MCGILL UNIVERSITY

Math 680 Final Project

Reproduce Tweedie's Compound Poisson Model With Grouped Elastic Net

Prepared by:
Mengtian Zhang and Menglan Pang

December 22, 2016

INTRODUCTION

This project aims to implement and reproduce the algorithm for computing grouped elastic net solutions for tweedie's compound poisson model and the simulation studies proposed in [1], and to design a new simulation study that demonstrate its real-world application. We will investigate its model selection ability by comparing it with LASSO and grouped LASSO penalties.

STATISTICAL MODELS

Tweedie's compound poisson model takes the form of

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

where N is poisson distributed with parameter ξ and X_i's are iid $Gamma(\alpha,\gamma)$.It is often used for modeling highly right-skewed data with probability mass at zero and nonnegative support.

The probability density function of tweedie's compound poisson model can be written as

$$f(y|\mu,\rho,\phi) = a(y,\phi)e^{\frac{1}{\phi}(\frac{y\mu^{1-\rho}}{1-\rho}-\frac{\mu^{2-\rho}}{2-\rho})}, \tag{1}$$

where μ is the mean, $1 < \rho < 2$ is the power parameter, $\phi \in (0,\infty)$ is the dispersion parameter and we have $\xi = \frac{\mu^{2-\rho}}{\phi(2-\rho)}$,$\alpha = \frac{2-\rho}{\rho-1}$,$\gamma = \phi(\rho-1)\mu^{\rho-1}$. Our goal here is to find a generalized linear model that capture the mean, μ, of Y. For convenience, $\log(\mu)$ is used as link function and therefore the generalized linear model can be written as $\log(\mu)$=$\beta_0 + \boldsymbol{x\beta}$, $\boldsymbol{\beta} \in \mathbb{R}^p$. Substituting μ with $e^{\beta_0+\boldsymbol{x\beta}}$ in (1), it is easy to obtain the negative log-likelihood function for β_0 and $\boldsymbol{\beta}$ given iid observations $y_i, x_i{}_{i=1}^{n}$,

$$l(\beta_0,\boldsymbol{\beta}) = \sum_{i=1}^{n} v_i(\frac{y_i e^{-(\rho-1)(\beta_0+\boldsymbol{x}_i\boldsymbol{\beta})}}{1-\rho} - \frac{e^{(2-\rho)(\beta_0+\boldsymbol{x}_i\boldsymbol{\beta})}}{2-\rho}), \tag{2}$$

where v_i's are observation weights and equals $\frac{1}{n}$ by default.

In real applications, however, $\boldsymbol{x}$, may involve spurious predictors for which we want the coefficient estimates to be as close to zero as possible, an efficient model selection method is therefore important when we solve for estimates β_0 and $\boldsymbol{\beta}$. Common used model selection method is LASSO which imposes a l_1 penalty to negative log-likelihood or loss function and shrinks estimated coefficients for spurious variables to zero. In situations where "group"-like predictors, e.g. factors with multiple levels, are included in the proposed model, a better choice for such problems is grouped LASSO which partitions variables into blocks and performs block-wise model selection. Apart from the existence of spurious predictors, another issue is the non-ignorable correlation between variables. Previous studies have showed that it could be addressed by elastic net method which adds additional l_2 penalties to l_1 penalties. In the light of above discussion, we adopt grouped elastic net method combining both features to handle the problem of minimizing (2).

ALGORITHMS

Let $\boldsymbol{\beta} \in \mathbb{R}^p$ be partitioned in the g blocks, i.e. $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_i^T, \dots, \boldsymbol{\beta}_g^T)^T$ and let $\boldsymbol{\beta}_j^T \in \mathbb{R}^{p_j}$ with $\sum_{j=1}^{g} p_j = p$. The objective function we aims to minimize with respect to β_0 and $\boldsymbol{\beta}$ is

$$l(\beta_0, \boldsymbol{\beta}) + \lambda \sum_{j=1}^{g} (\tau \omega_j \|\boldsymbol{\beta}_j\|_2 + \frac{1}{2}(1-\tau)\|\boldsymbol{\beta}_j\|_2^2), \tag{3}$$

where $\lambda > 0$ and $0 < \tau \leq 1$ are tuning parameters, and ω_j's are weights for grouped LASSO penalties which by default equals $\sqrt{p_j}$ for $j = 1, 2, \dots, g$. By varying the value of parameters, we may obtain solutions under different methods. For example, if $p_j = 1 \forall j$ and $\tau = 1$, it becomes a LASSO minimization problem and if $p_j > 1$ for some j and $\tau = 1$, then it is grouped LASSO. For (grouped) elastic net, τ takes value between 0 and 1.

As proposed in [1], we consider a two-layer minimization strategy, that is, at each iteration, we first approximate the log-likelihood function (2) using second-order Taylor expansion about the solution of $(\beta_0, \boldsymbol{\beta})$ from the most recent iteration $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ and obtain the penalized weighted least squares (WLS) objective function $P_Q(\beta_0, \boldsymbol{\beta})$ as outer layer. Inside the outer layer, we find the minimizer of $P_Q(\beta_0, \boldsymbol{\beta})$ by updating $\boldsymbol{\beta}_j$ for $j = 1, 2, \dots, g$ until convergence and then update the working weight and working response v_i and y_i for $i = 1, 2, \dots, n$ for the next iteration. The following paragraphs elaborate this idea in detail.

OUTER LAYER

For the outer layer, a penalized WLS approximation of (3) is

$$P_Q(\beta_0, \boldsymbol{\beta}) = l_Q(\beta_0, \boldsymbol{\beta}) + \lambda \sum_{j=1}^{g} (\tau \omega_j \|\boldsymbol{\beta}_j\|_2 + \frac{1}{2}(1-\tau)\|\boldsymbol{\beta}_j\|_2^2), \tag{4}$$

where

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

where

$$\begin{aligned} \tilde{v}_i &= v_i(y_i e^{-(\rho-1)(\beta_0+\boldsymbol{x}_i\boldsymbol{\beta})} + e^{(2-\rho)(\beta_0+\boldsymbol{x}_i\boldsymbol{\beta})}) \\ \tilde{y}_i &= \tilde{\beta}_0 + \boldsymbol{x}_i\boldsymbol{\beta}\frac{v_i}{\tilde{v}_i}(y_i e^{-(\rho-1)(\beta_0+\boldsymbol{x}_i\boldsymbol{\beta})} + e^{(2-\rho)(\beta_0+\boldsymbol{x}_i\boldsymbol{\beta})}) \end{aligned} \tag{5}$$

INNER LAYER

After obtaining the approximation, we proceed to find the minimizer of (3) in the inner layer taking the advantage of majorization-minimization (MM) principle and update the coefficients of each block sequentially. Assume that $(\breve{\beta}_0, \breve{\boldsymbol{\beta}}^T)^T$ is the most recent updated estimate. It is clear that for each j,

$$\begin{aligned} & l_Q(\beta_0, \boldsymbol{\beta}) + \lambda(\tau\omega_j\|\boldsymbol{\beta}_j\|_2 + \frac{1}{2}(1-\tau)\|\boldsymbol{\beta}_j\|_2^2) \\ & \leq l_Q(\breve{\beta}_0, \breve{\boldsymbol{\beta}}) + \breve{U}_j^T(\boldsymbol{\beta}_j - \breve{\boldsymbol{\beta}}_j) + \frac{\tilde{\gamma}_j}{2}(\boldsymbol{\beta}_j - \breve{\boldsymbol{\beta}}_j)^T(\boldsymbol{\beta}_j - \breve{\boldsymbol{\beta}}_j) + \lambda(\tau\omega_j\|\boldsymbol{\beta}_j\|_2 + \frac{1}{2}(1-\tau)\|\boldsymbol{\beta}_j\|_2^2) \end{aligned} \tag{6}$$

where

$$\breve{U}_j = U_j|_{\breve{\beta}_0,\breve{\boldsymbol{\beta}}} = \frac{\partial l_Q(\beta_0, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j}|_{\breve{\beta}_0,\breve{\boldsymbol{\beta}}} = -\sum_{i=1}^{n} \tilde{v}_i(\tilde{y}_i - \beta_0 - \boldsymbol{x}_i\boldsymbol{\beta}_j)\boldsymbol{x}_{ij}|_{\breve{\beta}_0,\breve{\boldsymbol{\beta}}}$$

and $\tilde{\gamma}_j$ is the greatest eigenvalue of $\tilde{H}_j = \frac{\partial l_Q(\beta_0,\boldsymbol{\beta})}{\partial\boldsymbol{\beta}_j\partial\boldsymbol{\beta}_j^T} = \sum_{i=1}^n \tilde{v}_i\boldsymbol{x}_{ij}\boldsymbol{x}_{ij}^T$. We update $\breve{\boldsymbol{\beta}}_j$ by finding the minimizer of RHS of (6), which can be written as

$$\begin{aligned} \breve{\boldsymbol{\beta}}_j^{(new)} &= \frac{(\tilde{\gamma}_j\breve{\boldsymbol{\beta}}_j - \breve{U}_j)(1 - \frac{\lambda\tau\omega_j}{\|\tilde{\gamma}_j\breve{\beta}_j - \breve{U}_j\|_2})_+}{\tilde{\gamma}_j - \lambda(1-\tau)} \\ \breve{\beta}_0^{(new)} &= \breve{\beta}_0 - \tilde{\gamma}_0^{-1}\breve{U}_0 \end{aligned}$$

where $\breve{U}_0 = -\sum_{i=1}^n \tilde{v}_i(\tilde{y}_i - \beta_0 - \boldsymbol{x}_i\boldsymbol{\beta}_j)$ and $\tilde{\gamma}_0 = \tilde{H}_0 = \sum_{i=1}^n \tilde{v}_i$. We keep updating until convergence. After obtaining the minimizer of $P_Q(\beta_0, \boldsymbol{\beta})$, we update the working weight and working response for the next iteration.

To efficiently compute the solution path for a given sequence of decreasing λ values $\lambda_1, \ldots, \lambda_m$ where λ_1 is the smallest λ value that yields $\hat{\boldsymbol{\beta}} = \mathbf{0}$, we employ KKT conditions check into above algorithm.

We start with finding $(\hat{\beta}_0^{(1)}, (\hat{\boldsymbol{\beta}}^{(1)})^T)^T$ corresponding to λ_1. $\hat{\boldsymbol{\beta}}^{(1)} = \mathbf{0}$ by definition, and we can therefore initiate $\tilde{\beta}_0^{(1)} = 0$ and continue updating by above algorithm with $\breve{\boldsymbol{\beta}}$ restricted to $\mathbf{0}$. λ_1 is obtain by KKT conditions that $\lambda_1 = \underset{1\leq j\leq g}{max}\|U_j(\hat{\beta}_0^{(1)}, \hat{\boldsymbol{\beta}}^{(1)})\|_2/\tau\omega_j$. A sequence of λ values $\{\lambda_k : \lambda_1, \ldots, \lambda_m\}$ is therefore can be determined given λ_1.

For each λ_k, we initiate $\tilde{\beta}_0 = \hat{\beta}_0^{(k-1)}$ and $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^{(k-1)}$ as a warm start and adopt strong rules to check if the KKT condition

$$\|U_j(\hat{\beta}_0^{(k-1)}, \hat{\boldsymbol{\beta}}^{(k-1)})\|_2 < \tau\omega_j(2\lambda_k - \lambda_{k-1}) \tag{7}$$

holds for $j = 1, 2, \ldots, g$. If (7) holds, then $\boldsymbol{\beta}_j^{(k)}$ is very likely to be zero which can be dropped in order to save the computing time. We apply above algorithm to the reduced data set and obtain $(\tilde{\beta}_0^{(*)}, (\tilde{\boldsymbol{\beta}}^{(*)})^T)^T$ as the reduced estimated coefficients. It is then important to see if the performance of strong rules. We apply again a KKT condition ckeck, i.e. for each j that satisfies (7), we check if $\|U_j(\tilde{\beta}_0^{(*)}, \tilde{\boldsymbol{\beta}}^{(*)})\|_2 < \lambda_k \tau \omega_j$. If all such j pass the above check, then we find the solution for λ_k that $(\hat{\beta}_0^{(k)}, (\hat{\boldsymbol{\beta}}^{(k)})^T)^T = (\tilde{\beta}_0^{(*)}, (\tilde{\boldsymbol{\beta}}^{(*)})^T)^T$; if some j fails to pass the test, we add $\boldsymbol{x}_j$ back to the reduced data set and perform the algorithm again until solution $(\hat{\beta}_0^{(k)}, (\hat{\boldsymbol{\beta}}^{(k)})^T)^T$ is found.

R Function Development and Validation

Based on the algorithms described in the previous section, we developed a R function named *tweedieAlgo1* to solve the grouped elastic net problem for a given λ value. Furthermore, we also created a R function named *tweedieAlgo2* to compute the whole solution path of grouped elastic net for a given vector of λ or for a number of λ values to be considered. We tested our functions with a single dataset. The results provided by our functions are the same as those provided by the **HDtweedie** package which is developed by the authors of the original paper. Therefore, we have validated that the functions developed by us can successfully perform, although the speed is somewhat slower than the **HDtweedie** package which was built based on Fortran. Moreover, we wrote another R function named *cvtweedie* to perform the k-folds cross validation for selecting the tuning parameters λ and τ in the Tweedie model.

Simulation Studies

We consider two simulation studies for this final project. For the first simulation study, we aim to reproduce the example 1 of the simulation study in the paper by Qian, Yang and Zou 2016 [1]. This simulation study is used to investigate the performance in terms of variable selection by the Tweedie model with lasso, grouped lasso, and grouped elastic net methods. However, all the covariates are artificially generated from multivariate normal distribution with mean zero and a certain covariate matrix, therefore we consider a second simulation study based on a real-life dataset in car insurance to better reflect the data structure in a real-world analysis. An additional objective of this simulation study is to evaluate the performance of these models in terms of bias and precision of each of the regression coefficients.

Simulation Study 1

Three difference cases were considered in Simulation study 1.

- General simulation study design for all the three cases

1000 observations were generated in each run of the simulation. We followed the strategy in the paper to create the design matrix. Eight-dimensional covariates $\mathbf{T} = (T_1, ..., T_8)$ were randomly generated from a certain multivariate distributions in each of the three cases. Then, for each covariate $T_j (j = 1, ..., 8)$, we used three polynomial terms $p_1(T_j)$, $p_2(T_j)$, $p_3(T_j)$ to produce three correlated covariates which would naturally form a block. The polynomial functions are given by: $p_1(x) = x$, $p_2 = (3x^2 - 1)/6$, and $p_2 = (5x^3 - 3x)/10$. The resulted design matrix was consisted of 24 terms. The outcome Y was generated by the Tweedie model with log link function, $\rho = 1.5$ and $\phi = 1$ using the random generating function **rtweedie** in R [2]. The corresponding link function is described in the following subsection.

The whole dataset was split into a training dataset and a testing dataset with 500 observations each. We fitted the Tweedie models with lasso, grouped lasso, and grouped elastic net models by using the training dataset to select the tuning parameters λ and τ, and evaluated the performances of these three methods by using the testing dataset.

More specifically, the tuning parameter λ was selected by using a five-fold cross-validations within the training dataset. The sequence of λ was pre-specified as a grid of m=10 values that uniformly located in the log scale on $[\lambda_{10}, \lambda_1]$. The best λ was chosen as the one that minimizes the overall negative log-likelihood. The additional tuning parameter τ in the grouped elastic net method was selected from the sequence {0.1, 0.3,...,0.9}.(Note, m=100, and the sequence for τ is {0.1,0.2,...,1.0} in the original paper and we chose a much smaller number for m and a shorter sequence for τ to reduce the computational time at the possible costs of the performance of each method.)

To investigate the performance of the three methods, the capability of variable selection are assessed based on the blocks as well as the individual covariates. The block of the covariates is defined as active if at least one of predictors was selected within the block, and a individual covariate is defined as active if its estimated coefficient is nonzero. Therefore, four criteria were used to evaluate the variable selection when fitting the models in the testing dataset with the tuning parameters chosen from the training dataset. First, block-C: the number of correctly identified active blocks; Second, block-IC: the number of incorrectly identified active blocks; Third, coefficient-C: the number of correctly identified active coefficients; and Fourth, the number of incorrectly identified active coefficients. The simulation was repeated 50 times, and the average values of the four criteria were calculated over the 50 simulation runs.

- Specifications for each of the cases

Case 1: $\mathbf{T}=(T_1, ..., T_8)$ was assumed to follow a multivariate normal distribution $\mathcal{N}(\mathbf{0}, \Sigma_1)$. The variance matrix Σ_1 is a compound symmetry correlation matrix, where $(\Sigma_1)_{ij} = \omega$ $(i \neq j$, and $i, j = 1, ..., 8)$, and $(\Sigma_1)_{ij} = 1$ $(i = j)$. ω was set to be 0 or 0.5. The link function is

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

In this setting, there were 8 blocks and 24 predictors. The first 3 blocks and the first 9 predictors were active in the true models.

Case 2: In this case, **T** was generated based on **Z**, where **Z**=$(Z_1,...,Z_6)$ was assumed to follow a multivariate normal distribution $\mathcal{N}(\mathbf{0},\Sigma_2)$. The variance matrix Σ_2 is a compound symmetry correlation matrix, where $(\Sigma_2)_{ij} = \omega$ $(i \neq j$, and $i,j = 1,...,6)$, and $(\Sigma_2)_{ij} = 1$ $(i = j)$. ω was set to be 0 or 0.5. Then **T** was generated by $T_1 = Z_1 + \varepsilon_1$, $T_2 = Z_1 + \varepsilon_2$, $T_3 = Z_1 + \varepsilon_3$, and $T_j = Z_{j-2} (j = 4,...,8)$, where ε_1, ε_2, ε_3 were independent Normal(0, 0.01). The link function was the same as that of Case 1. Again in this setting, the first 3 blocks and the first 9 predictor were active among the total 8 blocks and 24 predictors. The additional feature in this case is that the three active blocks are highly correlated with each other, as T_1, T_2, and T_3 were based on the same variable Z_1.

Case 3: The distribution of **T** was assumed to be the same Case 1. The link function is

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

In this scenario, the link function was used to favour the lasso method intentionally. There were again 8 blocks and 24 predictors. But only 6 blocks and 6 predictors were relevant in the true model. The first 6 blocks were active in the true models, and the predictors (1, 4, 7, 10, 13, 16) were truly active.

SIMULATION STUDY 2

This simulation was based on a real-life dataset in car insurance. We found this dataset named **dataCar** in the the R package **insuranceData** [3]. It is based on one-year vehicle insurance policies taken out in 2004 or 2005 that consists of 67,856 observations. We considered 6 relevant variables in our simulation studies, i.e., vehicle value in \$10,000 dollar (continuous variable), vehicle body (categorical variable with 13 levels), vehicle age (continuous variable), gender (binary variable), area (categorical variable with 6 levels), and driver's age (categorical variables with 6 levels). The distribution of these predictors is provided in Table 1. Each of the categorical variables were then represented by their corresponding dummy variables and these dummy variables naturally formed a block. In total, there were 6 blocks with the size of (1, 12, 1, 1, 5, 5), resulted in 25 predictors.

In order to retain the data structure and the correlation of the covariates in this real-life dataset, for each simulation run, we took all the variables as they were from the car insurance dataset and formed them as our design matrix in this simulation. In other words, there was no variation in the design matrix across different simulation runs. However, the outcome variable would be generated randomly for each observation in each simulation run. A log link function was used, i.e., $\log \mu = \beta_0 + \beta X$. We obtained the values for β by fitting a logistic regression using the design matrix as predictors and the binary variable occurrence of claim as outcome to get a sensible relationship between the covariates and the claim in real

Table 1: Descriptive statistics of the variables in the car insurance dataset

N	67,856	Coefficients
Vehicle Value (mean (sd))	1.78 (1.21)	0.05
Vehicle Body (%)		
BUS	48 (0.1)	Ref
CONVT	81 (0.1)	-2.14
COUPE	780 (1.1)	-0.89
HBACK	18,915 (27.9)	-1.13
HDTOP	1,579 (2.3)	-0.94
MCARA	127 (0.2)	-0.56
MIBUS	717 (1.1)	-1.26
PANVN	752 (1.1)	-0.92
RDSTR	27 (0.0)	-1.22
SEDAN	22,233 (32.8)	-1.12
STNWG	16,261 (24.0)	-1.11
TRUCK	1,750 (2.6)	-1.15
UTE	4,586 (6.8)	-1.35
Vehicle Age (mean,(sd))	2.67 (1.07)	-0.01
Gender=M (%)	29,253 (43.1)	-0.01
Area (%)		
A	16,312 (24.0)	Ref
B	13,341 (19.7)	0.10
C	20,540 (30.3)	0.04
D	8,173 (12.0)	-0.09
E	5,912 (8.7)	-0.01
F	3,578 (5.3)	0.11
Driver’s Age (%)		
1	5,742 (8.5)	Ref
2	12,875 (19.0)	-0.20
3	15,767 (23.2)	-0.22
4	16,189 (23.9)	-0.26
5	10,736 (15.8)	-0.45
6	6,547 (9.6)	-0.45

life. The estimates from the logistic regression were set to be the true values for the regression coefficients in the Tweedie model. The corresponding value for each variable was provided in Table 1. Finally, Y was generated by the Tweedie model with $\rho = 1.5$ and $\phi = 1$ using the **rtweedie** function. In addition, we artificially generated a 10-dimensional multivariate standard normal distribution as irrelevant noise variable.

As we have done in simulation 1, we evenly divided the whole dataset into a training dataset to obtain the optimal λ and τ by using 5-fold cross validation, and a testing dataset to fit the three Tweedie models. The simulation was repeated for 50 times, and we obtained the average $\hat{\beta}$ over the 50 simulation runs. The bias, standard error and the root mean squared error were calculated for each of the covariates.

RESULTS

RESULTS OF SIMULATION STUDY 1

The results of the Simulation Study 1 including all the three cases are summarized in Table 2.

In case 1, all the three relevant blocks were selected for almost all the time by the Tweedie models, and on average 2 irrelevant blocks among 6 were selected. It is shown that the grouped lasso and grouped elastic net have better block selection results than the lasso by identifying slightly more relevant blocks and less irrelevant blocks. In terms of the individual coefficient, the grouped lasso and grouped elastic net almost always selected the 9 relevant predictors, as the estimate coefficients were nonzero once the active blocks were selected. On the other hand, lasso identified less relevant predictors (on average 6 out of 9), as some estimated coefficients of an active block may be zero. For the same reason, grouped lasso and grouped elastic net also tended to select more incorrectly identified active coefficients. These results were expected, as the link function was specified to have an explicit blockwise structure.

In case 2, it was shown that the pattern of the results were similar to those in case 1. However, the performance of all the four criteria revealed somewhat worse than the performance in case 1. Moreover, the grouped elastic net only select slightly more relevant blocks on average compared to grouped lasso (average block-C: 2.61, and 2.5 by grouped elastic net vs. 2.54 and 2.46 by grouped lasso). It seemed that the advantage of grouped elastic net in the scenario of correlated covariates was subtle in our simulation. More investigation was needed to explain these unexpected results.

In case 3, all three methods correctly identified all six relevant blocks and coefficients. However, the chance of selecting the irrelevant blocks and coefficients was also very high by all three models, although lasso had better performance among all the methods.

Table 2: Average results of simulation study 1

	Block-		Coefficient-	
	C	IC	C	IC
	Case 1			
Oracle	3	0	9	0
ω=0				
Lasso	2.84	1.79	6.09	2.53
Grouped lasso	2.86	1.7	8.58	5.1
Grouped elastic net	2.86	1.7	8.58	5.1
ω=0.5				
Lasso	2.77	2.24	6.22	3
Grouped lasso	2.83	1.94	8.49	5.82
Grouped elastic net	2.83	2.02	8.49	6.06
	Case 2			
Oracle	3	0	9	0
ω=0				
Lasso	2.54	2.58	4.56	4.94
Grouped lasso	2.54	2.6	7.62	7.8
Grouped elastic net	2.61	2.56	7.83	7.68
ω=0.5				
Lasso	2.44	2.66	4.19	4.57
Grouped lasso	2.46	2.66	7.38	7.98
Grouped elastic net	2.5	2.62	7.5	7.86
	Case 3			
Oracle	6	0	6	0
ω=0				
Lasso	6	1.78	6	8.86
Grouped lasso	6	1.92	6	17.76
Grouped elastic net	6	1.92	6	17.76
ω=0.5				
Lasso	6	1.74	6	8.92
Grouped lasso	6	1.97	6	17.91
Grouped elastic net	6	1.97	6	17.91

RESULTS OF SIMULATION STUDY 2

The results of Simulation 2 are displayed in Table 3, showing the bias, standard error (SE) and the root mean squared error (rMSE) of each coefficient by using the three different Tweedie methods, i.e., lasso, grouped lasso, grouped elastic net. Overall, the three different Tweedie methods have very similar performances in terms of bias, SE and rMSE. It is very interesting to observe that all the methods provide unbiased estimates of all the true covariate coefficients β except for the covariate vehicle body. Vehicle body is a categorical variable with 13 levels, and the coefficients of its dummy variables are much larger comparing to the other variables. The biases for these dummy variables are high along with relatively large SE and rMSE, while the biases for all the other covariates are negligible with small SE and rMSE.

DISCUSSION

In this final project, we aimed to reproduce the proposed models and the simulation studies in the article by Qian, Yang and Zou 2016 [1]. We have successfully developed our own R programs following the algorithms described in the article.

The design of the first simulation study follows the three simulation cases provided in the example 1 in the paper. For all the three cases, our results are similar to the results from the original paper in terms of the correctly identified active blocks, incorrectly identified active blocks, and the correctly identified coefficients. However, our simulation showed a higher number of the incorrectly identified coefficient. This is possibly due to the fact that only a small grid of m and τ were considered in our study, and small λ values were more likely to be selected by our cross-validation, therefore more coefficients than needed were incorrectly included by the models. We chose this setting to increase our computational speed within limited time, however the performances of the models have been compromised as we expected. Further simulation with a larger grid of the tuning parameter values is needed to confirm if this non-ideal performance persist using our algorithms.

The second simulation study is designed to evaluate the performance of the Tweedie models in analysis that reflects data structure in a real-life study. Based on the car insurance data, we found that the three Tweedie models provided unbiased estimates for the coefficients of almost all the covariates in the true model. It is indicated that the Tweedie model can be used to not only identify relevant predictors, but also estimate the magnitudes of the association of the covariates on the outcome. However, cautions need to be taken when interpreting the results of a categorical variables with many levels.

In summary, we have achieved our objective to implement a computational intensive algorithm and conducted a comprehensive simulation study proposed in our chosen article. We also believe that the second simulation proposed by us provides additional contribution and insight about the application of Tweedie models in real-life studies.

Table 3: Comparison of the three Tweedie Models on parameter estimation

	Bias			SE			rMSE		
	Lasso	Group Lasso	Group Elastic Net	Lasso	Group Lasso	Group Elastic Net	Lasso	Group Lasso	Group Elastic Net
Vehicle Value	-0.003	-0.004	-0.005	0.013	0.013	0.013	0.014	0.014	0.014
Vehicle Body									
CONVT	0.796	0.928	0.978	0.585	0.504	0.477	0.988	1.056	1.088
COUPE	0.704	0.679	0.706	0.311	0.286	0.268	0.77	0.737	0.755
HBACK	0.717	0.69	0.717	0.302	0.274	0.253	0.778	0.742	0.76
HDTOP	0.722	0.696	0.724	0.3	0.274	0.254	0.782	0.748	0.767
MCARA	0.636	0.621	0.644	0.319	0.307	0.293	0.711	0.693	0.707
MIBUS	0.721	0.692	0.72	0.312	0.283	0.262	0.785	0.748	0.766
PANVN	0.711	0.684	0.712	0.313	0.296	0.275	0.777	0.746	0.763
RDSTR	0.792	0.783	0.819	0.495	0.369	0.347	0.934	0.866	0.889
SEDAN	0.721	0.694	0.721	0.3	0.272	0.251	0.781	0.745	0.764
STNWG	0.723	0.698	0.725	0.3	0.272	0.252	0.783	0.749	0.768
TRUCK	0.728	0.7	0.727	0.303	0.276	0.256	0.789	0.752	0.771
UTE	0.716	0.69	0.717	0.305	0.275	0.255	0.779	0.743	0.761
Vehicle Age	-0.003	-0.004	-0.004	0.014	0.014	0.014	0.015	0.015	0.015
Gender	0.003	0.003	0.003	0.022	0.022	0.022	0.022	0.022	0.022
Area									
B	-0.007	-0.006	-0.006	0.029	0.029	0.029	0.03	0.03	0.03
C	-0.005	-0.004	-0.004	0.028	0.028	0.028	0.028	0.028	0.028
D	-0.003	-0.002	-0.002	0.037	0.037	0.037	0.037	0.037	0.037
E	-0.002	-0.001	-0.001	0.041	0.042	0.042	0.041	0.042	0.042
F	-0.003	-0.002	-0.002	0.051	0.05	0.05	0.051	0.05	0.05
Driver's Age									
2	0.009	0.007	0.007	0.038	0.037	0.037	0.039	0.038	0.038
3	0.005	0.004	0.004	0.04	0.039	0.038	0.041	0.039	0.039
4	0.005	0.004	0.004	0.04	0.039	0.038	0.04	0.039	0.039
5	0.01	0.008	0.009	0.041	0.04	0.04	0.042	0.041	0.041
6	0.006	0.005	0.005	0.051	0.049	0.049	0.051	0.05	0.049

REFERENCES

[1] Qian, W., Yang, Y. and Zou, H.(2016). "Tweedie's Compound Poisson Model with Grouped Elastic Net". *Journal of Computational and Graphical Statistics*,25:2, 606-625.

[2] Dunn, Peter K., and Maintainer Peter K. Dunn (2013). "Package 'tweedie'." *R package version 2.7.*

[3] Wolny-Dominiak, A, Trzesiok, M. (2015). "Package 'insuranceData'." *R package version 1.0.*