

## Tweedie's Compound Poisson Model With Grouped Elastic Net

Wei Qian, Yi Yang & Hui Zou

To cite this article: Wei Qian, Yi Yang & Hui Zou (2016) Tweedie's Compound Poisson Model With Grouped Elastic Net, Journal of Computational and Graphical Statistics, 25:2, 606-625, DOI: [10.1080/10618600.2015.1005213](https://doi.org/10.1080/10618600.2015.1005213)

To link to this article: <http://dx.doi.org/10.1080/10618600.2015.1005213>



View supplementary material [↗](#)



Accepted author version posted online: 06 Mar 2015.  
Published online: 10 May 2016.



Submit your article to this journal [↗](#)



Article views: 102



View related articles [↗](#)



View Crossmark data [↗](#)



# Tweedie's Compound Poisson Model With Grouped Elastic Net

Wei QIAN, Yi YANG, and Hui ZOU

Tweedie's compound Poisson model is a popular method to model data with probability mass at zero and nonnegative, 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. Integrated 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 nonnegative 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

$$Y = \sum_{i=1}^N X_i, \quad (1)$$

---

Wei Qian is Assistant Professor, School of Mathematical Sciences, Rochester Institute of Technology, Rochester, NY 14623 (E-mail: [wxqsm@rit.edu](mailto:wxqsm@rit.edu)). Yi Yang is Assistant Professor, Department of Mathematics and Statistics, McGill University, Canada (E-mail: [yi.yang6@mcgill.ca](mailto:yi.yang6@mcgill.ca)) Hui Zou is Professor of Statistics, School of Statistics, University of Minnesota, Minneapolis, MN 55455 (E-mail: [zouxx019@umn.edu](mailto:zouxx019@umn.edu)).

© 2016 *American Statistical Association, Institute of Mathematical Statistics,*  
and *Interface Foundation of North America*

*Journal of Computational and Graphical Statistics*, Volume 25, Number 2, Pages 606–625

DOI: [10.1080/10618600.2015.1005213](https://doi.org/10.1080/10618600.2015.1005213)

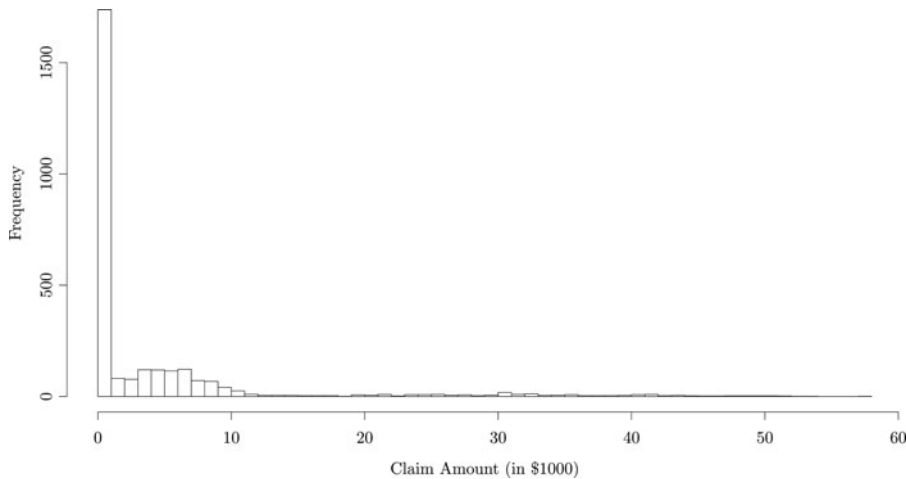


Figure 1. Histogram of an auto insurance claim data.

where  $N$  is a Poisson random variable with mean  $\xi$ , and conditional on  $N$ ,  $X_i$ 's ( $1 \leq i \leq N$ ) are iid Gamma( $\alpha, \gamma$ ) 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  $P(Y = 0) = P(N = 0) = \exp(-\xi)$ .

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  $i$ th claim (e.g., Smyth and Jørgensen 2002; Zhang 2013b). 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  $i$ th event (e.g., Dunn 2004). Also, data with patterns of the Tweedie distribution often arise 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  $i$ th 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 (Jørgensen 1987), which has the form

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

where  $a(\cdot)$  and  $\kappa(\cdot)$  are given functions,  $\theta$  is a parameter in  $\mathbb{R}$ , and  $\phi$  is the dispersion parameter in  $(0, +\infty)$ . By the well-known property of exponential family distributions,  $\mu := E(Y) = \dot{\kappa}(\theta)$  and  $\text{var}(Y) = \phi\ddot{\kappa}(\theta)$ , 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  $\text{var}(Y) = \phi\mu^\rho$ , where  $\rho$  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

$$f(y|\mu, \rho, \phi) = a(y, \phi) \exp \left( \frac{1}{\phi} \left( \frac{y\mu^{1-\rho}}{1-\rho} - \frac{\mu^{2-\rho}}{2-\rho} \right) \right). \quad (3)$$

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 Sections 2 and 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  $\mu$  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  $\beta_0$  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, for example, Murphy, Brockman, and Lee (2000) and Ohlsson and Johansson (2010, sec. 1.3). Let  $\{(y_i, \mathbf{x}_i), i = 1, 2, \dots, n\}$  be the iid observations with sample size  $n$ . Further assume that  $\phi$  is the same for all observations. Then, the *negative* log-likelihood can be written as

$$l(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^n v_i \left( \frac{y_i e^{-(\rho-1)(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i)}}{\rho-1} + \frac{e^{(2-\rho)(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i)}}{2-\rho} \right), \quad (4)$$

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, Presnell, and Turlach 2000) and coordinate descent (Tseng 2001; Friedman, Hastie, and Tibshirani 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 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 are 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, for example, Bach (2008), Nardi et al. (2008), Wang and Leng (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 have been studied for the grouped lasso usually under the linear regression and logistic regression settings. Yuan and Lin (2006) showed 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 proposed 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, Kim, and Kim (2006), who proposed a gradient descent algorithm to solve the constrained-form problem. Meier, Van De Geer, and Bühlmann (2008) also proposed 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, Hastie, and Tibshirani 2010). A blockwise ma-

jorization 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 online 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, Meyers, and Cummings (2011) proposed 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 Sections 4 and 5, respectively. A brief conclusion is given in Section 6.

## 2. ALGORITHM

Assume the  $p$ -dimensional coefficient vector  $\beta$  is partitioned into  $g$  blocks, that is,  $\beta = (\beta_1^T, \beta_2^T, \dots, \beta_g^T)^T$ , where  $\beta_j$  ( $1 \leq j \leq g$ ) 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

$$(\hat{\beta}_0, \hat{\beta}) = \operatorname{argmin}_{(\beta_0, \beta)} l(\beta_0, \beta) + \lambda \sum_{j=1}^g \left( \tau w_j \|\beta_j\|_2 + \frac{1}{2} (1 - \tau) \|\beta_j\|_2^2 \right), \quad (5)$$

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

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

where

$$\begin{aligned} \tilde{v}_i &= v_i \left( (\rho-1)y_i e^{-(\rho-1)(\tilde{\beta}_0 + \tilde{\beta}^T \mathbf{x}_i)} + (2-\rho)e^{(2-\rho)(\tilde{\beta}_0 + \tilde{\beta}^T \mathbf{x}_i)} \right), \\ \tilde{y}_i &= \tilde{\beta}_0 + \tilde{\beta}^T \mathbf{x}_i + \frac{v_i}{\tilde{v}_i} \left( y_i e^{-(\rho-1)(\tilde{\beta}_0 + \tilde{\beta}^T \mathbf{x}_i)} - e^{(2-\rho)(\tilde{\beta}_0 + \tilde{\beta}^T \mathbf{x}_i)} \right), \end{aligned} \quad (7)$$

and  $C_1(\tilde{\beta}_0, \tilde{\beta})$  is a constant given  $(\tilde{\beta}_0, \tilde{\beta})$ . Therefore, we can rewrite (6) in the form of a WLS function  $l_Q(\beta_0, \beta) = \frac{1}{2} \sum_{i=1}^n \tilde{v}_i (\tilde{y}_i - \beta_0 - \beta^T \mathbf{x}_i)^2 + C(\tilde{\beta}_0, \tilde{\beta})$ , where  $C(\tilde{\beta}_0, \tilde{\beta})$  is a constant given  $(\tilde{\beta}_0, \tilde{\beta})$ . 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

$$P_Q(\beta_0, \beta) := l_Q(\beta_0, \beta) + \lambda \sum_{j=1}^g \left( \tau w_j \|\beta_j\|_2 + \frac{1}{2} (1-\tau) \|\beta_j\|_2^2 \right). \quad (8)$$

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

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  $i$  ( $1 \leq i \leq n$ ), partitioning  $\mathbf{x}_i$  the same way as  $\beta$ , we have  $\mathbf{x}_i = (\mathbf{x}_{i1}^T, \mathbf{x}_{i2}^T, \dots, \mathbf{x}_{ig}^T)^T$ , where  $\mathbf{x}_{ij}$  ( $1 \leq j \leq g$ ) is the  $p_j$ -dimensional predictor vector corresponding to  $\beta_j$ . For  $1 \leq j \leq g$ , denote the gradient vector and the Hessian matrix of

$l_Q(\beta_0, \boldsymbol{\beta})$  with respect to  $\boldsymbol{\beta}_j$  by

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

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

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

To update the block  $j$  coefficients ( $1 \leq j \leq g$ ), suppose  $(\check{\beta}_0, \check{\boldsymbol{\beta}})$  is the most recently updated estimate and define  $\check{U}_j = \tilde{U}_j(\check{\beta}_0, \check{\boldsymbol{\beta}})$ . We update  $\check{\boldsymbol{\beta}}_j$  by solving

$$\begin{aligned} \operatorname{argmin}_{\boldsymbol{\beta}_j} l_Q(\check{\beta}_0, \check{\boldsymbol{\beta}}) + \check{U}_j^T (\boldsymbol{\beta}_j - \check{\boldsymbol{\beta}}_j) + \frac{\tilde{\gamma}_j}{2} (\boldsymbol{\beta}_j - \check{\boldsymbol{\beta}}_j)^T (\boldsymbol{\beta}_j - \check{\boldsymbol{\beta}}_j) \\ + \lambda \left( \tau w_j \|\boldsymbol{\beta}_j\|_2 + \frac{1}{2} (1 - \tau) \|\boldsymbol{\beta}_j\|_2^2 \right), \end{aligned} \quad (11)$$

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

$$\check{\boldsymbol{\beta}}_j^{(\text{new})} = \frac{(\tilde{\gamma}_j \check{\boldsymbol{\beta}}_j - \check{U}_j) \left( 1 - \frac{\lambda \tau w_j}{\|\tilde{\gamma}_j \check{\boldsymbol{\beta}}_j - \check{U}_j\|_2} \right)_+}{\tilde{\gamma}_j + \lambda(1 - \tau)}, \quad (12)$$

where  $z_+$  denotes the positive part of  $z$ . Similarly, to update the intercept term, define  $\check{U}_0 = - \sum_{i=1}^n \tilde{v}_i (\tilde{y}_i - \check{\beta}_0 - \check{\boldsymbol{\beta}}^T \mathbf{x}_i)$  and  $\tilde{\gamma}_0 = \tilde{H}_0 = \sum_{i=1}^n \tilde{v}_i$ . The intercept coefficient  $\check{\beta}_0$  is then updated by  $\check{\beta}_0^{(\text{new})} = \check{\beta}_0 - \tilde{\gamma}_0^{-1} \check{U}_0$ . 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  $j$  ( $1 \leq j \leq g$ ), suppose  $(\check{\beta}_0, \check{\boldsymbol{\beta}}^{(\text{new})})$  is the updated vector from  $(\check{\beta}_0, \check{\boldsymbol{\beta}})$  by (11). Then,

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

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,

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



$$\begin{aligned}
& + \lambda \left( \tau w_j \|\check{\beta}_j(\text{new})\|_2 + \frac{1}{2}(1 - \tau)\|\check{\beta}_j(\text{new})\|_2^2 \right) - l_Q(\check{\beta}_0, \check{\beta}) \\
& - \lambda \left( \tau w_j \|\check{\beta}_j\|_2 + \frac{1}{2}(1 - \tau)\|\check{\beta}_j\|_2^2 \right) \\
& \leq 0,
\end{aligned}$$

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

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

---

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

---

1. Initialize  $\check{\beta}_0$  and  $\check{\beta}$ .
  2. (Outer layer) Update the penalized WLS objective function (8).
    - For  $i = 1, 2, \dots, n$ , compute the working response  $\tilde{y}_i$  and the working weight  $\tilde{v}_i$  by (7).
    - For  $j = 1, \dots, g$ , 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  $\check{\beta}_0 = \check{\beta}_0$  and  $\check{\beta} = \check{\beta}$ .
    - Repeat the following updating scheme until  $(\check{\beta}_0, \check{\beta})$  converges.
      - Update  $\check{\beta}$ . For  $j = 1, 2, \dots, g$ , do
        - \* Compute  $\check{U}_j = \check{U}_j(\check{\beta}_0, \check{\beta})$  by (9).
        - \* Compute  $\check{\beta}_j(\text{new})$  by (12).
        - \* Set  $\check{\beta}_j = \check{\beta}_j(\text{new})$ .
      - Update  $\check{\beta}_0$ . Do
        - \* Compute  $\check{U}_0 = -\sum_{i=1}^n \tilde{v}_i(\tilde{y}_i - \check{\beta}_0 - \check{\beta}^T \mathbf{x}_i)$ .
        - \* Compute  $\check{\beta}_0(\text{new}) = \check{\beta}_0 - \tilde{\gamma}_0^{-1} \check{U}_0$ .
        - \* Set  $\check{\beta}_0 = \check{\beta}_0(\text{new})$ .
      - Set  $\check{\beta}_0 = \check{\beta}_0$  and  $\check{\beta} = \check{\beta}$ .
    - 4. Repeat Steps 2–3 until  $(\check{\beta}_0, \check{\beta})$  converges.
-

### 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  $\lambda$  value. Specifically, given  $\tau$ , we consider a decreasing sequence of  $\lambda$  values  $\{\lambda_k, k = 1, \dots, m\}$ . The grouped elastic net solution (5) of  $\lambda_k$  is denoted by  $(\hat{\beta}_0^{(k)}, \hat{\beta}^{(k)})$ . The sequence  $\{\lambda_k, k = 1, \dots, m\}$  is created by choosing a grid of  $m$  points uniformly in log scale on  $[\lambda_m, \lambda_1]$ , where  $\lambda_m$  is a fixed small proportion of  $\lambda_1$ , and  $\lambda_1$  is chosen to be the smallest value such that  $\hat{\beta} = \mathbf{0}$ . The default in the R package `HDtweedie` is  $m = 100$ ,  $\lambda_m = 0.001\lambda_1$  if  $p \leq n$  and  $\lambda_m = 0.05\lambda_1$  if  $p > n$ .

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

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

With  $\lambda_1$  and  $(\hat{\beta}_0^{(1)}, \hat{\beta}^{(1)})$  at hand, we can determine the decreasing sequence of  $\lambda_k$ 's and compute the grouped elastic net solutions sequentially by Algorithm 1. At each  $\lambda_k$  ( $2 \leq k \leq m$ ), the algorithm is “warm-started” by setting the initial coefficient estimate to be  $(\hat{\beta}_0^{(k-1)}, \hat{\beta}^{(k-1)})$ , the solution of the preceding  $\lambda$ . 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) showed 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

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

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

At  $\lambda_k$ , 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  $\{(y_i, \mathbf{x}_{iS}), i = 1, \dots, n\}$  to estimate  $\beta_0$  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  $(\beta_0, \beta)$  by  $(\tilde{\beta}_0^{(*)}, \tilde{\beta}^{(*)})$ . To check if the strong rule correctly identifies the zero estimates, we have to apply a KKT condition check, that is, if  $(\tilde{\beta}_0^{(*)}, \tilde{\beta}^{(*)})$  is the correct solution, then for every  $j \in S^c$ ,  $\|U_j(\tilde{\beta}_0^{(*)}, \tilde{\beta}^{(*)})\|_2 \leq \lambda_k \tau w_j$ . Define  $V = \{j \in S^c : \|U_j(\tilde{\beta}_0^{(*)}, \tilde{\beta}^{(*)})\|_2 > \lambda_k \tau w_j\}$ , the

set of blocks in  $S^c$  that does not pass the KKT condition check. If  $V = \emptyset$ , 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).

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

---

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

---

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

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

2. Initialize  $\tilde{\beta}_0 = \hat{\beta}_0^{(k-1)}$  and  $\tilde{\beta} = \hat{\beta}^{(k-1)}$ .
3. Apply Algorithm 1 on the reduced dataset  $\{(y_i, \mathbf{x}_{iS}), 1 \leq i \leq n\}$  to obtain the strong rule estimate  $(\tilde{\beta}_0^{(*)}, \tilde{\beta}^{(*)})$ .
4. Perform the KKT condition check and identify the set of blocks that fails the check:

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

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

## 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 eight-dimensional covariates  $\mathbf{T} = (T_1, \dots, T_8)$  from a certain distribution scenario. Then, similar to simulation settings of Kim et al. (2006), each covariate  $T_j$  ( $j = 1, \dots, 8$ ) 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 ( $j = 1, \dots, 8$ ). 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  $\lambda$  is selected by the five-fold cross-validations with deviance. For the grouped elastic net, the additional tuning parameter  $\tau$  is selected from  $\{0.1, 0.2, \dots, 1.0\}$ .

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  $\Sigma_1$ . Let  $(\Sigma_1)_{ij} = \omega$  ( $i \neq j$ , and  $i, j = 1, \dots, 8$ ), where  $\omega = 0$  or  $0.5$ . The link function is

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

Clearly, the true model has three relevant blocks and nine 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  $\mathbf{Z} = (Z_1, \dots, Z_6)$  be a multivariate normal random variable with the mean being  $\mathbf{0}$  and the variance being a compound symmetry correlation matrix  $\Sigma_2$ . Let  $(\Sigma_2)_{ij} = \omega$  ( $i \neq j$ , and  $i, j = 1, \dots, 6$ ), 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  $T_j = Z_{j-2}$  ( $j = 4, \dots, 8$ ), where  $\varepsilon_1, \varepsilon_2, \varepsilon_3$  are  $\text{Normal}(0, 0.01)$ . 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

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

	Block-		Coefficient-		Negative log-likelihood	Gini index
	C	IC	C	IC		
Case 1						
Oracle $\omega = 0$	3	0	9	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 $\omega = 0.5$	3.00	0.69	9.00	2.07	35 (15)	0.975 (0.002)
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 $\omega = 0$	3	0	9	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 $\omega = 0.5$	2.77	0.11	8.31	0.33	8.8 (0.6)	0.899 (0.010)
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 $\omega = 0$	6	0	6	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 $\omega = 0.5$	6.00	1.23	6.00	15.69	5.64 (0.02)	0.599 (0.003)
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)

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:

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

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

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

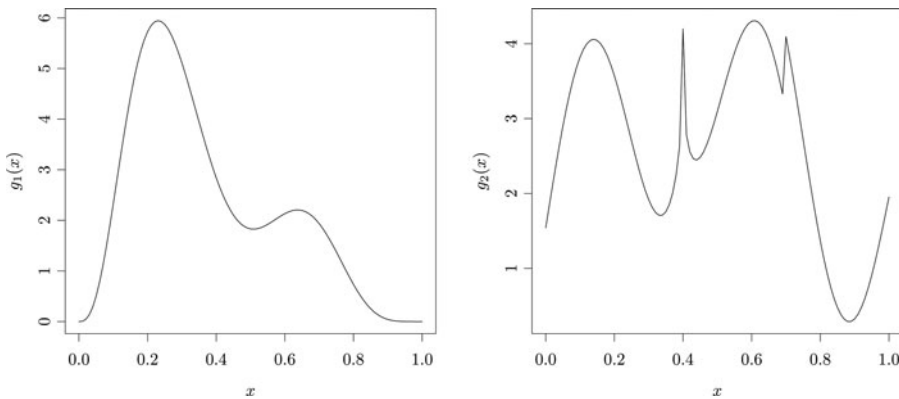
Recall that the strong rule is integrated into the proposed algorithm. Next, we use settings of Case 3 to evaluate the computing time reduction due to the strong rule. For evaluation of higher-dimensional situations, we add  $q$  more irrelevant variables into the original design matrix ( $q = 0, 10, 100, 500, 1000$ ). The distribution of these irrelevant variables are iid 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 five 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.

## 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  $\mathbf{X} = (X_1, X_2, \dots, X_{20})$  is multivariate normal with the mean being  $\mathbf{0}$  and the variance matrix being an exponential decay correlation matrix  $\Sigma$ . Let  $(\Sigma)_{i,j} = \omega^{|i-j|}$  ( $i, j = 1, \dots, 20$ ), 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  $g_1(x) = 10^4 x^3 (1-x)^6 [\frac{40}{3} x^8 + \frac{2}{3} (1-x)^4] I(0 \leq x \leq 1)$  and  $g_2(x) = \{2 \sin(4\pi x) - 6[|x - 0.4|^{0.3} - 1.1] - 0.5 \text{sgn}(0.7 - x)\} I(0 \leq x \leq 1)$ . 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.

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

Figure 2. Plots for functions  $g_1$  and  $g_2$ .

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 online 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 Table 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.

## 5. REAL DATA EXAMPLE

In this section, we use an auto insurance claim dataset studied in Yip and Yau (2005) and Zhang (2013a) 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 (2013a), 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 ( $x_6$ , six categories), whether the car color is red ( $x_7$ ), whether the driver's license

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

	Block-		Negative log-likelihood	Gini index
	C	IC		
Case 1				
Oracle $\omega = 0$	12	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 $\omega = 0.5$	1.03	0.41	2231 (16)	0.019 (0.006)
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 $\omega = 0$	12	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 $\omega = 0.5$	1.00	0.33	2380 (13)	0.029 (0.005)
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)

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 ( $x_{17}$ , eight categories), education level ( $x_{18}$ , five 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 (except 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 categorical 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.



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)

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 2014) 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 (2013a):  $x_3, x_9, x_{13}, x_{15}$ , 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  $\rho$  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, Meyers, and Cummings 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.

Next, we briefly explain the ordered Lorenz curve and the associated Gini index proposed by Frees, Meyers, and Cummings (2011), which is especially useful for insurance risk segmentation purposes. In the 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 insur-

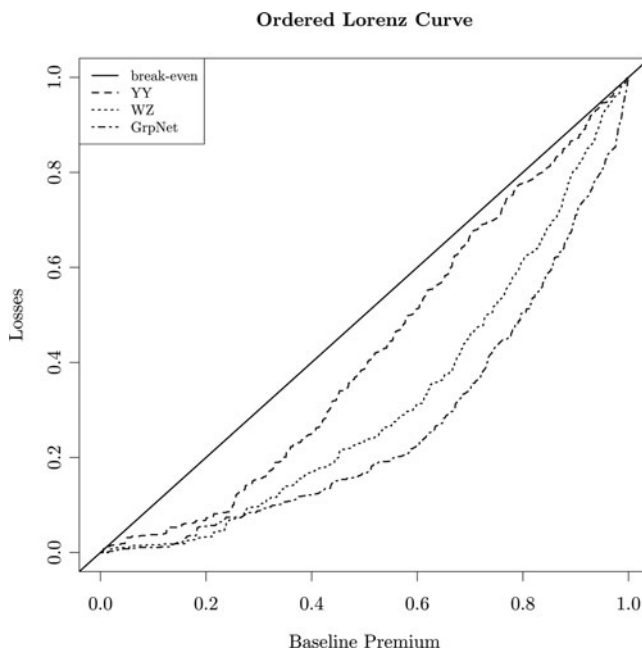


Figure 3. The ordered Lorenz curves for the auto insurance claim data. Larger area between the ordered Lorenz curve and the break-even line means better risk segmentation.

ance score  $S(\mathbf{X})$ . For example, under the settings of Section 2, the Tweedie model with the grouped elastic net chooses  $S(\mathbf{X}) = \exp(\hat{\beta}_0 + \hat{\beta}^T \mathbf{X})$ . 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)  $\{(\tilde{\mathbf{x}}_i, \tilde{y}_i), i = 1, \dots, n\}$  of  $(\mathbf{X}, Y)$ , define two empirical distribution functions

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

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

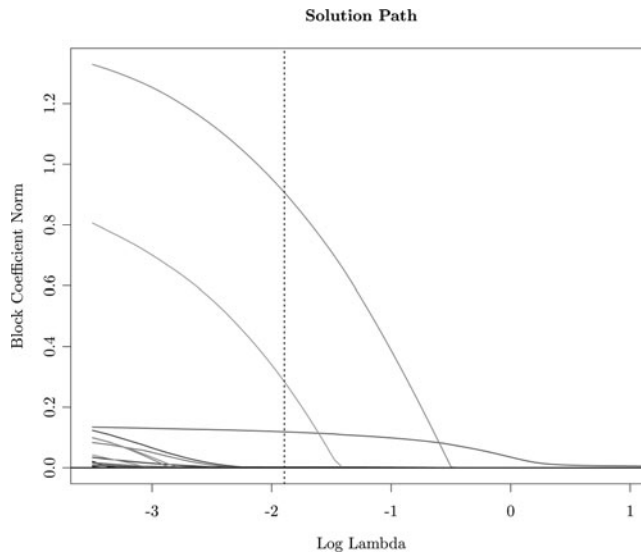


Figure 4. The solution path of the GrpNet model for the auto insurance claim data (the dotted line represents the  $\lambda$  selected by cross-validations;  $\tau = 0.7$ ).

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, available online in the supplementary materials (Figure A2).

## 6. CONCLUSION

Inspired by the success of the IRLS strategy in the coordinate descent algorithms (Friedman, Hastie, and Tibshirani 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)

## ACKNOWLEDGMENTS

The authors thank the Editor, the Associate Editor, and an anonymous referee for their valuable comments that helped improve this article significantly.

[Received October 2013. Revised January 2015.]

## REFERENCES

- Bach, F. R. (2008), “Consistency of the Group Lasso and Multiple Kernel Learning,” *Journal of Machine Learning Research*, 9, 1179–1225. [609]
- Dunn, P. K. (2004), “Occurrence and Quantity of Precipitation Can be Modelled Simultaneously,” *International Journal of Climatology*, 24, 1231–1239. [607]
- Dunn, P. K., and Smyth, G. K. (2005), “Series Evaluation of Tweedie Exponential Dispersion Model Densities,” *Statistics and Computing*, 15, 267–280. [607]
- Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), “Least Angle Regression,” *The Annals of Statistics*, 32, 407–499. [608]
- 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, 533–552. [607]
- Frees, E. W., Meyers, G., and Cummings, A. D. (2011), “Summarizing Insurance Scores Using a Gini Index,” *Journal of the American Statistical Association*, 106, 1085–1098. [610,621]
- Friedman, J., Hastie, T., and Tibshirani, R. (2010), “Regularization Paths for Generalized Linear Models via Coordinate Descent,” *Journal of Statistical Software*, 33, 1–22. [608,609,623]
- Fu, W. J. (1998), “Penalized Regressions: The Bridge Versus the Lasso,” *Journal of Computational and Graphical Statistics*, 7, 397–416. [609]
- Huang, J., and Zhang, T. (2010), “The Benefit of Group Sparsity,” *The Annals of Statistics*, 38, 1978–2004. [609]
- Jørgensen, B. (1987), “Exponential Dispersion Models” (with discussion), *Journal of the Royal Statistical Society, Series B*, 49, 127–162. [607]
- Kim, Y., Kim, J., and Kim, Y. (2006), “Blockwise Sparse Regression,” *Statistica Sinica*, 16, 375. [609,615]
- Lauderdale, B. E. (2012), “Compound Poisson–Gamma Regression Models for Dollar Outcomes That are Sometimes Zero,” *Political Analysis*, 20, 387–399. [607]
- 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, 53–71. [609]
- 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. [608]
- Nardi, Y., and Rinaldo, A. (2008), “On the Asymptotic Properties of the Group Lasso Estimator for Linear Models,” *Electronic Journal of Statistics*, 2, 605–633. [609]
- Ohlsson, E., and Johansson, B. (2010), *Non-Life Insurance Pricing With Generalized Linear Models*, Berlin: Springer. [608]
- 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, 389–403. [608]
- Shono, H. (2008), “Application of the Tweedie Distribution to Zero-Catch Data in CPUE Analysis,” *Fisheries Research*, 93, 154–162. [607]
- 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. [608]

- Smyth, G. K., and Jørgensen, B. (2002), "Fitting Tweedie's Compound Poisson Model to Insurance Claims Data: Dispersion Modelling," *ASTIN Bulletin*, 32, 143–157. [607]
- Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Series B*, 58, 267–288. [608]
- 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, 245–266. [610,614]
- Tseng, P. (2001), "Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization," *Journal of Optimization Theory and Applications*, 109, 475–494. [608]
- Wang, H., and Leng, C. (2008), "A Note on Adaptive Group Lasso," *Computational Statistics & Data Analysis*, 52, 5277–5286. [609]
- Wei, F., and Huang, J. (2010), "Consistent Group Selection in High-Dimensional Linear Regression," *Bernoulli*, 16, 1369. [609]
- Wu, T. T., and Lange, K. (2010), "The MM Alternative to EM," *Statistical Science*, 25, 492–505. [611]
- Yang, Y., and Zou, H. (2014), "A Fast Unified Algorithm for Solving Group-Lasso Penalized Learning Problems," *Statistics and Computing*, DOI: 10.1007/s11222-014-9498-5. [621]
- Yip, K. C., and Yau, K. K. (2005), "On Modeling Claim Frequency Data in General Insurance With Extra Zeros," *Insurance: Mathematics and Economics*, 36, 153–163. [619,621]
- Yuan, M., and Lin, Y. (2006), "Model Selection and Estimation in Regression With Grouped Variables," *Journal of the Royal Statistical Society, Series B*, 68, 49–67. [609]
- Zhang, Y. (2013a), "cplm: Compound Poisson Linear Models," A vignette for R package cplm. Available at <http://cran.r-project.org/web/packages/cplm>. [619,621]
- (2013b), "Likelihood-Based and Bayesian Methods for Tweedie Compound Poisson Linear Mixed Models," *Statistics and Computing*, 23, 743–757. [607]
- Zou, H. (2006), "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429. [609]
- Zou, H., and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society, Series B*, 67, 301–320. [609,616]

# Appendix to “Tweedie’s Compound Poisson Model With Grouped Elastic Net”

Wei Qian, Yi Yang and Hui Zou  
School of Statistics  
University of Minnesota

## A. Additional Numerical Results of Simulation Example 1

We repeat the procedures in section 4.1 with smaller sample size ( $n = 100, 200$  instead of 500), and the results appear to be close to that of sample size 500 in this example. The results are summarized in Tables A1-A3.

## B. Misspecification of Link Function

we use the log link in our implementation to ensure that the estimated conditional mean  $\hat{\mu}(\mathbf{x})$  is always positive, and at the same time, impose a multiplicative structure. In the context of insurance premium estimation, the log link is much used as it decomposes the expected premium into different risk factors in a multiplicative fashion (when only main effects are considered), and generates convenient model for intuitive interpretation. However, if the true data is not generated by the log link, building models with the log link is subject to estimation bias. Inspired by the Associate Editor’s comment, we intend to illustrate this issue by a data simulation setting similar to that of Case 3 in Example 1 except for the link function. Let  $\eta := 0.3 + \sum_{j=1}^6 p_1(T_j)$ . Consider the following two scenarios for the true link functions, from which the data is generated: (A)  $\mu = \frac{[\exp(\eta)+1]^r-1}{r}$ ; (B)  $\mu = \exp(\text{sgn}(\eta)|\eta|^r)$ , where  $r > 0$  and  $\text{sgn}(\cdot)$  is the sign function. Clearly, when  $r = 1$ , both scenarios reduce to the log link. When  $r \neq 1$ , however, the presumed log-link in our algorithm implementation does not match the true link function, resulting in the misspecification of the link function. To generate data, we choose  $r = 0.5, 0.7, 1, 1.3, 1.5$ , and assume  $\omega = 0$ . The corresponding data

Table A1: (Case 1) Averaged simulation results based on 100 runs.

	block-		coefficient-		Gini index
	C	IC	C	IC	
oracle	3	0	9	0	-
$n = 100, \omega = 0$					
lasso	2.87	1.57	5.59	1.94	0.871 (0.018)
grouped lasso	2.93	0.98	8.79	2.94	0.889 (0.015)
grouped elastic net	2.94	1.50	8.82	4.50	0.891 (0.015)
$n = 100, \omega = 0.5$					
lasso	2.81	1.58	5.08	1.89	0.839 (0.024)
grouped lasso	2.93	1.34	8.79	4.02	0.877 (0.011)
grouped elastic net	2.95	1.87	8.85	5.61	0.882 (0.011)
$n = 200, \omega = 0$					
lasso	2.84	1.13	5.55	1.37	0.908 (0.023)
grouped lasso	2.90	0.65	8.70	1.95	0.925 (0.017)
grouped elastic net	2.96	1.13	8.88	3.39	0.948 (0.004)
$n = 200, \omega = 0.5$					
lasso	2.78	1.65	5.46	1.84	0.891 (0.024)
grouped lasso	2.94	1.04	8.82	3.12	0.938 (0.006)
grouped elastic net	2.98	1.42	8.94	4.26	0.941 (0.006)

each has 18 predictor terms, and 6 of them are relevant. We use the lasso method to obtain coefficient-C, coefficient-IC and the mean squared estimation error of  $\eta$  ( $\text{MSE}_\eta$ ). Based on the results summarized in Table A4 and Figure A1, even though the implemented algorithm may still identify the relevant variables, the estimation of  $\eta$  becomes unreliable when the true link function is very different from the log-link. We leave the further investigation regarding robust estimation for the misspecified link function to the future.

Table A2: (Case 2) Averaged simulation results based on 100 runs.

	block-		coefficient-		Gini index
	C	IC	C	IC	
oracle	3	0	9	0	-
$n = 100, \omega = 0$					
lasso	1.61	0.82	2.13	1.08	0.738 (0.017)
grouped lasso	1.42	0.48	4.26	1.44	0.733 (0.020)
grouped elastic net	2.81	0.72	8.43	2.16	0.746 (0.017)
$n = 100, \omega = 0.5$					
lasso	1.67	0.92	2.28	1.11	0.758 (0.019)
grouped lasso	1.55	0.65	4.65	1.95	0.766 (0.018)
grouped elastic net	2.83	0.92	8.49	2.76	0.766 (0.018)
$n = 200, \omega = 0$					
lasso	1.88	0.30	2.59	0.30	0.819 (0.016)
grouped lasso	1.78	0.18	5.34	0.54	0.825 (0.015)
grouped elastic net	2.97	0.30	8.91	0.90	0.825 (0.015)
$n = 200, \omega = 0.5$					
lasso	1.83	0.70	2.46	0.74	0.817 (0.016)
grouped lasso	1.73	0.43	5.19	1.29	0.823 (0.015)
grouped elastic net	2.89	0.55	8.67	1.65	0.823 (0.015)

### C. Partial Fits in Real Data Example

Based on the auto insurance claim data example in section 5, Figure A2 shows the partial fits of the variables selected by the Tweedie model with grouped elastic net (GrpNet).



Table A3: (Case 3) Averaged simulation results based on 100 runs.

	block-		coefficient-		Gini index
	C	IC	C	IC	
oracle	6	0	6	0	-
$n = 100, \omega = 0$					
lasso	5.86	0.68	5.73	4.29	0.541 (0.009)
grouped lasso	5.81	0.65	5.81	13.57	0.525 (0.010)
grouped elastic net	5.97	1.12	5.97	15.30	0.540 (0.007)
$n = 100, \omega = 0.5$					
lasso	5.31	0.79	4.66	4.30	0.360 (0.015)
grouped lasso	4.76	0.48	4.76	10.96	0.317 (0.014)
grouped elastic net	5.52	0.91	5.52	13.77	0.369 (0.010)
$n = 200, \omega = 0$					
lasso	6.00	0.78	5.99	4.26	0.591 (0.004)
grouped lasso	5.99	0.94	5.99	14.80	0.580 (0.004)
grouped elastic net	6.00	1.32	6.00	15.96	0.582 (0.004)
$n = 200, \omega = 0.5$					
lasso	5.92	0.72	5.80	4.34	0.416 (0.006)
grouped lasso	5.85	0.68	5.85	13.74	0.403 (0.006)
grouped elastic net	6.00	0.99	6.00	14.97	0.416 (0.004)

Table A4: Averaged simulation results for link function misspecification based on 100 runs.

Scenario	$r$	coefficient-		$\text{MSE}_\eta$
		C	IC	
A	0.5	5.90	4.04	0.457
	0.7	5.97	4.20	0.313
	1.0	6.00	4.41	0.160
	1.3	6.00	4.60	0.162
	1.5	6.00	4.59	0.256
B	0.5	5.88	4.04	0.546
	0.7	5.94	4.12	0.391
	1.0	6.00	4.41	0.160
	1.3	6.00	5.44	0.165
	1.5	6.00	6.13	0.534

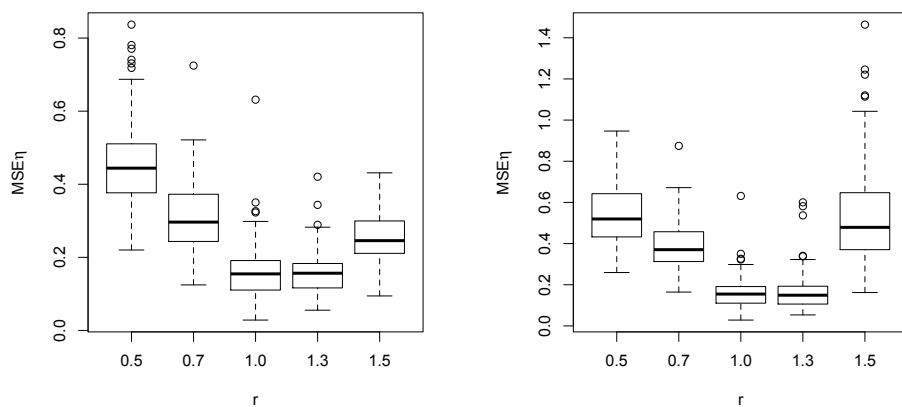


Figure A1: Boxplots of  $\text{MSE}_\eta$  for link function misspecification. Left panel: scenario (i). Right panel: scenario (ii).

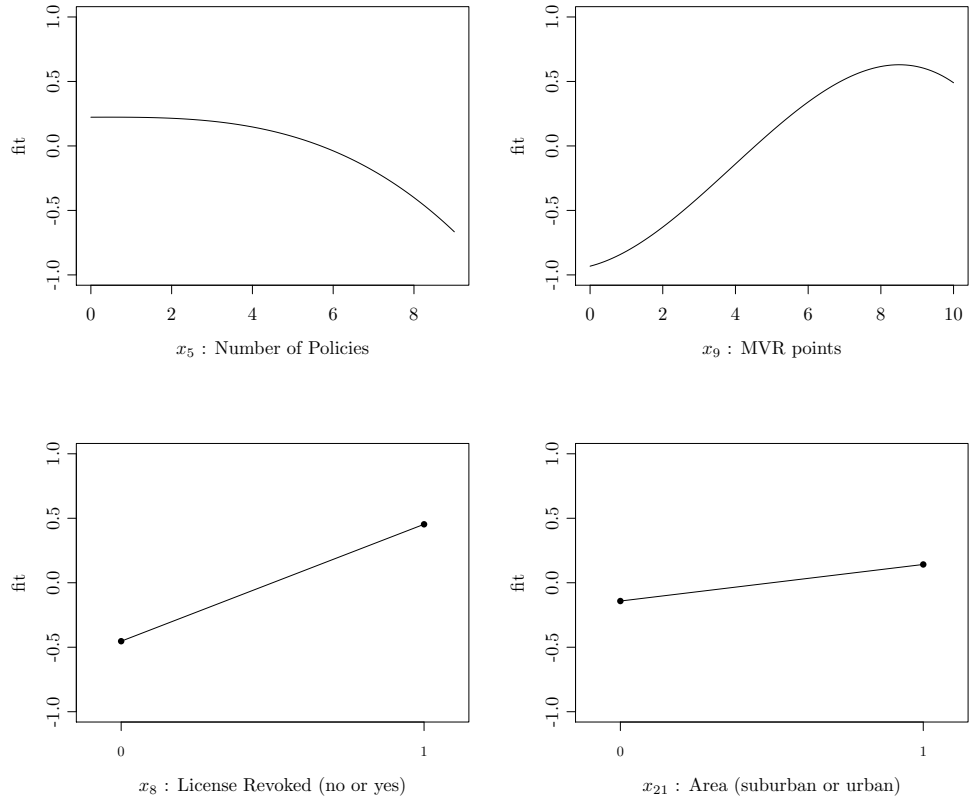


Figure A2: The partial fits of the selected variables by the GrpNet model for the auto insurance claim data.