WLS) problem. In addition, the strong rule (Tibshirani et al., 2012) is integrated to the algorithm to further speed up the computation of the whole solution path. The algorithm is implemented in an easy-to-use R package named `HDtweedie`, which is available in the supplementary materials.

As we mentioned before, one of the primary motivations resides in the promising applications in actuarial science. In particular, the regression functions obtained by Tweedie models with the grouped elastic net can serve as a candidate insurance score to achieve risk segmentation. Frees et al. (2011) propose an ordered version of the Lorenze curve to identify the discrepancy between the loss distribution and the baseline insurance premium distribution. The associated Gini index is used to gauge the performance of an insurance score for risk segmentation. Typically, a larger Gini index implies better risk segmentation, hence the better insurance score and underlying statistical model. In our numerical studies, we use the Gini index of the ordered Lorenze curve as a specific model comparison tool. A brief description of the ordered Lorenze curve and the Gini index in the context of insurance risk segmentation is deferred to the real data example in section 5, although the scope of their use may not be restricted to such context.

The rest of the article is organized as follows. Section 2 describes the algorithm for solving the Tweedie model with the grouped elastic net penalty. The computation of the solution path and the application of the strong rule are explained in section 3. The simulations and an insurance data example are presented in section 4 and section 5, respectively. A brief conclusion is given in section 6.

2. ALGORITHM

Assume the p-dimensional coefficient vector $\boldsymbol{\beta}$ is partitioned into g blocks, that is, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \cdots, \boldsymbol{\beta}_g^T)^T$, where $\boldsymbol{\beta}_j$ $(1 \leq j \leq g)$ is p_j-dimensional vector and $\sum_{j=1}^{g} p_j = p$. In the

following, we focus on the minimization problem with grouped elastic net penalties

$$(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) = \text{argmin}_{(\beta_0,\boldsymbol{\beta})}\, l(\beta_0, \boldsymbol{\beta}) + \lambda \sum_{j=1}^{g} \Big(\tau w_j \|\boldsymbol{\beta}_j\|_2 + \frac{1}{2}(1-\tau)\|\boldsymbol{\beta}_j\|_2^2 \Big), \tag{5}$$

where $\lambda > 0$ and $0 < \tau \leq 1$ are tuning parameters, and w_j's $(1 \leq j \leq g)$ are the positive weights for the grouped lasso penalties. Conforming to common practice, we do not penalize the intercept term β_0. Note that the grouped elastic net penalty used above is very general for solving lasso-type problems. Indeed, if $p_j = 1$ for all $1 \leq j \leq g$, the problem is reduced to the (adaptive) lasso when $\tau = 1$, and the (adaptive) elastic net when $0 < \tau < 1$. If $\tau = 1$ and $p_j > 1$ for some j, we have the (adaptive) grouped lasso problem. In the R package `HDtweedie`, the default choice of the observation weights v_i's $(1 \leq i \leq n)$ and the grouped lasso penalty weights w_j's $(1 \leq j \leq g)$ are $1/n$ and $\sqrt{p}_j$, respectively. The users can choose different values for v_i's and w_j's to meet their specific application needs.

The proposed algorithm essentially consists of two layers of loops. The outer layer is the IRLS strategy, which, at each iteration, approximates the objective function in (5) by a penalized WLS objective function. After obtaining the minimizer of the penalized WLS objective function, the next iteration begins by updating the working response and weight. Such outer-layer cycle continues until convergence. The inner layer is dedicated to obtaining the minimizer of the penalized WLS objective function by a BMD method. For simplicity, we call this two-layer strategy IRLS-BMD algorithm.

Specifically, for the outer-layer IRLS strategy, suppose $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ is the solution of $(\beta_0, \boldsymbol{\beta})$ from the most recent iteration. We approximate the negative log-likelihood $l(\beta_0, \boldsymbol{\beta})$ by the second-order Taylor expansion of $l(\beta_0, \boldsymbol{\beta})$ about $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$:

$$\begin{aligned} l_Q(\beta_0, \boldsymbol{\beta}) = {} & l(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) + \sum_{i=1}^{n} v_i \big(-y_i e^{-(\rho-1)(\tilde{\beta}_0 + \tilde{\boldsymbol{\beta}}^T \mathbf{x}_i)} + e^{(2-\rho)(\tilde{\beta}_0 + \tilde{\boldsymbol{\beta}}^T \mathbf{x}_i)} \big) \begin{pmatrix} 1 \\ \mathbf{x}_i \end{pmatrix}^T \begin{pmatrix} \beta_0 - \tilde{\beta}_0 \\ \boldsymbol{\beta} - \tilde{\boldsymbol{\beta}} \end{pmatrix} \\ & + \frac{1}{2} \sum_{i=1}^{n} v_i \big((\rho-1) y_i e^{-(\rho-1)(\tilde{\beta}_0 + \tilde{\boldsymbol{\beta}}^T \mathbf{x}_i)} + (2-\rho) e^{(2-\rho)(\tilde{\beta}_0 + \tilde{\boldsymbol{\beta}}^T \mathbf{x}_i)} \big) \begin{pmatrix} \beta_0 - \tilde{\beta}_0 \\ \boldsymbol{\beta} - \tilde{\boldsymbol{\beta}} \end{pmatrix}^T \begin{pmatrix} 1 \\ \mathbf{x}_i \end{pmatrix} \begin{pmatrix} 1 \\ \mathbf{x}_i \end{pmatrix}^T \begin{pmatrix} \beta_0 - \tilde{\beta}_0 \\ \boldsymbol{\beta} - \tilde{\boldsymbol{\beta}} \end{pmatrix} \\ = {} & -\sum_{i=1}^{n} \tilde{v}_i \tilde{y}_i \begin{pmatrix} 1 \\ \mathbf{x}_i \end{pmatrix}^T \begin{pmatrix} \beta_0 \\ \boldsymbol{\beta} \end{pmatrix} + \frac{1}{2} \sum_{i=1}^{n} \tilde{v}_i \begin{pmatrix} \beta_0 \\ \boldsymbol{\beta} \end{pmatrix}^T \begin{pmatrix} 1 \\ \mathbf{x}_i \end{pmatrix} \begin{pmatrix} 1 \\ \mathbf{x}_i \end{pmatrix}^T \begin{pmatrix} \beta_0 \\ \boldsymbol{\beta} \end{pmatrix} + C_1(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}), \end{aligned} \tag{6}$$

where

$$\tilde{v}_i = v_i\big((\rho-1)y_i e^{-(\rho-1)(\tilde{\beta}_0+\tilde{\boldsymbol{\beta}}^T\mathbf{x}_i)} + (2-\rho)e^{(2-\rho)(\tilde{\beta}_0+\tilde{\boldsymbol{\beta}}^T\mathbf{x}_i)}\big),$$
$$\tilde{y}_i = \tilde{\beta}_0 + \tilde{\boldsymbol{\beta}}^T\mathbf{x}_i + \frac{v_i}{\tilde{v}_i}\big(y_i e^{-(\rho-1)(\tilde{\beta}_0+\tilde{\boldsymbol{\beta}}^T\mathbf{x}_i)} - e^{(2-\rho)(\tilde{\beta}_0+\tilde{\boldsymbol{\beta}}^T\mathbf{x}_i)}\big), \tag{7}$$

and $C_1(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ is a constant given $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$. Therefore, we can rewrite (6) in the form of a WLS function $l_Q(\beta_0, \boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^n \tilde{v}_i(\tilde{y}_i - \beta_0 - \boldsymbol{\beta}^T\mathbf{x}_i)^2 + C(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$, where $C(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ is a constant given $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$. Here, we call $\tilde{v}_i$ and $\tilde{y}_i$ the working weight and the working response, respectively. Then, the penalized WLS objective function we intend to minimize is

$$P_Q(\beta_0, \boldsymbol{\beta}) := l_Q(\beta_0, \boldsymbol{\beta}) + \lambda\sum_{j=1}^g\Big(\tau w_j\|\boldsymbol{\beta}_j\|_2 + \frac{1}{2}(1-\tau)\|\boldsymbol{\beta}_j\|_2^2\Big). \tag{8}$$

The minimizer of (8) is used as the new $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ to update the working response and weight of $l_Q(\beta_0, \boldsymbol{\beta})$ to start a new IRLS iteration.

In order to find the minimizer of (8), we resort to the inner-layer loops, which employs a BMD method that sequentially updates the coefficients of each block by taking advantage of a majorization-minimization (MM) principle (see Wu et al. (2010) for a recent overview of the MM principle). In the following, we present the inner-layer BMD algorithm and its properties.

Given observation i $(1 \le i \le n)$, partitioning $\mathbf{x}_i$ the same way as $\boldsymbol{\beta}$, we have $\mathbf{x}_i = (\mathbf{x}_{i1}^T, \mathbf{x}_{i2}^T, \cdots, \mathbf{x}_{ig}^T)^T$, where $\mathbf{x}_{ij}$ $(1 \le j \le g)$ is the p_j-dimensional predictor vector corresponding to $\boldsymbol{\beta}_j$. For $1 \le j \le g$, denote the gradient vector and the Hessian matrix of $l_Q(\beta_0, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}_j$ by

$$\tilde{U}_j(\beta_0, \boldsymbol{\beta}) = \frac{\partial l_Q(\beta_0, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} = -\sum_{i=1}^n \tilde{v}_i(\tilde{y}_i - \beta_0 - \boldsymbol{\beta}^T\mathbf{x}_i)\mathbf{x}_{ij}, \tag{9}$$
$$\tilde{H}_j(\beta_0, \boldsymbol{\beta}) = \frac{\partial^2 l_Q(\beta_0, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j \partial \boldsymbol{\beta}_j^T} = \sum_{i=1}^n \tilde{v}_i\mathbf{x}_{ij}\mathbf{x}_{ij}^T =: \tilde{H}_j, \tag{10}$$

respectively. Let $\tilde{\gamma}_j$ $(1 \le j \le g)$ be the largest eigenvalue of $\tilde{H}_j$.

To update the block j coefficients $(1 \le j \le g)$, suppose $(\breve{\beta}_0, \breve{\boldsymbol{\beta}})$ is the most recently

updated estimate and define $\breve{U}_j = \tilde{U}_j(\breve{\beta}_0, \breve{\boldsymbol{\beta}})$. We update $\breve{\boldsymbol{\beta}}_j$ by solving

$$\text{argmin}_{\boldsymbol{\beta}_j}\, l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}) + \breve{U}_j^T(\boldsymbol{\beta}_j - \breve{\boldsymbol{\beta}}_j) + \frac{\tilde{\gamma}_j}{2}(\boldsymbol{\beta}_j - \breve{\boldsymbol{\beta}}_j)^T(\boldsymbol{\beta}_j - \breve{\boldsymbol{\beta}}_j) + \lambda\Big(\tau w_j \|\boldsymbol{\beta}_j\|_2 + \frac{1}{2}(1-\tau)\|\boldsymbol{\beta}_j\|_2^2\Big), \tag{11}$$

which has a closed form solution. Indeed, if we denote the solution of (11) by $\breve{\boldsymbol{\beta}}_j(\text{new})$, it is not hard to see by the Karush-Kuhn-Tucker (KKT) conditions that

$$\breve{\boldsymbol{\beta}}_j(\text{new}) = \frac{(\tilde{\gamma}_j \breve{\boldsymbol{\beta}}_j - \breve{U}_j)\big(1 - \frac{\lambda\tau w_j}{\|\tilde{\gamma}_j \breve{\boldsymbol{\beta}}_j - \breve{U}_j\|_2}\big)_+}{\tilde{\gamma}_j + \lambda(1-\tau)}, \tag{12}$$

where z_+ denotes the positive part of z. Similarly, to update the intercept term, define $\breve{U}_0 = -\sum_{i=1}^n \tilde{v}_i(\tilde{y}_i - \breve{\beta}_0 - \breve{\boldsymbol{\beta}}^T\mathbf{x}_i)$ and $\tilde{\gamma}_0 = \tilde{H}_0 = \sum_{i=1}^n \tilde{v}_i$. The intercept coefficient $\breve{\beta}_0$ is then updated by $\breve{\beta}_0(\text{new}) = \breve{\beta}_0 - \tilde{\gamma}_0^{-1}\breve{U}_0$. Such BMD updates cycle through all the blocks and the intercept sequentially until convergence, resulting in the minimizer of (8).

The update in (11) is justified by the fact that the updated value of the given penalized WLS objective function (8) always decreases. Indeed, given a block j $(1 \le j \le g)$, suppose $(\breve{\beta}_0, \breve{\boldsymbol{\beta}}(\text{new}))$ is the updated vector from $(\breve{\beta}_0, \breve{\boldsymbol{\beta}})$ by (11). Then,

$$\begin{aligned} l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}(\text{new})) &= \frac{1}{2}\sum_{i=1}^n \tilde{v}_i\Big(\tilde{y}_i - \breve{\beta}_0 - \breve{\boldsymbol{\beta}}^T\mathbf{x}_i - \big(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j\big)\mathbf{x}_{ij}\Big)^2 + C(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) \\ &= l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}) + \breve{U}_j^T\big(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j\big) + \frac{1}{2}\big(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j\big)^T\tilde{H}_j(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j) \\ &\le l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}) + \breve{U}_j^T\big(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j\big) + \frac{\tilde{\gamma}_j}{2}\big(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j\big)^T(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j), \end{aligned} \tag{13}$$

where the last inequality follows by the fact that $\tilde{H}_j$ is a positive definite matrix, and $\tilde{\gamma}_j$ is

its largest eigenvalue. Therefore,

$$
\begin{aligned}
&P_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}(\text{new})) - P_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}) \\
=& l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}(\text{new})) + \lambda\Big(\tau w_j \|\breve{\boldsymbol{\beta}}_j(\text{new})\|_2 + \frac{1}{2}(1-\tau)\|\breve{\boldsymbol{\beta}}_j(\text{new})\|_2^2\Big) \\
&\quad - l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}) - \lambda\Big(\tau w_j \|\breve{\boldsymbol{\beta}}_j\|_2 + \frac{1}{2}(1-\tau)\|\breve{\boldsymbol{\beta}}_j\|_2^2\Big) \\
\le& l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}) + \breve{U}_j^T\Big(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j\Big) + \frac{\tilde{\gamma}_j}{2}\Big(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j\Big)^T\Big(\breve{\boldsymbol{\beta}}_j(\text{new}) - \breve{\boldsymbol{\beta}}_j\Big) \\
&\quad + \lambda\Big(\tau w_j \|\breve{\boldsymbol{\beta}}_j(\text{new})\|_2 + \frac{1}{2}(1-\tau)\|\breve{\boldsymbol{\beta}}_j(\text{new})\|_2^2\Big) - l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}) - \lambda\Big(\tau w_j \|\breve{\boldsymbol{\beta}}_j\|_2 + \frac{1}{2}(1-\tau)\|\breve{\boldsymbol{\beta}}_j\|_2^2\Big) \\
\le& 0,
\end{aligned}
$$

where the first inequality follows by (13) and the second inequality follows by the update scheme (11). Similarly, the downhill-going property that $P_Q(\breve{\beta}_0(\text{new}), \breve{\boldsymbol{\beta}}) \le P_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}})$ also holds for the intercept update.

The IRLS-BMD algorithm described above is summarized in Algorithm 1.

3. SOLUTION PATH AND STRONG RULE

As a common practice for lasso-type methods, we want to solve the solution path of the grouped elastic net rather than only giving the solution for one λ value. Specifically, given τ, we consider a decreasing sequence of λ values $\{\lambda_k,\, k = 1, \cdots, m\}$. The grouped elastic net solution (5) of λ_k is denoted by $(\hat{\beta}_0^{(k)}, \hat{\boldsymbol{\beta}}^{(k)})$. The sequence $\{\lambda_k, k = 1, \cdots, m\}$ is created by choosing a grid of m points uniformly in log scale on $[\lambda_m, \lambda_1]$, where λ_m is a fixed small proportion of λ_1, and λ_1 is chosen to be the smallest value such that $\hat{\boldsymbol{\beta}} = \mathbf{0}$. The default in the R package `HDtweedie` is $m = 100$, $\lambda_m = 0.001\lambda_1$ if $p \le n$ and $\lambda_m = 0.05\lambda_1$ if $p > n$.

To compute the whole solution path, we start with the computation of λ_1 and $(\hat{\beta}_0^{(1)}, \hat{\boldsymbol{\beta}}^{(1)})$. By definition of λ_1, $\hat{\boldsymbol{\beta}}^{(1)} = \mathbf{0}$. The intercept estimate $\hat{\beta}_0^{(1)}$ can be easily obtained by the β_0 updating scheme in Algorithm 1. That is, we first initialize $\tilde{\beta}_0 = 0$ and $\tilde{\boldsymbol{\beta}} = \mathbf{0}$, and then repeatedly update $\tilde{\beta}_0$ until convergence with the following steps: (a) compute the working response and weight by (7); (b) compute $\tilde{U}_0 = -\sum_{i=1}^n \tilde{v}_i(\tilde{y}_i - \tilde{\beta}_0 - \tilde{\boldsymbol{\beta}}^T\mathbf{x}_i)$ and $\tilde{\gamma}_0 = \sum_{i=1}^n \tilde{v}_i$; (c) compute $\tilde{\beta}_0(\text{new}) = \tilde{\beta}_0 - \tilde{\gamma}_0^{-1}\tilde{U}_0$; (d) set $\tilde{\beta}_0 = \tilde{\beta}_0(\text{new})$. Subsequently, λ_1 is obtained by the KKT conditions that $\lambda_1 = \max_{1\le j\le g}\|U_j(\hat{\beta}_0^{(1)}, \hat{\boldsymbol{\beta}}^{(1)})\|_2/\tau w_j$, where $U_j(\beta_0, \boldsymbol{\beta}_j)$ is the

Algorithm 1 The IRLS-BMD algorithm for solving the Tweedie model grouped elastic net.

1. Initialize $\tilde{\beta}_0$ and $\tilde{\boldsymbol{\beta}}$.

2. (Outer layer) Update the penalized WLS objective function (8).

   - For $i = 1, 2, \cdots, n$, compute the working response $\tilde{y}_i$ and the working weight $\tilde{v}_i$ by (7).
   - For $j = 1, \cdots, g$, compute $\tilde{H}_j$ and its maximum eigenvalue $\tilde{\gamma}_j$ by (10); compute $\tilde{\gamma}_0 = \tilde{H}_0 = \sum_{i=1}^{n} \tilde{v}_i$.

3. (Inner layer) Apply the BMD algorithm to obtain the minimizer of the penalized WLS objective function (8).

   - Initialize $\breve{\beta}_0 = \tilde{\beta}_0$ and $\breve{\boldsymbol{\beta}} = \tilde{\boldsymbol{\beta}}$.
   - Repeat the following updating scheme until $(\breve{\beta}_0, \breve{\boldsymbol{\beta}})$ converges.
     - Update $\breve{\boldsymbol{\beta}}$. For $j = 1, 2, \cdots, g$, do
       - Compute $\breve{U}_j = \tilde{U}_j(\breve{\beta}_0, \breve{\boldsymbol{\beta}})$ by (9).
       - Compute $\breve{\boldsymbol{\beta}}_j$(new) by (12).
       - Set $\breve{\boldsymbol{\beta}}_j = \breve{\boldsymbol{\beta}}_j$(new).
     - Update β_0. Do
       - Compute $\breve{U}_0 = -\sum_{i=1}^{n} \tilde{v}_i(\tilde{y}_i - \breve{\beta}_0 - \breve{\boldsymbol{\beta}}^T \mathbf{x}_i)$.
       - Compute $\breve{\beta}_0$(new) $= \breve{\beta}_0 - \tilde{\gamma}_0^{-1}\breve{U}_0$.
       - Set $\breve{\beta}_0 = \breve{\beta}_0$(new).
   - Set $\tilde{\beta}_0 = \breve{\beta}_0$ and $\tilde{\boldsymbol{\beta}} = \breve{\boldsymbol{\beta}}$.

4. Repeat steps 2-3 until $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ converges.

gradient vector of $l(\beta_0, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}_j$ that

$$U_j(\beta_0, \boldsymbol{\beta}) = \frac{\partial l(\beta_0, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} = \sum_{i=1}^{n} (-y_i e^{-(\rho-1)(\beta_0+\boldsymbol{\beta}^T \mathbf{x}_i)} + e^{(2-\rho)(\beta_0+\boldsymbol{\beta}^T \mathbf{x}_i)}) v_i \mathbf{x}_{ij}.$$

With λ_1 and $(\hat{\beta}_0^{(1)}, \hat{\boldsymbol{\beta}}^{(1)})$ at hand, we can determine the decreasing sequence of λ_k's and compute the grouped elastic net solutions sequentially by Algorithm 1. At each λ_k ($2 \leq k \leq m$), the algorithm is "warm-started" by setting the initial coefficient estimate to be $(\hat{\beta}_0^{(k-1)}, \hat{\boldsymbol{\beta}}^{(k-1)})$, the solution of the preceding λ. In addition, we apply on top of Algorithm 1 the strong rule (Tibshirani et al., 2012), which is known to be a very effective technique to save computing time by guessing the likely zero-coefficient estimates at the beginning of the algorithm and discarding them from the updating scheme. Tibshirani et al. (2012) show that such practice is amazingly safe, and very rarely gives violations of the guess in linear regression and logistic regression. In the context of the grouped elastic net, the strong rule states that given $2 \leq k \leq m$ and $1 \leq j \leq g$, if

$$\|U_j(\hat{\beta}_0^{(k-1)}, \hat{\boldsymbol{\beta}}^{(k-1)})\|_2 < \tau w_k (2\lambda_k - \lambda_{k-1}), \tag{14}$$

then $\hat{\boldsymbol{\beta}}_j^{(k)}$ (the block j coefficient estimate at λ_k) is very likely to be zero.

At λ_k, let S be the set of j's ($1 \leq j \leq g$) such that (14) is *not* satisfied. Let $\mathbf{x}_{iS}$ ($1 \leq i \leq n$) be the subvector of $\mathbf{x}_i$ that contains only variables of blocks in S. Then the strong rule statement implies that Algorithm 1 can be applied to the reduced dataset $\{(y_i, \mathbf{x}_{iS}), i = 1, \cdots, n\}$ to estimate β_0 and the coefficients of x_{iS}, while estimated coefficients for blocks in S^c are set to be zero. We denote such obtained strong rule estimate of $(\beta_0, \boldsymbol{\beta})$ by $(\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})$. In order to check if the strong rule correctly identifies the zero estimates, we have to apply a KKT condition check, that is, if $(\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})$ is the correct solution, then for every $j \in S^c$, $\|U_j(\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})\|_2 \leq \lambda_k \tau w_j$. Define $V = \{j \in S^c : \|U_j(\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})\|_2 > \lambda_k \tau w_j\}$, the set of blocks in S^c that does not pass the KKT condition check. If $V = \varnothing$, the correct solution is obtained. Otherwise, add all the elements in V to S, and repeat Algorithm 1 on the new reduced dataset followed by the KKT condition check until we find the correct solution. It turns out that the strong rule works very well for the Tweedie model, and the computing time can often be significantly reduced (see numerical results in section 4).

Algorithm 2 The algorithm with the strong rule for solving the Tweedie model grouped elastic net at λ_k $(2 \leq k \leq m)$.

1. Identify the set of groups for the updating scheme by the strong rule:

$$S = \{1 \leq j \leq g : \|U_j(\hat{\beta}_0^{(k-1)}, \hat{\boldsymbol{\beta}}^{(k-1)})\|_2 \geq \tau w_k(2\lambda_k - \lambda_{k-1})\}.$$

2. Initialize $\tilde{\beta}_0 = \hat{\beta}_0^{(k-1)}$ and $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^{(k-1)}$.

3. Apply Algorithm 1 on the reduced dataset $\{(y_i, \mathbf{x}_{iS}), 1 \leq i \leq n\}$ to obtain the strong rule estimate $(\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})$.

4. Perform the KKT condition check and identify the set of blocks that fails the check:

$$V = \{j \in S^c : \|U_j(\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})\|_2 > \lambda_k \tau w_j\}.$$

5. If $V = \varnothing$, return $(\hat{\beta}_0^{(k)}, \hat{\boldsymbol{\beta}}^{(k)}) = (\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})$. Otherwise, set $S = S \cup V$, initialize $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) = (\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})$, and repeat steps 3-5.

The algorithm with the strong rule for the grouped elastic net solution at λ_k $(2 \leq k \leq m)$ is summarized in Algorithm 2.

4. SIMULATION

In the simulation study, we intend to investigate the performance of the Tweedie model with lasso, grouped lasso and grouped elastic net methods using the following two examples.

4.1. Example 1

In each run of the simulation, we sample 500 observations for both the training and testing datasets. The design matrix is created as follows. First, sample 8-dimensional covariates $\mathbf{T} = (T_1, \cdots, T_8)$ from a certain distribution scenario. Then, similar to simulation settings of Kim et al. (2006), each covariate T_j $(j = 1, \cdots, 8)$ generates three polynomial terms $p_1(T_j)$, $p_2(T_j)$ and $p_3(T_j)$, where $p_1(x) = x$, $p_2(x) = (3x^2 - 1)/6$ and $p_3(x) = (5x^3 - 3x)/10$. The design matrix is formed by the resulting 24 terms. Naturally, we assign $p_1(T_j)$, $p_2(T_j)$ and $p_3(T_j)$ to the same block $(j = 1, \cdots, 8)$. The response is generated by the Tweedie model with $\rho = 1.5$ and $\phi = 1$ (with different scenarios considered). Then, the Tweedie models with

lasso, grouped lasso and grouped elastic net methods are fitted using the training dataset. The tuning parameter λ is selected by the five-fold cross validations with deviance. For the grouped elastic net, the additional tuning parameter τ is selected from $\{0.1, 0.2, \cdots, 1.0\}$.

To compare the variable selection performance, we consider the blocks of the covariates and say that the block of a covariate is identified as active if at least one of the estimated coefficients in this block is nonzero. Similarly, we consider the individual predictor terms and say that a predictor term is identified as active if its estimated coefficient is nonzero. With the fitted models, we count the number of correctly identified active blocks (block-C) and the number of incorrectly identified active blocks (block-IC). In addition, we consider the coefficients of the individual terms, and count the number of correctly identified active coefficients (coefficient-C) and the number of incorrectly identified active coefficients (coefficient-IC). Also, we use the testing dataset to calculate the negative log-likelihood score and the Gini index (see section 5 for a brief description of the Gini index). The experiment is repeated 100 times to obtain the averaged values for the aforementioned model fitting measurements. In the following, we consider three different cases for the distribution of covariates $\mathbf{T}$ and the link function.

4.1.1 CASE 1

We assume that $\mathbf{T}$ is multivariate normal with the mean being $\mathbf{0}$ and the variance matrix being a compound symmetry correlation matrix Σ_1. Let $(\Sigma_1)_{ij} = \omega$ ($i \neq j$, and $i, j = 1, \cdots, 8$), where $\omega = 0$ or 0.5. The link function is

$$\log \mu = 0.3 + \sum_{j=1}^{3} (-1)^{(j+1)} \big(0.5 p_1(T_j) + 0.2 p_2(T_j) + 0.5 p_3(T_j)\big).$$

Clearly, the true model has 3 relevant blocks and 9 relevant predictor terms. The simulation results are summarized in Table 1 (values in the parenthesis are standard errors).

Since the link function is specified to have an explicit blockwise structure, there is no surprise that the grouped lasso and the grouped elastic net have better variable selection results than the lasso by identifying more relevant blocks and less irrelevant blocks. The grouped lasso and the grouped elastic net also show some advantages over the lasso when comparing the negative log-likelihood and the Gini index. As expected, coefficient-C and

coefficient-IC show that for the grouped lasso and the grouped elastic net, all estimated coefficients of an active block are nonzero, while for the lasso, some estimated coefficients of an active block may be zero. In addition, the grouped elastic net appears to choose more blocks than the grouped lasso. Such feature of the grouped elastic net can be appealing in some situations, as shown in the next case.

4.1.2 CASE 2

This case is inspired by the simulation results of the elastic net in Zou and Hastie (2005). Let $\mathbf{Z} = (Z_1, \cdots, Z_6)$ be a multivariate normal random variable with the mean being $\mathbf{0}$ and the variance being a compound symmetry correlation matrix Σ_2. Let $(\Sigma_2)_{ij} = \omega$ ($i \neq j$, and $i, j = 1, \cdots, 6$), where $\omega = 0$ or 0.5. Then $\mathbf{T}$ is generated by $T_1 = Z_1 + \varepsilon_1$, $T_2 = Z_1 + \varepsilon_2$, $T_3 = Z_1 + \varepsilon_3$, and $T_j = Z_{j-2}$ ($j = 4, \cdots, 8$), where ε_1, ε_2, ε_3 are Normal(0, 0.01). The link function remains the same as that of Case 1. Under this setting, T_1, T_2 and T_3 are highly correlated, and their blocks are all active in the true model.

Based on the results summarized in Table 1, it is interesting to see that the grouped elastic net correctly identifies almost all three relevant blocks (averaged block-C: 2.77 and 2.89), while the grouped lasso on average misses more than one relevant blocks (averaged block-C: 1.88 and 1.83). Such phenomenon shows the ability of the grouped elastic net to better handle the correlated covariates (and the blocks generated from them), which is reminiscent of the unique property of the elastic net (Zou and Hastie, 2005). From a practical viewpoint, the grouped elastic net reveals more relevant (and possibly highly correlated) covariates to an analyst so that a larger pool of variables is available for further investigation. Also, similar to Case 1, the grouped lasso and the grouped elastic net perform better than the lasso in terms of the negative log-likelihood and the Gini index, as is expected from the structure of the link function.

4.1.3 CASE 3

In this case, we intentionally use the link function that favors the lasso:

$$\log \mu = 0.3 + \sum_{j=1}^{6} (-1)^{j+1} p_1(T_j).$$

Table 1: (Example 1) Averaged simulation results based on 100 runs.

	block-C	block-IC	coefficient-C	coefficient-IC	negative log-likelihood	Gini index
Case 1						
oracle	3	0	9	0	-	-
$\omega = 0$						
lasso	2.95	0.80	5.65	0.87	45 (22)	0.961 (0.011)
grouped lasso	3.00	0.26	9.00	0.78	33 (14)	0.972 (0.003)
grouped elastic net	3.00	0.69	9.00	2.07	35 (15)	0.975 (0.002)
$\omega = 0.5$						
lasso	2.86	1.21	5.25	1.25	46 (22)	0.961 (0.012)
grouped lasso	2.87	0.60	8.61	1.80	37 (17)	0.961 (0.012)
grouped elastic net	2.94	1.08	8.82	3.24	35 (15)	0.963 (0.012)
Case 2						
oracle	3	0	9	0	-	-
$\omega = 0$						
lasso	1.94	0.18	2.78	0.18	11.5 (2.0)	0.896 (0.011)
grouped lasso	1.88	0.04	5.64	0.12	9.1 (0.7)	0.898 (0.010)
grouped elastic net	2.77	0.11	8.31	0.33	8.8 (0.6)	0.899 (0.010)
$\omega = 0.5$						
lasso	2.01	0.49	2.84	0.49	23 (12)	0.914 (0.009)
grouped lasso	1.83	0.07	5.49	0.21	14 (4)	0.916 (0.009)
grouped elastic net	2.89	0.39	8.67	1.17	13 (3)	0.916 (0.009)
Case 3						
oracle	6	0	6	0	-	-
$\omega = 0$						
lasso	6.00	0.51	6.00	4.42	5.63 (0.02)	0.604 (0.003)
grouped lasso	6.00	0.85	6.00	14.55	5.64 (0.02)	0.599 (0.003)
grouped elastic net	6.00	1.23	6.00	15.69	5.64 (0.02)	0.599 (0.003)
$\omega = 0.5$						
lasso	6.00	0.63	6.00	4.26	5.137 (0.017)	0.447 (0.002)
grouped lasso	6.00	0.94	6.00	14.82	5.151 (0.017)	0.441 (0.002)
grouped elastic net	6.00	1.22	6.00	15.66	5.146 (0.017)	0.442 (0.002)

The distribution of $\mathbf{T}$ is the same as Case 1. Under this senario, the true model has 6 relevant blocks and 6 relevant predictor terms. As summarized in Table 1, while all three methods correctly recover all 6 relevant blocks, the Gini index show that the lasso is favored over the grouped lasso.

Recall that the strong rule is integrated into the proposed algorithm. Next, we use settings of Case 3 to evaluate the computing time reduction due to the strong rule. For evaluation of higher-dimensional situations, we add q more irrelevant variables into the original design matrix ($q = 0, 10, 100, 500, 1000$). The distribution of these irrelevant variables are i.i.d. Normal(0,1). Using the covariate $\mathbf{T}$ with $\omega = 0.5$, the enlarged designed matrix and the same link function, we fit Tweedie models with the lasso, and record the total computing time based on 5 runs of the experiment. For comparison, we remove the strong rule from the algorithm and repeat the experiment under exactly the same setting. From the results summarized in Table 2, we can see that the algorithm with the strong rule consistently has shorter computing time than its counterpart without the strong rule. The effects of the strong rule in terms of time saving become even more apparent as the predictor dimension grows larger. The corresponding variable selection results are given in Table 3, which satisfactorily show that the number of incorrectly identified active variables grows only moderately as q increases.

Table 2: Comparing the computing time (in seconds) with and without the strong rule.

strong rule	$q = 0$	$q = 10$	$q = 100$	$q = 500$	$q = 1000$
no	2.5	3.6	12.6	26.7	56.7
yes	2.1	3.1	11.1	11.7	26.7

Table 3: Variable selection results obtained in the computing time study.

	$q = 0$	$q = 10$	$q = 100$	$q = 500$	$q = 1000$
block-C	6.0	6.0	6.0	6.0	6.0
block-IC	1.0	1.4	3.2	5.6	9.4

4.2. Example 2

In this example, we provide numerical comparisons between our method and some existing model selection methods. We consider a 20-dimensional covariate example. Assume the covariate $\mathbf{X} = (X_1, X_2, \cdots, X_{20})$ is multivariate normal with the mean being $\mathbf{0}$ and the variance matrix being an exponential decay correlation matrix Σ. Let $(\Sigma)_{i,j} = \omega^{|i-j|}$ $(i, j = 1, \cdots, 20)$, where $\omega = 0$ or 0.5. Different from the idealized settings of Example 1 where the (nonparametric) component of each covariate in the link function can be expanded by up to third-degree polynomials, we consider in Example 2 the following link functions for data generation: **(Case 1)** $\log \mu = \sum_{j=1}^{12} g_1(X_j)$; **(Case 2)** $\log \mu = \sum_{j=1}^{8} g_2(X_j) + \sum_{j=9}^{12} g_1(X_j)$, where $g_1(x) = 10^4 x^3(1-x)^6[\frac{40}{3}x^8 + \frac{2}{3}(1-x)^4]I(0 \le x \le 1)$ and $g_2(x) = \{2\sin(4\pi x) - 6[|x - 0.4|^{0.3} - 1.1] - 0.5\text{sgn}(0.7-x)\}I(0 \le x \le 1)$. The plots for $g_1(\cdot)$ and $g_2(\cdot)$ are given in Figure 2. With the link functions above, the response is generated by the Tweedie model with $\rho = 1.5$ and $\phi = 1$. We sample 100 observations for the training dataset and 300 observations for the testing dataset.

To fit the training dataset with the proposed methods, we create the design matrix by expanding each covariate to cubic B-splines with eight degrees of freedom. The resulting design matrix has 160 terms, and we can naturally assign the eight terms of each covariate into one block. Then, we fit the data using the Tweedie models with lasso, grouped lasso and grouped elastic net (the tuning parameters are selected the same way as described in Example 1). For comparison, we also implement the backward-forward stepwise selection method for the Tweedie model, which is commonly used in actuarial studies for variable selection purposes (R code for the stepwise selection is available in the supplementary materials; the default p-values for entering and removal of a covariate are set to be 0.05 and 0.10, respectively).

The procedures described above is repeated 100 times, and the results are summarized in Tables 4. We can see in this example that under almost all considered scenarios, the estimation performance of the grouped elastic net is significantly better than that of the lasso, the grouped lasso and the stepwise method in terms of both negative log-likelihood and Gini index. As expected, the stepwise method performs poorly in both variable selection and estimation due to its inability to allow flexible nonlinear structure.

Table 4: (Example 2) Averaged simulation results based on 100 runs.

	block-		negative	Gini index
	C	IC	log-likelihood	
Case 1				
oracle	12	0	-	-
$\omega = 0$				
lasso	2.00	0.28	2135 (9)	0.092 (0.011)
grouped lasso	1.88	0.28	2135 (9)	0.087 (0.011)
grouped elastic net	5.78	1.78	2114 (9)	0.209 (0.011)
stepwise	1.03	0.41	2231 (16)	0.019 (0.006)
$\omega = 0.5$				
lasso	1.73	0.22	2160 (10)	0.104 (0.013)
grouped lasso	1.92	0.19	2154 (10)	0.124 (0.013)
grouped elastic net	6.95	1.83	2108 (11)	0.290 (0.010)
stepwise	1.27	0.25	2236 (14)	0.071 (0.007)
Case 2				
oracle	12	0	-	-
$\omega = 0$				
lasso	2.94	0.29	2305 (10)	0.125 (0.011)
grouped lasso	3.36	0.41	2301 (10)	0.144 (0.013)
grouped elastic net	7.76	2.21	2270 (10)	0.267 (0.009)
stepwise	1.00	0.33	2380 (13)	0.029 (0.005)
$\omega = 0.5$				
lasso	4.24	0.42	2302 (10)	0.210 (0.014)
grouped lasso	3.77	0.28	2306 (10)	0.199 (0.014)
grouped elastic net	9.23	2.43	2251 (10)	0.354 (0.006)
stepwise	1.81	0.31	2398 (11)	0.128 (0.005)

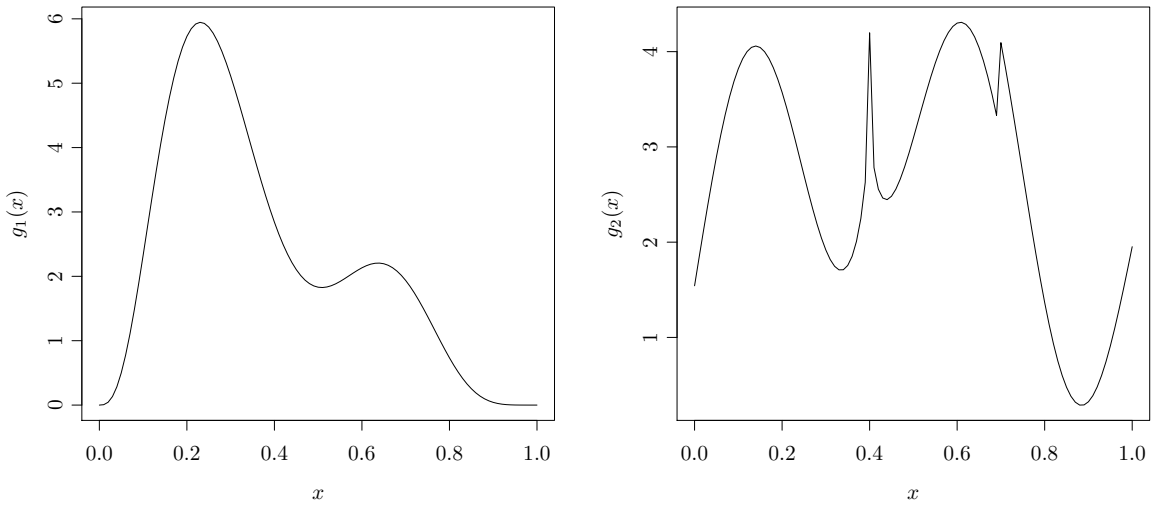

Figure 2: Plots for functions g_1 and g_2.

5. REAL DATA EXAMPLE

In this section, we use an auto insurance claim dataset studied in Yip and Yau (2005) and Zhang (2013*a*) to illustrate applications and performance of the Tweedie model with the grouped elastic net. The response (y) we want to predict is the aggregate claim loss of an auto insurance policy. Similar to the data treatment performed by Yip and Yau (2005) and Zhang (2013*a*), we only consider the new customers in the dataset and transform the response by $y^* = y/1000$. Then, the reduced dataset has 2812 insurance policy records, among which 60.7% of the policies has no claims (i.e., $y^* = 0$). The histogram of y^* is shown in Figure 1. The data also contains 21 predictor variables associated with the vehicle and the policy holder: number of children passengers (x_1), time to travel from home to work (x_2), whether the car is for commercial use (x_3), car value (x_4), number of policies (x_5), car type (x_6, 6 categories), whether the car color is red (x_7), whether the driver's license was revoked in the past (x_8), motor vehicle record (MVR) point (x_9), age (x_{10}), number of children at home (x_{11}), years on job (x_{12}), income (x_{13}), gender (x_{14}), whether married (x_{15}), whether a single parent (x_{16}), job class (x_{17}, 8 categories), education level (x_{18}, 5 categories), home value (x_{19}), years in current address (x_{20}) and whether the driver lives in urban area (x_{21}).

We transform $x_4^* = \log x_4$, $x_{13}^* = \log(x_{13} + 10)$, and scale all the numerical variables (ex-

cept for x_1, x_5, x_9 and x_{11}) to have mean 0 and standard deviation 1. The polynomial terms (up to the third order) of the 11 numerical variable are created the same way as we do in the simulations. These polynomial terms of each variable are treated as one coefficient block for the grouped lasso and the grouped elastic net. For the categorical variables with more than two levels (x_6, x_{17}, x_{18}), we treat the first level (by alphabetical order) as the base level, and create dummy variables accordingly for the other levels. Naturally, the dummy variables belonging to the same categorial variable are treated as one block. In addition, binary variables (0-1) are created for categorical variables with two levels ($x_3, x_7, x_8, x_{14}, x_{15}, x_{16}, x_{21}$).

The entire dataset is then randomly partitioned into a training set and a testing set with equal sample size. The training set is used to fit the Tweedie models with the grouped lasso (GrpLasso) and the grouped elastic net (GrpNet). Besides Tweedie models, another popular approach in analyzing insurance loss data is to model the frequency (whether a claim occurs) and severity (the amount of claim loss if a claim occurs) separately. Specifically, we first fit a logistic regression with the grouped lasso to model whether a claim occurs (frequency submodel); then we consider only the records with positive claim loss and fit a Gamma model with the grouped lasso for the claim loss (severity submodel). We refer to this two-component model as LogGam for short. The R package `gglasso` (Yang and Zou, 2012) is used to fit the logistic regression with the grouped lasso. For the Gamma model with the grouped lasso, since it is a special case of the Tweedie model with $\rho = 2$, our implementation described in the previous sections still applies. The aforementioned models select their tuning parameters by cross-validations as we do for the simulations. For comparison, we fit a regular Tweedie model with only the main effect variables considered in Yip and Yau (2005): x_3, x_{13}, x_{14}, x_{15} and x_{21} (for short, we call it YY model). Similarly, another regular Tweedie model is fitted with only the main effect variables considered in Zhang (2013*a*): x_3, x_9, x_{13}, x_{15} and x_{21} (for short, we call it WZ model). For simplicity, we set $\rho = 1.7$ for all the Tweedie models in this section (the value of ρ appears to have little influence on the results, the details of which are thus omitted). After fitting the models with the training set, we count the number of active variables (NAV for short), and use the testing set to calculate the Gini index of the ordered Lorenz curve (Frees et al., 2011). The procedures above are repeated 10 times, and the averaged values of NAV and the Gini index are summarized in Table 5 (the numbers in parenthesis are standard errors). As we have seen in the simulations, the grouped elastic net

tends to select more variables than the grouped lasso. For the LogGam model, we have to settle with two possibly different sets of variables, one for the frequency submodel and one for the severity submodel. Although each of the submodels may involve less active variables than the grouped lasso and the grouped elastic net, such two submodel structure may not be desirable for a straightforward interpretation of the loss-variable association.

Table 5: The averaged number of active variables and the averaged Gini index in the auto insurance claim data example.

	GrpLasso	GrpNet	LogGam		YY	WZ
			Frequency	Severity		
NAV	4.9	6.2	3.6	3.6	-	-
Gini index	0.462 (0.007)	0.463 (0.008)	0.460 (0.008)		0.156 (0.009)	0.360 (0.007)

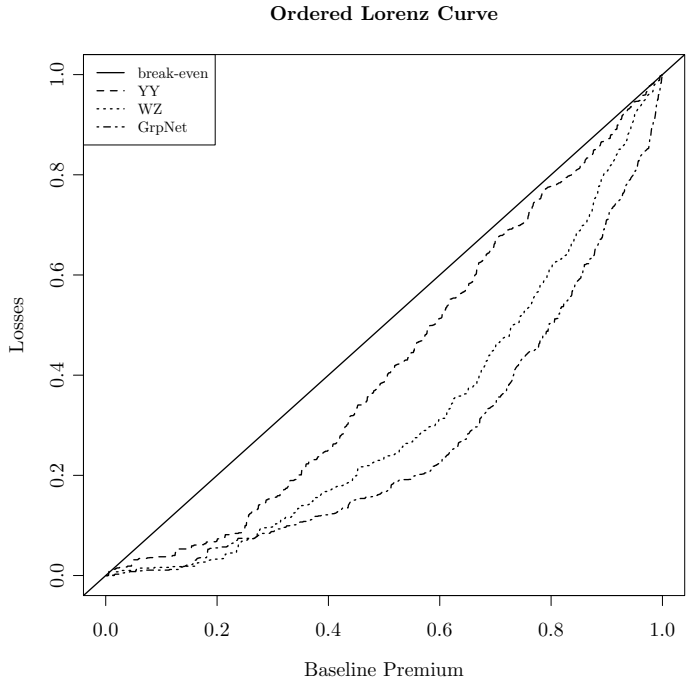

Figure 3: The ordered Lorenz curves for the auto insurance claim data. Larger area between the ordered Lorenz curve and the break-even line means better risk segmentation.

Next, we briefly explain the ordered Lorenz curve and the associated Gini index proposed

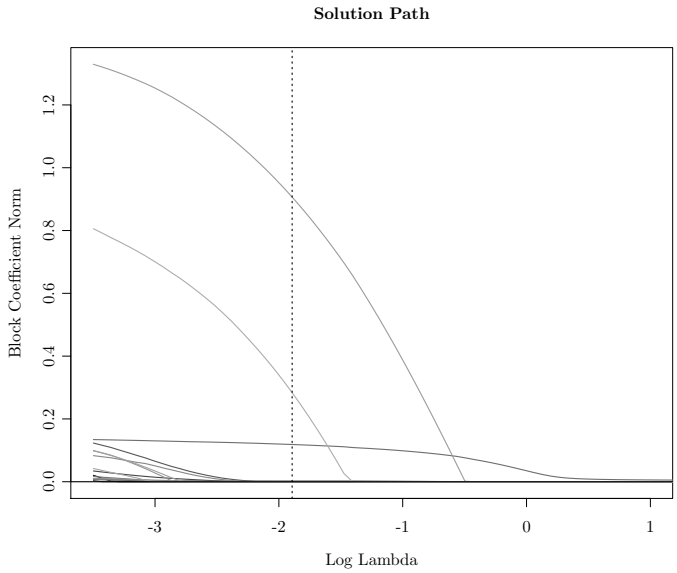

Figure 4: The solution path of the GrpNet model for the auto insurance claim data (the dotted line represents the λ selected by cross-validations; τ=0.7).

by Frees et al. (2011), which is especially useful for insurance risk segmentation purposes. In insurance business, when an insurer writes a policy for a customer, the insurer is exchanging a future loss (or risk) Y for an immediate premium income (or policy price) $M(\mathbf{X})$, where $\mathbf{X}$ is the covariate related to the policy holder's characteristics, and $M(\cdot)$ is a *known* baseline premium calculation function currently in use and possibly dependent on $\mathbf{X}$. Since the baseline premium $M(\mathbf{X})$ is not given in the auto insurance dataset, for simplicity, in this section, we choose the constant premium function $M(\cdot) \equiv 1$ for Gini index calculation. The constant $M(\cdot)$ is also used for the simulations of section 4. Given an alternative statistical model and a new covariate $\mathbf{X}$, we can choose the predicted value $\hat{\mu}(\mathbf{X})$ of the response to be an insurance score $S(\mathbf{X})$. For example, under the settings of section 2, the Tweedie model with the grouped elastic net chooses $S(\mathbf{X}) = \exp(\hat{\beta}_0 + \hat{\boldsymbol{\beta}}^T \mathbf{X})$. It is certainly desirable that the insurance score $S(\mathbf{X})$ can properly estimate the expected risk $E(Y|\mathbf{X})$ and differentiate the profitable business from the unprofitable business. Such insurance score, if shown effective in risk segmentation, can be used to improve the existing premium calculation mechanism.

To gauge whether an insurance score (and the underlying statistical model that generates the score) is effective, define a relativity $R(\mathbf{X}) = S(\mathbf{X})/M(\mathbf{X})$. Given a random sample (namely, the testing dataset in our example) $\{(\tilde{\mathbf{x}}_i, \tilde{y}_i), i = 1, \cdots, n\}$ of $(\mathbf{X}, Y)$, define two

empirical distribution functions

$$\hat{F}_L(r) = \frac{\sum_{i=1}^{n}[\tilde{y}_i I(R(\tilde{\mathbf{x}}_i) \le r)]}{\sum_{i=1}^{n}\tilde{y}_i} \quad \text{and } \hat{F}_M(r) = \frac{\sum_{i=1}^{n}[M(\tilde{\mathbf{x}}_i) I(R(\tilde{\mathbf{x}}_i) \le r)]}{\sum_{i=1}^{n}[M(\tilde{\mathbf{x}}_i)]}.$$

The ordered Lorenz curve is the graph of $(\hat{F}_M(r), \hat{F}_L(r))$. The ordered Lorenz curves of GrpNet, YY and WZ models in one run of our real data example are given in Figure 3, where the diagonal line is the so-called *break-even* line. Since the diagonal line means that the proportion of premium selected by the cut-off relativity value r is always equal to the proportion of the selected loss, it observes no risk segmentation, which gives the break-even situation. The Gini index is defined as twice the area between the ordered Lorenz curve and the break-even line.

The Gini index summarized in Table 5 implies that the GrpNet and the GrpLasso models perform slightly better than the LogGam model. The YY and the WZ models clearly underperform the others due to the lack of some relevant variables. We fit the GrpNet model with the entire dataset, and the variables x_5, x_8, x_9, x_{21} are selected as the important variables. The solution path of this model is shown in Figure 4. The partial fits of the selected variables are left in the Appendix (Figure A2).

6. CONCLUSION

Inspired by the success of the IRLS strategy in the coordinate descent algorithms (Friedman et al., 2010), we embed the BMD method into the IRLS loops to form the IRLS-BMD algorithm. The IRLS-BMD algorithm requires no blockwise orthonormal condition for the design matrix, and is shown to be computationally very efficient. In addition, by integrating the strong rule to the algorithm, we achieve faster computation of the solution path. With the wide applications of Tweedie models, our efficiently implemented R package `HDtweedie` can be an appealing tool for relevant scientific communities to investigate their own problems of interest when a large number of variables is involved.

SUPPLEMENTARY MATERIALS

Appendix: This supplementary file contains additional numerical examples and results not shown in the main article. (appendix.pdf)

R package HDtweedie: This supplementary file is the R package HDtweedie, which implements the algorithms proposed in this article. (HDtweedie_1.1.tar.gz)

R function step.tweedie: This supplementary file contains the R code to perform the stepwise selection for the Tweedie model. (Tweedie_stepwise.R)

R code for data analysis: This supplementary file contains the auto insurance claim data example illustrated in this article. (data_analysis.R)

ACKNOWLEDGEMENTS

The authors thank the Editor, the Associate Editor and an anonymous referee for their valuable comments that help improving this manuscript significantly.

REFERENCES

Bach, F. R. (2008), 'Consistency of the group lasso and multiple kernel learning', *Journal of Machine Learning Research* **9**, 1179–1225.

Dunn, P. K. (2004), 'Occurrence and quantity of precipitation can be modelled simultaneously', *International Journal of Climatology* **24**(10), 1231–1239.

Dunn, P. K. and Smyth, G. K. (2005), 'Series evaluation of Tweedie exponential dispersion model densities', *Statistics and Computing* **15**(4), 267–280.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. et al. (2004), 'Least angle regression', *The Annals of Statistics* **32**(2), 407–499.

Foster, S. D. and Bravington, M. V. (2013), 'A Poisson–Gamma model for analysis of ecological non-negative continuous data', *Environmental and Ecological Statistics* **20**(4), 533–552.

Frees, E. W., Meyers, G. and Cummings, A. D. (2011), 'Summarizing insurance scores using a gini index', *Journal of the American Statistical Association* **106**(495).

Friedman, J., Hastie, T. and Tibshirani, R. (2010), 'Regularization paths for generalized linear models via coordinate descent', *Journal of Statistical Software* **33**, 1–22.

Fu, W. J. (1998), 'Penalized regressions: the bridge versus the lasso', *Journal of Computational and Graphical Statistics* **7**(3), 397–416.

Huang, J. and Zhang, T. (2010), 'The benefit of group sparsity', *The Annals of Statistics* **38**(4), 1978–2004.

Jϕrgensen, B. (1987), 'Exponential dispersion models (with discussion)', *Journal of the Royal Statistical Society, Series B* **49**, 127–162.

Kim, Y., Kim, J. and Kim, Y. (2006), 'Blockwise sparse regression', *Statistica Sinica* **16**(2), 375.

Lauderdale, B. E. (2012), 'Compound Poisson–Gamma regression models for dollar outcomes that are sometimes zero', *Political Analysis* **20**, 387–399.

Meier, L., Van De Geer, S. and Bühlmann, P. (2008), 'The group lasso for logistic regression', *Journal of the Royal Statistical Society, Series B* **70**(1), 53–71.

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

Nardi, Y., Rinaldo, A. et al. (2008), 'On the asymptotic properties of the group lasso estimator for linear models', *Electronic Journal of Statistics* **2**, 605–633.

Ohlsson, E. and Johansson, B. (2010), *Non-Life Insurance Pricing with Generalized Linear Models*, Springer.

Osborne, M. R., Presnell, B. and Turlach, B. A. (2000), 'A new approach to variable selection in least squares problems', *IMA Journal of Numerical Analysis* **20**(3), 389–403.

Shono, H. (2008), 'Application of the Tweedie distribution to zero-catch data in CPUE analysis', *Fisheries Research* **93**(1), 154–162.

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, Gold Coast, Australia', pp. 17–19.

Smyth, G. K. and Jϕrgensen, B. (2002), 'Fitting Tweedie's compound Poisson model to insurance claims data: dispersion modelling', *ASTIN Bulletin* **32**(1), 143–157.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society, Series B* **58**, 267–288.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J. and Tibshirani, R. J. (2012), 'Strong rules for discarding predictors in lasso-type problems', *Journal of the Royal Statistical Society, Series B* **74**(2), 245–266.

Tseng, P. (2001), 'Convergence of a block coordinate descent method for nondifferentiable minimization', *Journal of Optimization Theory and Applications* **109**(3), 475–494.

Wang, H. and Leng, C. (2008), 'A note on adaptive group lasso', *Computational Statistics & Data Analysis* **52**(12), 5277–5286.

Wei, F. and Huang, J. (2010), 'Consistent group selection in high-dimensional linear regression', *Bernoulli* **16**(4), 1369.

Wu, T. T., Lange, K. et al. (2010), 'The MM alternative to EM', *Statistical Science* **25**(4), 492–505.

Yang, Y. and Zou, H. (2012), 'A fast unified algorithm for solving group-lasso penalized learning problems', *Statistics and Computing* . to appear.

Yip, K. C. and Yau, K. K. (2005), 'On modeling claim frequency data in general insurance with extra zeros', *Insurance: Mathematics and Economics* **36**(2), 153–163.

Yuan, M. and Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society, Series B* **68**(1), 49–67.

Zhang, Y. (2013*a*), '`cplm`: Compound Poisson linear models'. A vignette for R package `cplm`. Available from http://cran.r-project.org/web/packages/cplm.

Zhang, Y. (2013*b*), 'Likelihood-based and Bayesian methods for Tweedie compound poisson linear mixed models', *Statistics and Computing* **23**(6), 743–757.

Zou, H. (2006), 'The adaptive lasso and its oracle properties', *Journal of the American Statistical Association* **101**(476), 1418–1429.

Zou, H. and Hastie, T. (2005), 'Regularization and variable selection via the elastic net', *Journal of the Royal Statistical Society, Series B* **67**(2), 301–320.