Advancing Insurance Intelligence: Integrated Statistical and Machine Learning Models in Loss Reserving and Auto Insurance Claim Prediction using Telematics Data

FARHA USMAN
BS (Hons) Mathematics & Statistics
Master (Hons) in Applied Statistics

Supervisor: A/Prof. Jennifer S. K. Chan

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

Faculty of Science
Department of Mathematics & Statistics
The University of Sydney

2024

Statement of integrity

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Abstract

This thesis delves into the forefront of insurance modeling, advancing methodologies for loss reserving and auto insurance claim prediction with telematics data. The chapters of this thesis encapsulate the transformative journey through advanced insurance modeling and achieve a paradigm shift in forecasting loss reserves and predicting auto insurance claims. These integrated approaches address the intricacies of modeling loss reserve data and navigate the complexities of telematics data, ensuring accurate risk assessment. The thesis unfolds across three research avenues: the Bayesian model for trapezoidal loss reserves, the lasso regularized model, and deep learning neural networks for claim counts with telematics data. Each approach is strategically chosen to enhance understanding and practices in non-life insurance risk management. After offering a comprehensive overview in Chapter 1, each chapter succinctly encapsulates pivotal discoveries from the preceding chapters and leads the continued discovery into a new chapter.

Chapter 2 focuses on the Bayesian loss reserve models to tackle the intricate challenge of modeling the loss reserve data in run-off triangles, employing an analysis rooted in the trapezoidal format. The complexity arises from the unknown dynamics in the claim/loss process. Popular loss reserve models utilize analysis of variance (ANOVA) and covariance (ANCOVA) models to describe the mean process through development year, accident year, and calendar year effects. The chapter proposes an enhancement to the mean function by incorporating persistence terms in the conditional autoregressive range model, addressing the persistence of claims across development years. Linear trends are adopted for accident and calendar year effects, while a quadratic trend is applied to the development year effect. Four error distributions, including generalized Beta type 2, generalized gamma, Weibull, and exponential extension, are considered to improve model flexibility. The proposed models are implemented using the Bayesian package `Stan` in the `R` environment. Results indicate that models with log-transformed mean functions and persistence terms offer better fits. The best model is then applied to forecast partial loss and calendar year reserves for three years.

Chapter 3 focuses on the telematics dataset analysis, advances claim prediction based on driving behavior, and introduces a robust two-stage regression framework. It an-

alyzes a large telematics dataset featuring 65 driving variables (DVs) and insurance claim counts for 14,157 drivers. The primary objective is to predict future claims based on driving behavior measured by DVs, identify crucial DVs distinguishing driving behaviors, classify drivers accordingly into safe and risky groups, and calculate premiums reflecting actual driving risk. Two-stage Poisson, Poisson mixture, and zero-inflated Poisson regression models with lasso regularization are employed to handle the noisy data with over 92 percent zero claims. Various lasso regularization techniques are considered, and the results are robustified through 100 repeats. DVs are selected from those frequently chosen DVs from regularized models over the repeats. Model performances are assessed using the Bayesian information criterion, mean square error, Pearson correlation, and area under the receiver operating characteristic curve. Finally, a usage-based experience-rating premium calculation method is proposed using the predicted claims and the classified driver groups.

Chapter 4 extends and explores the role of neural networks in insurance claim prediction using the Poisson mixture deep learning neural network. The model is designed to handle drivers' insurance claims with excessive zero occurrences by setting a higher prior probability of a safe (low claim) group. The chapter evaluates the PM network using the PM negative log-likelihood loss function instead of the common choice of mean square error loss function. This loss function captures the asymmetric distribution of claim counts. On the other hand, mean square error is adopted as a prediction accuracy measure in model selection. The meticulous search for network architecture, employing both manual and Bayesian optimization search techniques, further improves the prediction accuracy of Poisson mixture networks. The commitment of this chapter to pushing the boundaries of predictive analytics is evident throughout the modeling, architecture selection, and evaluation process, positioning the Poisson mixture deep learning neural network as a noteworthy advancement in insurance claim prediction.

Chapter 5 culminates and summarizes the importance of the three proposed statistical and machine learning techniques in elevating predictive capabilities within the insurance sector, particularly in loss reserves and claims prediction with telematics data. In addition, this chapter charts a course for future research endeavors. It advocates for potential refinements for loss reserves models by incorporating a linear function in later development years, especially pertinent for larger run-off triangles, a bilinear persistent effect for development years, and other persistent terms in financial time series models, enriching the model capacity and forecasting accuracy. For the analysis of claim counts with telematics data, it advocates a 3- or more group Poisson mixture model, presenting an untapped avenue for capturing more diverse driving behaviors to tailor-made individual premium that reflects real driving risk.

Moreover, the two lasso-regularized and deep learning models should be developed to address the more granular telematics data, particularly with continuous time, instead of aggregated cross-sectional monitoring of driving behaviors. These forward-looking stances present challenges and encourage continuous innovation and exploration in the dynamic landscape of insurance practices.

In summary, these three research avenues cement the thesis's position as a trailblazer in reshaping the landscape of insurance intelligence.

Acknowledgements

I express my deepest gratitude to my supervisor, Jennifer Chan, for her steadfast support, guidance, and mentorship throughout my PhD journey. Her expertise, patience, and encouragement played a pivotal role in shaping my research trajectory and fostering academic and personal growth.

I sincerely thank my supervisor's students, colleagues, and the Department of Mathematics and Statistics team. Their collaborative spirit, insightful discussions, and camaraderie enriched my academic experience, contributing significantly to the success of this research.

Special appreciation is reserved for the Disability Services team, whose invaluable support during my PhD journey, seminars, and tutorials created an inclusive academic environment. Their dedication allowed me to focus on my studies without unnecessary barriers, and I am truly grateful for this.

Heartfelt thanks go to my husband, whose unwavering encouragement, understanding, and belief in my abilities provided the emotional foundation necessary to navigate the challenges of doctoral research. To my beautiful sons, Fahd and Faruq, your laughter and joy illuminated the long study hours. Your patience during my busy periods and endless enthusiasm filled our home with positivity. This achievement is as much yours as it is mine.

This journey would not have been possible without the collective support and encouragement of these remarkable individuals and institutions. Thank you for being an integral part of my academic adventure.

In remembrance of my mother "Yasmeen," whose love and sacrifices made this PhD journey possible. This thesis is a testament to her enduring impact on my life and a tribute to the woman who will always hold a special place in my heart.

Contents

List of Figures

List of Tables

Nomenclature

List of Acronyms

A	Adaptive lasso
ACF	Autocorrelation function
ANOVA	Analysis of variance
ANCOVA	Analysis of covariance
AIC	Akaike information criterion
AUC	Area under curve
BIC	Bayesian information criterion
BO	Bayesian optimization
CARR	Conditional autoregressive range
CL	Chain ladder
CNN	Convolution neural network
CoV	Coverage
CV	Cross-validation
DIC	Deviance information criteria
DLNN	Deep learning neural network
DP	Dynamic prior probability
DVs	Driver behaviour variables
E	Elastic net
EE	Exponential extension distribution
GB2	Generalized Beta type 2 distribution
GG	Generalized gamma distribution
GLM	Generalized linear model
GP	Gaussian processes
GPS	Global Positioning System
IG	Information gain
L	Lasso
LSTM	Long short-term memory
MAN	Manual
MAE	Mean absolute error

MAPE	Mean absolute percentage error
MCMC	Markov chain Monte Carlo
MHYD	Manage-How-You-Drive
ML	Machine learning
MLP	Multi-layer perception
MSE	Mean squared error
N	Adaptive elastic-net
NB	Negative binomial
NN	Neural network
NLL	Negative likelihood function
PAYD	Pay-As-You-Drive
PCA	Principal component analysis
PHYD	Pay-How-You-Drive
PM	Poisson mixture
RMSE	Root mean squared error
RNN	Recurrent neural network
ROC	Receiver operating characteristic curve
SGD	Stochastic gradient descent
TP	Two-stage threshold Poisson
UBI	Usage-based auto insurance
ZIP	Zero-inflated Poisson

List of Symbols (Trapezoidal loss reserves data)

$y_{i,j}$	observable incremental losses in accident year i-th row and development year j-th column
T	all observable losses in the run-off triangle
i	accident year
j	development year
k	calendar year in the diagonal
t	current year
$\mathbf{y}_k^c$	vector of losses in calendar year k
$\mathbf{y}_t^p$	vector of non-observable losses in lower triangle
$\mathbf{y}_t^t$	vector of all non-observable losses outside run-off triangle
$\mu_{i,j}$	mean functions
α_i	accident year effect i
α_s	linear accident year effect
β_j	development year effect i
β_{ij}	accident-development year interaction effect
β_q	quadratic development year effect
δ_{i+j-1}	calendar year effect
δ_l	linear calendar year effect
γ_{1l}	weak persistence autoregressive effect
γ_{2l}	strong persistence autoregressive effect
f_{GB2}	density of GB2
f_{GG}	density of GG
f_{W}	density of Weibull
f_{EE}	density of Weibull
a	a shape parameter for GB2, GG, Weibull, EE
b	a scale parameter for GB2
p	a shape parameter for GB2, GG
q	a shape parameter for GB2
λ	a scale parameter for GG, Weibull, EE
F^{-1}	inverse CDF function
h	future years of losses
n_{eff}	number of effective sample size
$\widehat{R}$	scale reduction measure

List of Symbols (Telematics loss claims data)

N	number of drivers (rows)
J	number of driving variables (columns)
$\boldsymbol{x}_{\bullet j}$	data features in column vector j
$\boldsymbol{y}$	claim counts in column vector
$\boldsymbol{n}$	exposure/offset in column vector
$\boldsymbol{a}$	annual claims
b	index for driver claim classes: 0, 1 and 2^+
H_j	entropy of the j-th DV
ρ	correlation
G	number of driver groups
d	Euclidean distance
$\mathcal{C}_g$	cluster as proxy to true driver group g
$\mathcal{G}$	estimated driver group
$\boldsymbol{\beta}$	Poisson regression parameters
$\boldsymbol{\psi}$	logistic regression parameters in zero-inflated Poisson model
$\mathcal{L}$	data likelihood function
π	prior probability of PM model
$\mathfrak{z}_{ig}$	posterior probability for driver i in group g
λ	penalty parameter for $\boldsymbol{\beta}$ in sum of squared residual loss function
$\mathcal{D}$	deviance
$\mathfrak{m}$	performance measures in lasso regression
$\mathcal{R}_{\mathcal{M}}$	rank of model $\mathcal{M}$
w	weights for the weighted sum of ranks in selecting regularized models
ϱ	penalty function for $\boldsymbol{\beta}$ in lasso regression
$\mathfrak{w}$	adaptive weights in adaptive lasso regularization
τ	thresholds for 2-stage TP model
K	folds of CV
R	number of repeated samples
$\mathcal{S}_{rk}$	subsample for repeats, CV fold, etc
$\mathcal{I}$	index set of drivers for repeats, CV fold, models, threshold, etc
$\boldsymbol{I}$	selection frequency measures for DVs
P_{it}	premium for year t
$\mathsf{L}_{\mathsf{i,t-1}}$	historical claim/loss in year $t-1$
$\mathfrak{L}$	number of hidden layers
m_l	number of neurons in layer l
f_l	activation function in layer l
N_T	size of training data
N_V	size of validation data
p	proportion of data assigned to training
M	number of DLNN parameters

$\boldsymbol{W}^{(l)}$	weight parameters of DLNN in layer l
$\boldsymbol{b}^{(l)}$	bias parameters of DLNN in layer l
$\mathcal{E}$	number of epoch
$\mathcal{B}$	batch size
ζ	learning rates
$\mathfrak{p}$	dropout rate
$\mathcal{P}$	patience in early stopping
$\mathcal{J}_{\texttt{PMNLL}}$	loss function of DLNN using NLL of PM model, etc
$\mathfrak{I}$	number of batches in running one epoch

Glossary

AUC-ROC: Area Under the Receiver Operating Characteristic Curve - a performance metric for binary classification models. It represents a graphical plot that illustrates the model's ability to distinguish between the positive and negative classes across various threshold values.

Burn-in: Set of samples at the start of the MCMC run.

CNN: Convolutional neural network - a neural network designed for processing structured grid data, such as images. CNNs use convolutional layers to learn spatial hierarchies of features from the input data automatically and adaptively.

DIC: Deviance Information Criterion - a statistical measure used for model comparison and selection in Bayesian statistics.

`glmnet`**:** A package in `R` refers to the "Lasso and Elastic-Net Regularized Generalised Linear Models" - fitting generalized linear models with L1 (lasso) and L2 (ridge) regularization.

MCMC: Markov chain Monte Carlo - a stochastic algorithm for drawing samples from a posterior distribution to estimate the distribution.

Non-linear: Shows patterns, curves, or irregular behavior.

PCA: Principal component analysis - a dimensionality-reduction approach for reducing the dimensionality of large data sets by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

`Stan`**:** A probabilistic `R` programming language used for statistical modelling and Bayesian inference.

State-space model: This model describes the evolution of a system over time. It consists of two components: the state equation, which represents the unobservable system dynamics, and the observation equation, which links the unobservable states to the observed data.

Ridge regression: It performs L2 regularization and adds the "squared magnitude" of the coefficient as the penalty term to the loss function.

Authorship attribution statement

This thesis contains material as paragraph in Chapter 1 and Chapter 5, and Chapter 2 as paper is published as:

Usman, F., & Chan, J. (2022). New loss reserve models with persistence effects to forecast trapezoidal losses in run-off triangles. ASTIN Bulletin: The Journal of the IAA, 52(3), 877-920

I designed the study with the co-author Jennifer Chan, developed new methods, wrote a draft, and interpreted the analysis.

Attesting authorship attribution statement

As a supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Chapter 1

Introduction

1.1 Background and motivation

Insurance has become increasingly important in contemporary society as it provides a safety net in an increasingly complex and interconnected world, offering financial protection and stability in the face of diverse and evolving risks. These risks refer to the uncertainty in the future that may lead to severe consequences of unfortunate events such as death, accident, injury, etc, to individuals and companies. To mitigate risks, modern insurance becomes a functional financial innovation that allows market participants to identify, estimate, and relocate risks, thereby minimizing individual risks by pooling and diversification and safeguarding financial stability and operational continuity. Among the different types of insurance, this thesis excludes life insurance, which requires mortality modeling. To put the importance into perspective, the market size (gross written premium) of the non-life insurance market in Australia is projected to reach US$58.93 billion in 2023. The gross written premium is expected to show an annual growth rate of 3.61%, resulting in a market volume of US$70.36 billion by 2028.

As insurance companies operate as intermediaries to connect individual risks via insurance policies or contracts, the cash flows between the insurer and the insured are determined by the "premium" and "claim" of the policies. The first task of an

insurance company is to set premiums, which are the fees that the insured needs to pay to hedge different risks. The second task is loss reserving, which is the estimated amount of reserve to cover the claims from the settled policies. The loss reserving and premium setting are two important problems this thesis focuses on.

An insurance company promises through an insurance policy to provide financial protection to the insured if some defined events occur. However, in many cases, claims incurred in a particular year are often not reported/settled in that year but with a time delay of years or perhaps decades. These claims are often called incurred but not reported (IBNR) claims. For this reason, the company has to set aside IBNR reserves, or simply loss reserves, to cover liabilities for the outstanding claims due to accidents incurred but not yet reported or settled. Since loss reserves generally represent the largest source of financial uncertainty in an insurance company, accurate prediction of the level of loss reserves is vital for the profitability and solvency of the company and for avoiding bankruptcy. Underestimating the level of loss reserve may lead to inadequate capital allocation for potential large claims, while overestimation can tie up financial resources unnecessarily. If loss reserve can be estimated precisely, the companies' overall efficiency and profitability can be improved. Hence, loss reserving is a multifaceted process that protects the financial health of insurance companies and promotes fairness, competitiveness, and customer satisfaction within the industry.

Apart from loss reserving, an insurance company needs to set up premium payments for the insured as an obligation for their right to claim according to an insurance policy. A fair calculation of premiums are very important for insurers to balance the risk. This applies particularly to auto insurance when factors contributing to claims can be plenty. In principle, auto insurance premium pricing is considered most equitable and economically efficient if it reflects the insurance costs covered by each policy.

Traditional auto insurance premiums are based on *driver-related risk factors* such as age, gender, marital status, claim history, credit risk, and living district and *vehicle-related risk factors* such as vehicle year/make/model which represents the residual value of an insured vehicle. Although these factors indicate claim frequency and

claim size, they do not reflect true driving risk. They often lead to cross-subsidizing higher-risk drivers by lower-risk drivers to balance the claim cost. These premiums have been criticized for being inefficient and socially unfair because they do not punish aggressive driving, nor encourage prudent driving. Chassagnon et al. (1997) reported that the accident risk depends not only on demographic variables but also on driver behavior that reflects how drivers drive cautiously to reduce accident risk.

The next two subsections review the background of these two research venues: loss reserve models for run-off triangles and claim count models for telematics data.

1.1.1 Loss reserve models for run-off triangle data

Loss reserve is an important component of an insurance company's overall reserves and financial statements. Modeling loss reserve is one of the most challenging tasks for actuaries. To estimate loss reserves, insurance companies aggregate individual claim history and align the aggregated claims into a run-off triangle with rows denoting claims arising from certain policy years and columns denoting the lag years that claims settled. Section 2.2.1 provides details of run-off triangles. Pigeon et al. (2014) shows in Figure 1 of the paper the operation for the run-off triangle, explaining the lag between the occurrence of an insurable event and the filing of a claim by the policyholder. In the figure, a claim occurred at time t_1 and was declared to the insurer at time t_2. Factors contributing to IBNR cases include reporting lag, claims processing time, and long-tail claims for some insurance lines, such as liability or workers' compensation.

With uncertainties in the time lags inherently involved in the claims settlement process, the estimation of the loss reserves is complex and highly complicated. Over the years, algorithms or models have been developed for run-off triangle data. Most statistical loss models assume that historical patterns of claim development will continue into the future (Wüthrich and Merz, 2008). The algorithm behind these models is to apply the persistent patterns to forecast future losses. There are two types of loss reserving models: deterministic and stochastic models. The latter model accounts

for certain levels of unpredictability or randomness of the data. In insurance, the chain ladder (CL) algorithm is a commonly used technique for estimating the reserves needed to cover future claims payments. This oldest and most popular algorithm is a deterministic algorithm, meaning it does not explicitly incorporate uncertainty or random variation into its calculations. Instead, it uses development ratios and loss reserves factors on the cumulative claims (aggregated individual claims across development years) in the run-off triangle to estimate claim reserves (Renshaw, 1989; Mack, 1993).

The fact that the CL algorithm is distribution-free allows more applications with no explicit assumptions. However, it fails to provide standard errors as measures of uncertainty for the CL reserve estimates. Mack (1993) proposed a distribution-free formula to calculate the standard errors of CL reserve estimates. On stochastic extension, Renshaw and Verrall (1998) showed the Poisson model for incremental losses is a stochastic model that can reproduce the historical CL estimates. Mack and Venter (2000) further proposed the over-dispersed Poisson model as the stochastic model for CL estimates and argued against the Poisson model. Moreover, Kuang and Nielsen (2020) chose the log-normal CL model as the best model for loss reserving. Other literature on stochastic CL includes Adam (2018).

Although the CL algorithm was extended to incorporate stochastic errors, it is still considered inadequate in projecting future claims payments by simply extrapolating historical patterns of claim development. Different loss reserve models have been developed within the generalized linear model (GLMs) family to enhance the capability and flexibility of claim projection for run-off triangle data. Note also loss models developed for individual claim payments (Dong and Chan, 2013; Pigeon et al., 2014). Chapter 2 reviews these literature and proposes new loss models using the conditional auto-regressive range (CARR) models for financial time series to capture the persistence of claims across development years and a range of flexible distributions. Model performances are evaluated using simulations and model fit measures. The applicability of the proposed models is demonstrated using empirical run-off triangle data from an insurance company in Israel from 1978 to 1995.

1.1.2 Claim count models with telematics data

As research found that traditional demographic proxy factors such as driver age, gender, marital status, claim history, postcode, vehicle year/make/model, and credit risk etc are inadequate to reflect true driving risk, the installation of telematics utilizing smartphone sensors (Global Positioning System – GPS, accelerometer, gyroscope, etc.) or On-Board Diagnostic (OBD) devices in vehicles offers the possibility of evaluating true driver behavior. Granular information can be extracted from these devices and transformed to construct driver behavior variables (DVs) for each driver. Sagberg et al. (2015) reviewed state-of-the-art research on driving behaviors or styles in relation to road safety and concluded that these factors are crucial to road safety.

Usage-based auto insurance (UBI) represents an innovative evolution in automobile insurance premium pricing using these DVs from telematics-related data. Soleymanian et al. (2019) showed that individuals drove less and more safely with the UBI premium. UBI schemes can be categorized into three types (Ziakopoulos et al., 2022):

1- Pay-As-You-Drive (PAYD) charges premiums based on the drive-less-pay-less principle supplemented with *driving habit* and *travel information*, such as drivers' strategic choices of the road network, travel time, mileage, etc. The premium is calculated based on the number of miles driven based on the milometer readings (Arumugam and Bhargavi, 2019).

2. Pay-How-You-Drive (PHYD) extends PAYD based on travel behavior profiling to driving behavior profiling. Drivers' *driving behavior*, defined as drivers' operational choices in driving their vehicles, include speeding, harsh braking, hard acceleration, sharp cornering, and lane changing in a certain road type, traffic, and weather condition (Tselentis et al., 2016; Winlaw et al., 2019).

3. Manage-How-You-Drive (MHYD) extends PAYD to monitor further driving patterns in relation to fatigue, distraction, drowsiness, etc. The premium is calculated in the same way as PHYD and suggestions to the driver for ensuring safety.

Under the three UBI schemes, information is used to determine premium changes from demographic proxies for traditional auto insurance to driving distance for PAYD schemes and then to driving style for PHYD and MHYD schemes. PAYD is an improvement over traditional auto insurance. Not only actuarially justified, the other benefits of the PAYD scheme were discussed by Litman (2007). Based on a study of 450,000 insurance policies, Gusman (2009) (mentioned in P.18 of Litman (2011)) found "a significant difference in average claim costs between groups of higher and lower annual mileage".

However, Kantor and Stárek (2014) pointed out the weaknesses and shortcomings of the PAYD policies on counting only the driven kilometers and completely ignoring drivers' behavior. Many researchers have reported the significant impact of driving style data on modeling driving risk. Zantema et al. (2008) showed more than a 5% reduction in total crashes in the Netherlands by implementing the PHYD scheme. Elvik (2014) found that the PHYD scheme can reduce speeding and mileage by incentivizing drivers to consider car sharing or even public transport.

The benefit of the PHYD scheme stems from monitoring driving behavior to determine the next UBI premium, an advancement over the traditional experience-rating premium as it incorporates both historical claim experience and current driving risks. As PHYD policies reward good drivers with lower premiums, the number of PHYD policies has grown in recent years. QBE Australia (Q for Queensland Insurance, B for Bankers and Traders' and E for The Equitable Probate and General Insurance Company) released a product called Insurance Box, which is a PHYD policy with telematics installed in the vehicle. This product also provides risk scores as feedback to the insureds for their driving performances and the risk scores are used to update premiums regularly to feedback to drivers and encourage them to improve their driving skills through premium reduction.

The benefits of the PHYD scheme were confirmed by many research results. Toledo and Lotan (2006) found a significant connection between drivers' risk indices and historic crashes, suggesting that the risk indices are useful indicators of risks in car crashes. Bolderdijk et al. (2011) showed that speeding can be reduced by monitor-

ing the driving since drivers' awareness of being monitored can simulate behavioral change. Similar results were also obtained by Wouters and Bos (2000), who found that monitoring of driving resulted in accident reduction by 20%. Through the monitoring system, risky drivers are more alert in time. As their risky driving can be intervened earlier, more lives can be saved (Hurley et al., 2015). Lastly, Ellison et al. (2015) concluded that personalized feedback can induce significant changes in driving behavior. However, the largest reductions in risk are observed when drivers are also awarded a financial incentive to change. In summary, the PHYD policies promote personalized insurance, foster traffic safety, lessen traffic congestion, and reduce oil consumption and pollutant emission (Barry and Charpentier, 2020).

Considering the fact that most vehicle manufacturers will install telematics and OBD devices, UBI will be rapidly subscribed worldwide (Ptolemus Consulting Group, 2016). In response to this changing environment, actuarial practitioners have commented that "Focus on improved data management, Big Data, and improved analytics are critical success factors within the industry and will be a key differentiator through 2020". Despite the growing interest in UBI (Husnjak et al., 2015), there is still limited research on machine learning predictive models required for UBI schemes (e.g., Paefgen et al. 2014; Makov and Weiss 2016). A great barrier to development is the inability to analyze the big telematics data. Another issue is the access to telematics data. Fortunately, Cambridge Mobile Telematics (CMT; `https://www.cmtelematics.com`), the world's largest telematics service provider, has made the access possible. Founded in 2010 by two MIT professors, Sam Madden, and Hari Balakrishnan, along with entrepreneur Bill Powers, CMT is known for its DriveWell platform, which has pioneered signal processing and artificial intelligence algorithms that specialize in providing mobile telematics and analytics solutions using mobile sensing, internet of things (IoT) devices, and behavioral science to measure driving behavior. The platform is utilized by various stakeholders, including automobile insurance companies implementing UBI policies, cities conducting public safety campaigns offering incentives based on telematics data, and numerous standalone applications focused on safe driving and crash detection. Reagan et al. (2023) and Pinals et al. (2022) have worked with CMT to

identify risky driving and estimate the prevalence of driver handheld cellphone use, respectively.

Recent research has underscored the significance of machine learning in analyzing telematic driving data. Its ability to uncover patterns, make predictions, and derive insights from large and complex datasets is highly valuable. Chapters 3 and 4 aim to contribute to this field by extending machine learning algorithms within GLMs to analyze the big telematics data. Chapter 3 reviews the literature on developing statistical models with machine learning techniques to make sense of the telematics data. It proposes lasso regularization and extensions within GLMs with Poisson and Poisson mixture models. The application of lasso regularization to mitigate overfitting in the context of telematics data analysis is novel in this field. Chapter 4 extends the research on telematics data with the deep learning neural networks (DLNNs) to predict claims with Poisson mixture loss function. Both types of models are applied to a telematics dataset with 65 DVs. More elaboration on lasso regularization and DLNN models is provided in the next section on the objectives of this thesis.

1.2 Objectives and contributions

It is clear that loss reserving and claim prediction for UBI premiums, within the realm of non-life insurance, are two important areas in actuarial research. The motivation for propelling this thesis is to advance insurance intelligence by deriving predictive models using advanced statistical and machine learning techniques, focusing, in particular, on extreme losses/claims. The overall aims encompass choices of modeling techniques to understand the contributing factors to extreme events and differentiate between extreme and rare catastrophic events. The thesis strategically unfolds challenges across these two distinct research avenues, each delineating a unique approach.

1. **Loss reserve models for run-off triangle data**: This avenue delves into the intricate realm of loss reserves in Chapter 2, aiming to augment the sophistication of loss models by introducing the CARR models to capture the persistence

of claims across development years. Moreover, a diverse array of error distributions is also introduced, including the generalized Beta type two distribution with the thick tail to capture extreme losses. The chosen methodologies revolve around the Bayesian approach, leveraging its flexibility and adaptability to encapsulate the nuances inherent in the data.

2. **Claim count models for telematics data**: The focus then shifts to the domain of claim counts with telematics data, where a methodical exploration is conducted through the lens of lasso regularized regression models in Chapter 3 and DLNN models in Chapter 4. Both approaches contribute to a refined and optimized understanding of the relationships between variables collected from telematics and annual claims.

   This research seeks to answer four research questions:

   1. What DVs and effect patterns are predictive for UBI claims?
   2. How to ensure the stability of these variables in prediction?
   3. How to classify drivers using these variables?
   4. How to tailor individual premium rates that reflect driving skills?

   To select important DVs and ensure stability in prediction, the first approach introduces lasso regularization. Besides, cross-validation and resampling are also conducted to avoid overfitting. To classify safe and risky drivers, the first approach proposes lasso regularized two-stage threshold Poisson (TP), Poisson mixture (PM), and zero-inflated Poisson (ZIP) regression models and extends the regularization to adaptive lasso and elastic net to identify separate DV sets for the two distinct safe and risky driver groups. Hence, extreme claim counts relative to normal counts from the risky group can be captured in separate components of the mixture model.

   The second approach explores and culminates cutting-edge DLNNs for claim count. This approach signifies a paradigm shift from the popular mean squared error (MSE) loss function to the negative loglikelihood (NLL) loss function, embracing the capabilities of DLNNs to model the asymmetric and multi-

modal claim distributions such as the Poisson and Poisson mixture distributions. Moreover, instead of dropping or shrinking less important DVs in the linear mean function of lasso regression, DLNNs unravel nonlinear and complex patterns in the mean function. These extended DLNNs are particularly apt for capturing the intricate dynamics of driving behavior revealed from DVs on claim counts.

These proposed models will enhance the predictive power of future claims and, at the same time, be more parsimonious in the modeling context familiar to actuaries. Lastly, a new UBI experience-rating premium method is proposed, tailor-made to capture individual driving styles.

This endeavor is geared towards fostering a profound comprehension of actuarial risks and enhancing the landscape of risk management practices. Each of these research avenues is approached with a meticulous and tailored strategy, acknowledging and addressing the distinct characteristics of the data under scrutiny. The amalgamation of the CARR model within loss reserve models, the adoption of lasso regularisation, and the application of deep learning neural network (DLNN) techniques contribute to a comprehensive and multifaceted exploration poised to significantly advance the understanding and modeling of non-life insurance risks. Specific contributions are also described in Chapters 2 to 4 with more details.

1.3 Organization of the thesis

This thesis comprises five chapters. Chapter 2 reviews the development of the loss reserve model for run-off triangle data and proposes a new loss reserve model using the CARR model for the mean and a range of distributions for the multiplicative errors. Chapter 3 adopts lasso regularized Poisson and Poisson mixture regression to investigate the relationship between driving behaviors and claim frequencies. Chapter 4 extends the research to DLNNs. Chapter 5 concludes the research findings in this thesis and proposes some interesting future research.

Chapter 2

New loss reserve models with persistence effects to forecast trapezoidal losses in run-off triangles

This chapter is based on:

Usman, F., & Chan, J. (2022). New loss reserve models with persistence effects to forecast trapezoidal losses in run-off triangles. ASTIN Bulletin: The Journal of the IAA, 52(3), 877-920.

The journey of advancing insurance intelligence begins with loss reserve models. Section 2.1 reviews the development of loss reserve models. Section 2.2 introduces the run-off triangle data extended to the trapezoidal format for fitting the proposed loss reserve models. Section 2.3 describes the proposed loss reserve model according to the mean functions and error distributions. Section 2.4 explains the implementation used with the Bayesian MCMC approach to estimate model parameters, monitor MCMC convergence, and perform model selection. Section 2.5 conducts an empirical study and assesses parameter estimate accuracy and CARR model robustness

through simulation experiments. Section 2.6.3 provides partial and calendar year reserve forecasts, compares forecast performance, and studies the impacts of over and under-loss reserve forecasts. Finally, Section 2.7 concludes the results.

2.1 Introduction

Over the years, GLMs have been a research focus for stochastic loss reserving models using the incremental loss run-off triangle (Taylor and McGuire, 2016). GLMs are often formulated in terms of mean functions and error distributions. For the mean function, Chan et al. (2008), and Dong and Chan (2013) adopted the analysis of variance (ANOVA), analysis of covariance (ANCOVA) and state-space models for modeling and forecasting losses. As an illustration, the ANOVA, ANCOVA, and state-space models for the incremental losses $y_{i,j},\ i,j = 1,\ldots,t$ made in accident year i and settled in development year j in the run-off triangle are given by

$$\begin{aligned}
\text{ANOVA model:}\quad \mu_{i,j} &= \mu_0 + \alpha_i + \beta_j, && (2.1)\\
\text{ANCOVA model:}\quad \mu_{i,j} &= \mu_0 + \alpha \cdot i + \beta_j, && (2.2)\\
\text{State-space model:}\quad \mu_{i,j} &= \mu_0 + \alpha_i + \beta_{ij}, && (2.3)\\
\alpha_i &= \alpha_{i-1} + e_i,\ e_i \sim \mathrm{N}(0,\sigma_e^2), && (2.4)\\
\beta_{ij} &= \beta_{i,j-1} + \varepsilon_i,\ \varepsilon_i \sim \mathrm{N}(0,\sigma_\varepsilon^2) && (2.5)
\end{aligned}$$

respectively, where $\mu_{i,j} = \mathrm{E}\big(\log(y_{i,j})\big)$, $y_{i,j}$ may follow some distributions such as lognormal, α_i denotes the effect from accident year i, β_j denotes the effect from development year j, and $\alpha \cdot i$ is the linear accident-year effect. Parameter β_j can capture some development year effect patterns, which often display a rapid increase to a peak, a milder decrease, and a slow level off. It is worth noting that the expression "year" used here to indicate the frequency of the data could be quarterly, monthly, and so on.

The state-space models have observation equations (2.3) and state equations (2.4)

and (2.5) for state variables α_i and β_{ij} which follow certain dynamic evolution with stochastic errors across accident years and development years respectively. The state variables β_{ij} allow the development year pattern to change with accident years. Hence, they capture the accident-development year interaction effects but with fewer parameters. For example, the parameters in (2.5) are $\beta_{i1}, i = 1, \ldots, t$ and σ_ε^2 instead of $\beta_{ij}, i, j = 1, \ldots, t$. Additional references for state-space models include De Jong and Zehnwirth (1983) and Costa and Pizzinga (2020), who used the Kalman filter estimator.

An alternative model to allow accident-development year interaction includes the Hoerl curve model of England and Verrall (2001) with the mean function

$$\mu_{i,j} = \mu_0 + \alpha_i + \beta_i \log j + \gamma_i j \tag{2.6}$$

and the distinct interaction pattern $\beta_{ij} = \beta_i \log j + \gamma_i j$. One common cause of the change in the development year pattern is the change in the rate of processing claims across accident years. See, for example, the comment on page 34 of Meyers (2015) that "claims are reported and settled faster today due to technology". He introduced the changing settlement rate (CSR) model with the mean function

$$\mu_{i,j} = \mu_0 + \alpha_i + \beta_j (1 - \omega)^{i-1}, \tag{2.7}$$

where the interaction pattern $\beta_{ij} = \beta_j (1 - \omega)^{i-1}$ and ω follows a uniform distribution with support $(-0.5, 0.5)$, that is, $\omega \sim \mathrm{U}(-0.5, 0.5)$ in a Bayesian approach. A positive ω causes the mean to decrease with i for early j in which $\beta_j > 0$ indicates a slowdown in claim settlement, whereas a negative ω will speed up the claim settlement.

Since high and low-level claims tend to cluster across development years, a new way to describe the progression of losses is to model the persistence across development years. Kremer (1984) introduced an autoregressive model to describe the short-term persistence in the development year pattern. This research proposes to adopt an order (p, q) conditional autoregressive range (called CARR(p, q)) model proposed by Chou et al. (2015) for financial volatility to capture partially the accident-development year interaction pattern due to development year persistence. Together with the accident

and development year effects, the mean function is given by

$$\mu_{i,j} = \mu_0 + \alpha_i + \beta_j + \sum_{l=1}^{p} \gamma_{1l}\, y_{i,j-l} + \sum_{l=1}^{q} \gamma_{2l}\, \mu_{i,j-l} \tag{2.8}$$

where γ_{1l} are weak persistence autoregressive terms, γ_{2l} are strong persistence terms, and the new interaction structure is $\beta_{ij} = \beta_j + \sum_{l=1}^{p} \gamma_{1l}\, y_{i,j-l} + \sum_{l=1}^{q} \gamma_{2l}\, \mu_{i,j-l}$. There is no literature on loss reserving to adopt the CARR model to model the persistence across development years. It is worth noting that the CARR model shares the same structure as the autoregressive condition duration (ACD) model, and the properties of the ACD model can be found in Fernandes and Grammig (2006).

While the above models considered only accident and development year effects, Björkwall et al. (2011) added the linear calendar effect δ_k, where δ is the calendar year effect parameter and $k = i + j - 1$ indicates calendar year. They proposed curve functions to smooth the accident, development, and calendar year parameters. To account for the trends in the accident, development, and calendar year effects, Zhang et al. (2018) adopted the probabilistic trend family (PTF) model

$$\mu_{i,j} = \alpha_i + \sum_{j'=1}^{j-1} \beta_{j'} + \sum_{k'=2}^{i+j-1} \delta_{k'} \tag{2.9}$$

where β_j and δ_k denote the incremental development and calendar year effects at year j and k respectively. While the interpretation of parameters is different, the idea in (2.9) is similar to (2.1) when modeling the development year and calendar year effects. The calendar year effect was modeled using the state-space model (De Jong, 2006).

Although individual losses are directly observed, the interest in loss reserving is to predict total loss. Hence, previous models considered aggregated losses in the run-off triangle. Nevertheless, exploring whether one should model individual or aggregated losses is still interesting. Estimating loss reserve using aggregated data reduces the impact of potential outliers, particularly the low-value outliers discussed in Chan et al. (2008). However, in the aggregation process, the heterogeneity of data information is lost. Hence, some studies focused on individual losses (Antonio and Plat, 2014). Cummins et al. (2007) considered subgroups of claims by accident and development years and applied different generalized beta type 2 (GB2) distributions with a constant

mean to each cell of the run-off triangle. This model allows greater flexibility than fitting a single severity distribution across cells. However, as a unique model is applied to each cell, the model can not capture the trend movement of losses across accident and development years. Hence, aggregated incremental losses in the run-off triangle are considered here to allow the modeling of trends across accident and development years.

Another consideration for GLMs is the choice of error distributions. De Jong et al. (2008) considered the gamma regression model. Venter (2007) extended the choice beyond the exponential family to capture a wide range of heterogeneity and tail behavior for the losses. These sophisticated distributions include the log-generalised-t (log-GT) (Chan et al., 2008), Pareto (Embrechts et al., 2013), Stable family (Paulson and Faris, 1985), Pearson family (Aiuppa, 1988), log-gamma and lognormal (Ramlau-Hansen, 1988), Burr 12 (Cummins et al., 1999), GB2 (Cummins et al., 1990; Dong and Chan, 2013) and exponential extension (EE) (Nadarajah and Haghighi, 2011) distributions. Although the log-GT distribution is flexible and nests the log-exponential-power, log-Student-t, lognormal, and log-uniform distributions as special cases, this distribution, as well as other log-transformed distributions, considers log-transformed losses, subsequently contaminating the model with negative skewness arising from taking the log of near-zero losses in the tail of the development year trend (Chan et al., 2008). This problem applies to the models in (2.1) to (2.9) with mean functions $\mu_{i,j} = \mathrm{E}(\log(y_{i,j}))$ for the log-transformed losses. Alternative distributions one may consider include GB2 and EE, which are defined on the positive support without the need to log-transforming the losses.

For more sophisticated distributions, Dong and Chan (2013) proposed a mixture of GB2 distributions to model the majority of small losses and a small portion of large losses. Besides, Bakar et al. (2015) proposed the composite distribution as an alternative approach to adopting different distributions for the small and extreme losses by piecing together two weighted distributions at a specified threshold with continuity and differentiability. They proposed seven distributions with Weibull as the head distribution and Burr, log-logistic, paralogistic, their inverse, and generalized Pareto as

the tails. As an extension to Dong and Chan (2013), Chan et al. (2018) proposed the contaminated GB2 mixture distribution family. They demonstrated the superiority of the distribution family over the folded normal and t distributions and the composite lognormal-Pareto and Weibull-Pareto distributions. However, these sophisticated distributions must fit with more data and are more suitable for individual loss data.

Popular approaches to estimate the loss reserve models include the maximum likelihood (ML) and Bayesian. The use of the Bayesian approach dates back to Verrall (2000), who forecasted outstanding losses using the Bayesian models. Over the years, the Bayesian approach has become popular in loss reserving (Meyers, 2015). The benefit of the Bayesian approach is the ability to add prior information to the model and provide predictive distributions for the required reserves via Markov chain Monte Carlo (MCMC) simulation. Ntzoufras and Dellaportas (2002) demonstrated the computational flexibility of the Bayesian approach to forecast loss quantiles using complex models, such as the state space and threshold models. On the other hand, the ML approach to forecast loss quantiles with flexible but complicated distributions can be tedious.

The contribution of this chapter is four-fold:

1. **Integration of CARR model**: Adopting the CARR model into the mean function of the loss reserve distribution provides a novel approach to capture the persistence of the development year effect. While originally developed for modeling return volatility in finance, this innovative application to loss reserving adds a new dimension to its utility. This chapter uses ANOVA and ANCOVA models to model accident, development, and calendar year effects.

2. **Adoption of diverse data distributions and Bayesian approach:** Four data distributions—GB2, generalized gamma (GG), Weibull, and EE—are considered. The Bayesian MCMC approach, implemented through the `Stan` package in the `R` environment, is employed for the proposed models. Ways to specify the log densities of these distributions in `Stan` are provided, and procedures to simulate posterior predictive distributions for loss forecasts are also illustrated.

These distributions offer mean and quantile estimates for loss forecasts.

3. **Application to run-off triangle data:** The applicability of the proposed models and estimation techniques are demonstrated using real run-off triangle data. Extensive model selection uses deviance information criterion (DIC) and mean squared error (MSE). Selected models are compared with the CL method and the CSR model. The chapter reports partial and three-year calendar year reserve forecasts using the best model and the chain ladder method.

4. **Validation through simulation experiments:** Two simulation experiments are conducted to validate the accuracy of parameter estimates and forecasts. The results affirm the robustness of the CARR model for in-sample fit and calendar year reserve forecasts in data without a persistence effect. The amount of over and under-reserve for three calendar year reserve forecasts are also provided. Results confirm the superior performance of the mean estimate for the best model compared to its quantile estimate counterpart and the CL estimates.

These new paradigms will facilitate new forecasting procedures for the actuaries.

2.2 Loss reserve data

2.2.1 Data structure for loss reserve

The run-off triangle aligns the claim/loss history of claim/loss size or frequency for a certain type of insurance across accident and development years. The triangular array in yellow color in Table 2.1 displays this structure. The entry $y_{i,j}$ are incremental losses in accident year i and development year j where $i, j = 1, 2, \ldots, t$ where t represents the current year. Let the set of all observable $T = t(t+1)/2$ losses in the run-off triangle in calendar year $k = i + j - 1 \leq t$ be the vector

$$\mathbf{y}_{1:T} = \{\mathrm{y}_n = y_{i,j} : (i,j) \in \mathcal{I}_{1:t},\ n = \sum_{r=1}^{i-1}(t - r + 1) + j\} \tag{2.10}$$

that appends rows of the run-off triangle into a vector where $\mathrm{y}_n, n = 1, \ldots, T$ denotes each loss in $\mathbf{y}_{1:T}$ that corresponds to certain $y_{i,j}$ in the run-off triangle and the index sets of losses in calendar year k and in calendar years 1 to t are given respectively by

$$\mathcal{I}_k = \{(i,j) : i + j = k + 1\} \text{ and } \mathcal{I}_{1:t} = \bigcup_{k=1}^{t} \mathcal{I}_k. \tag{2.11}$$

Accident Year i	**Development Year** j										
	1	2	3	.	.	.	$t-1$	t	$t+1$	$t+2$	$t+3$
1	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$	.	.	.	$y_{1,t-1}$	$y_{1,t}$	$y_{1,t+1}$	$y_{1,t+2}$	$y_{1,t+3}$
2	$y_{2,1}$	$y_{2,2}$	$y_{2,3}$	.	.	.	$y_{2,t-1}$	$y_{2,t}$	$y_{2,t+1}$	$y_{2,t+2}$	
3	$y_{3,1}$	$y_{3,2}$	$y_{3,3}$	.	.	.	$y_{3,t-1}$	$y_{3,t}$	$y_{3,t+1}$		
...	.	.	.	.	.	.	.	.			
...	.	.	.	.	.	.	.				
...	.	.	.	.	.	.					
$t-1$	$y_{t-1,1}$	$y_{t-1,2}$	$y_{t-1,3}$	.	.						
t	$y_{t,1}$	$y_{t,2}$	$y_{t,3}$	.							
$t+1$	$y_{t+1,1}$	$y_{t+1,2}$	$y_{t+1,3}$								
$t+2$	$y_{t+2,1}$	$y_{t+2,2}$									
$t+3$	$y_{t+3,1}$										

Table 2.1 – Layout of incremental loss data with the run-off triangle highlighted in yellow. Unobserved losses within t accident and development years are in blue or otherwise in green. Losses for the future three calendar years are in deep blue and mid to deep green in the *trapezoid*.

Let $\mathbf{y}_k^c = \{y_{i,j} : (i,j) \in \mathcal{I}_k\} = (y_{1,k}, y_{2,k-1}, \ldots, y_{k,1})$ be the vector of losses in calendar year k. Then the sum of losses for calendar year k is the sum of the k-th diagonal entries

$$s_k = \sum_{r=1}^{k} y_{r,k-r+1}, \quad k = t+1, t+2, \ldots. \tag{2.12}$$

Let $\mathbf{y}_t^p = \{y_{i,j} : i, j \leq t, i + j > t + 1\}$ be the vector of non-observable losses in the lower triangle (blue cells of Table 2.1) and $\mathbf{y}_t^t = \mathbf{y}_t^p \cup \{y_{i,j} : i > t \text{ or } j > t \text{ or } i, j > t\}$ be the vector of all non-observable losses (blue and green cells in Table 2.1) outside the run-off triangle. Costa and Pizzinga (2020) called the term "tail effects" for losses $y_{i,j}, i > t$ or $j > t$ or both (light, mid and dark green cells). Most loss reserve models focus on predicting losses in $\mathbf{y}_t^p$. Since these losses are within the rectangle, they are easier to predict using observable losses $y_{i,j}, i + j \leq t + 1$. However, those tail effect losses are harder to predict because the observable losses do not provide information

for future accident, development, and calendar year effects when $i, j > t$. Costa et al. (2016) and Costa and Pizzinga (2020) defined three types of reserves:

1. *Partial reserve* refers to the sum of losses in $\mathbf{y}_t^p$ (light and dark blue cells in Table 2.1) in the lower triangle (Renshaw and Verrall, 1998; Chan et al., 2008; Costa et al., 2016).
2. *Total reserve* represents the sum of losses in $\mathbf{y}_t^t$ (all blue and green cells) outside the run-off triangle (Atherino et al., 2010; Costa et al., 2016).
3. *Calendar year reserve* corresponds to the sum of losses in $\mathbf{y}_k^d$ for calendar year $k > t$ (each lower diagonal of the run-off triangle). These reserves are crucial for managing short-term financial decisions, as they are the reserves for payments expected to occur soon. Costa et al. (2016) considered one step ahead calendar year reserve forecast.

This study forecasts $h = 3$ calendar year reserve s_k, $k = t+1, t+2, t+3$ and provides the popular partial reserve forecast. The calendar year reserve forecasts rely on losses identified in dark green ($i, j = 1$) and mid-green in Table 2.1 for three future calendar years. Models that capture the trend effects through the ANCOVA and CARR models for development year effect facilitate the forecast of losses beyond year t. For the ANOVA models, further assumptions are introduced to facilitate the forecast, with detailed explanations in Section 2.3.1. The $h = 3$ diagonals of loss forecasts adopt the shape of a trapezoid and are referred to as trapezoidal losses.

The data structure layout presented in 2.1 is further illustrated in Figure 2.1 for comprehensive data analysis. Prior examinations by Chan et al. (2008) and Chan (2015) have contributed to this analysis. The dataset encompasses payments made to the insured by an Israeli insurance company during the accident years 1978 to 1995 and the development years 1 to 18, spanning 18 accident, development, and calendar years ($t = 18$). The upper triangle, comprising $T = 171$ observations, is highlighted in grey. The blue trapezoid with red dashed lines also emphasizes the three calendar year reserves for 1996 to 1998.

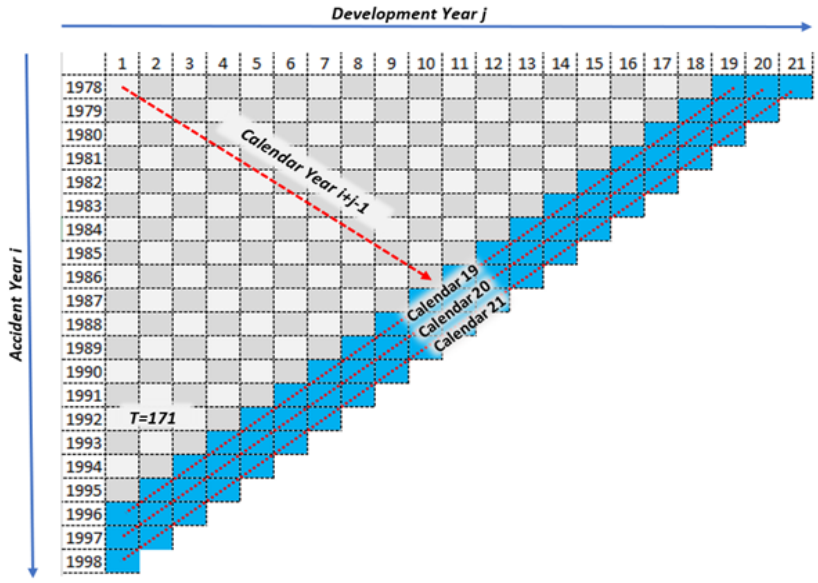
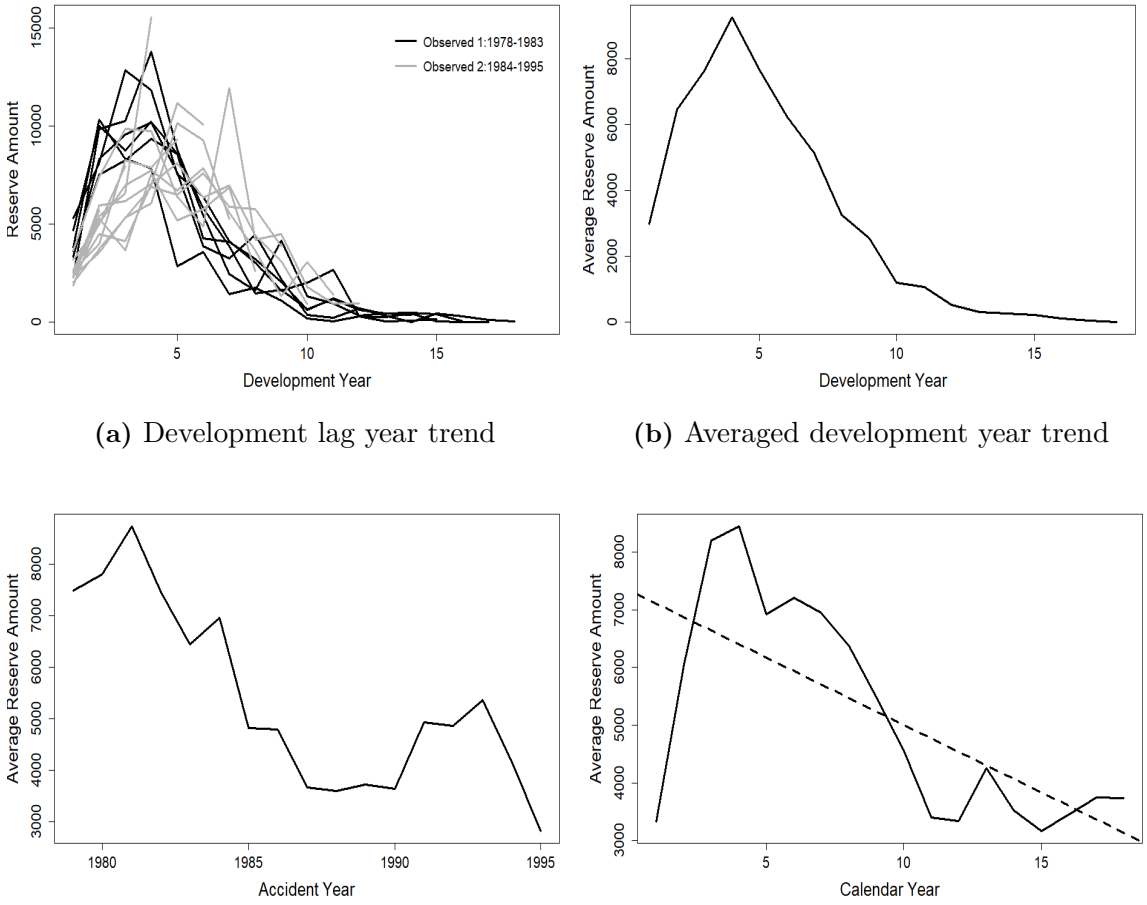


Figure 2.1 – Data structure with observed losses in the upper triangle in grey and forecast losses in the *trapezoid* in blue. The red dashed line indicates the three future calendar years.

2.2.2 Data features

Figure 2.2 visualizes the trend of effects across accident year, development year, and calendar year. In Figure 2.2a, the development year effect exhibits two distinct quadratic trends. The trend for 1978 to 1983 (depicted in black) peaks in the fourth development year and gradually decreases. On the other hand, the trend for 1984 to 1995 (shown in grey) peaks in the sixth development year and has a more gentle decline. Figure 2.2b displays the average of these development year trends, revealing a sharp peak in the fourth development year. To display the trend of the accident year effect, it is important to note that averaging over development years can be misleading as later accident years do not have lower losses coming from later development years. As a compromise, the trend of accident year losses is plotted in Figure 2.2c and averaged over the first three development years. Notably, for 1994 and 1995, averages are based on the first two losses and the first loss, respectively. The observed trend shows a general decrease with some noise. Finally, Figure 2.2d presents the calendar year effect by plotting the average of diagonal entities across cal-

endar years. Despite some noise in the first two values, a downward trend is evident, suggesting a linear effect.



(a) Development lag year trend

(b) Averaged development year trend

(c) Averaged accident year trend (first 3)

(d) Averaged calendar year trend

Figure 2.2 – Trends of development, accident, and calendar year effects in the run-off triangle data

2.3 Loss reserve techniques

2.3.1 Proposed statistical models

Mean functions

To model the incremental loss $y_{i,j}$, various error distributions $\mathcal{D}(\boldsymbol{\theta})$ and mean functions $\mu_{i,j}$ are employed in the following formulation:

$$y_{i,j} \quad = \quad \mu_{i,j}\epsilon_{i,j}, \quad \epsilon_{i,j} \sim \mathcal{D}(1, \boldsymbol{\nu}), \ \mathrm{E}(\epsilon_{i,j}) = 1 \tag{2.13}$$

where $y_{i,j} \sim \mathcal{D}(\mu_{i,j}, \boldsymbol{\nu})$ denotes any distribution defined on the positive real support $\mathbb{R}^+$ such as GB2, GG, Weibull, and EE (see the upcoming **Data distributions** section), $\mathrm{E}(y_{i,j}) = \mu_{i,j} = g(\eta_{i,j})$, $\eta_{i,j}$ is a linear function of predictors and $\boldsymbol{\nu}$ is a vector of shape parameters for the distribution. The mean function $\mu_{i,j}$ is defined in three aspects:

1. the type of parameters in $\eta_{i,j}$;
2. the functional form of the predictors, for example, in the ANOVA and ANCOVA models;
3. the link function $g(\eta_{i,j})$ that links $\eta_{i,j}$ to $\mu_{i,j}$.

For the first aspect, the linear function $\eta_{i,j}$ contains four types of parameters:

1. α_i for the accident year effect,
2. β_j for the development year effect,
3. δ_{i+j-1} for the calendar year effect, and
4. γ_{1l} for the autoregressive effect and γ_{2l} for the persistent effect across development years using the CARR model.

The model with ANOVA and CARR(p, q) mean function is named the ANOVA-CARR(p, q) model. It is defined as:

$$\mu_{i,j}|\boldsymbol{y}_{i,1:(j-1)}, \boldsymbol{\mu}_{i,1:(j-1)} = \eta_{i,j} = \mu_0 + \alpha_i + \beta_j + \delta_{i+j-1} + \sum_{l=1}^{p} \gamma_{1l}\, y_{i,j-l} + \sum_{l=1}^{q} \gamma_{2l}\, \mu_{i,j-l} \quad (2.14)$$

where the parameter constraints are $\sum_{i=1}^{t} \alpha_i = \sum_{j=1}^{t} \beta_j = \sum_{k=1}^{t} \delta_k = 0$. The model further assumes that $y_{i,0} = r_0 y_{i,1}$ and $\mu_{i,0} = 0$ where the initial development factor $r_0 \simeq 0.7$ is estimated by the average of $y_{i,1}/y_{i,2},\ i = 1, \ldots, 17$. This study considers only order $(p, q) = (1, 1)$ as higher-order effects are often insignificant, and they also need longer development year data to estimate.

For the second aspect, there are three models to specify the predictor effects: the ANOVA, ANCOVA, and mixed ANOVA/ANCOVA models. Predictors in the ANCOVA model can describe the trend patterns using the general additive model $\theta_u = \sum_{l=1}^{g} w_l f_l(u)$ where $\theta_u = \alpha_i, \beta_j, \delta_{i+j-1}$, w_l is the weight for function $f_l(\cdot)$ and g is the number of $f_l(\cdot)$ to describe the trend pattern (Verrall, 1996). Although the general additive model can be very general, the trend patterns observed in Figure 2.2 are not sophisticated but noisy. Hence, the parsimonious ANCOVA-type predictors are proposed as below:

1. *linear* accident year effect $\alpha_i = \alpha_s = \alpha\, y_{i1}$ proportional to the first development year loss y_{i1} (denoted by α_s in Table 2.2),
2. *quadratic* development year effect $\beta_j := \beta_q = \beta_1\, j + \beta_2\, j^2$ (denoted by β_q), and
3. *linear* calendar year effect $\delta_{i+j-1} = \delta\,(i + j - 1)$ (denoted by δ_l) (Verrall, 1996).

These terms can replace the corresponding ANOVA-type predictors in (2.14).

For the third aspect, the mean function $\mu_{i,j} = g(\eta_{i,j})$ is linked to a linear function of predictors $\eta_{i,j}$ through the link function $g(\cdot)$. Two types of link function are considered as follows:

1. The identity link $\mu_{i,j} = g(\eta_{i,j}) = \eta_{i,j}$ in which all location parameters μ_0,α_i,β_j,δ_k, γ_{1l},$\gamma_{2l} > 0$, for $i, j, k < t$ and μ_0 is adjusted to ensure a positive mean $\mu_{i,j} > 0$ for the distributions $\mathcal{D}(\mu_{i,j}, \boldsymbol{\nu})$ on the positive support, and

2. The log link $\log(\mu_{i,j}) = \eta_{i,j}$ or $\mu_{i,j} = \exp(\eta_{i,j})$ to describe the non-linear effects of predictors on the mean function. Accordingly, the model in (2.14) becomes

$$\begin{aligned}\mu_{i,j}|\boldsymbol{y}_{i,1:(j-1)}, \boldsymbol{\mu}_{i,1:(j-1)} &= \exp(\eta_{i,j}) \\ &= \exp\left[\mu_0 + \alpha_i + \beta_j + \delta_{i+j-1} + \sum_{l=1}^{p} \gamma_{1l} \log(y_{i,j-l}) + \sum_{l=1}^{q} \gamma_{2l} \log(\mu_{i,j-l})\right] \quad (2.15)\end{aligned}$$

where $\alpha_i, \beta_j, \delta_k$ have no positive constraints.

The models with mean functions (2.14) and (2.15) are used to estimate in-sample losses (yellow cells in Table 2.1) and forecast out-of-sample losses (blue cells in Table 2.1). However, for losses outside the observation period with $i, j > t$ (green cells in Table 2.1), the ANOVA terms $\alpha_i, \beta_j, \delta_k$, $i, j, k > t$ in the mean function $\mu_{i,j}$ are unavailable. This study proposes to estimate $\hat{\alpha}_i = \hat{\alpha}_t$, $\hat{\beta}_j = \hat{\beta}_t, \hat{\delta}_{k+1} = r_d\hat{\delta}_k$, $i, j, k > t$.

The first assumption is justified by Figure 2.2c showing no clear trend in later accident years. The impact of the second assumption for the development year effect should be minimal as it approaches zero in later development years. The last assumption for the calendar year effect capture the decreasing trend in Figure 2.2d where the ratio r_d can be estimated from the data.

For the autoregressive and persistence terms, the configuration is established with $\mu_{i,0} = \frac{1}{T}\sum_{i,j} y_{i,j}$, $y_{i,0} = r_0 y_{i,1}, i \leq t$ and $y_{i,0} = y_{t,0}, i > t$. due to the absence of a distinct trend for later accident years. The ratio r_0 is estimated using $r_0 = \frac{1}{17}\sum_{i=1}^{17} \frac{y_{i,1}}{y_{i,2}} \simeq 0.7$. Unobserved $y_{i,j-1}$, $i + j > t + 2$ in the CARR model forecast is estimated through simulation with $\hat{y}_{i,j-1} \sim \mathcal{D}(\mu_{i,j-1}, \boldsymbol{\nu})$. These assumptions allow reasonable guesses for the unknown parameters using data information. More refined techniques for estimating these boundary parameters can be explored with additional information from the insurance companies. Details regarding the four distributions and their shape parameters are presented on the next page.

Data distributions

1. **Generalized beta type II distribution**

Flexible data distribution is crucial for capturing irregular and extreme claims. Hence, the GB2 distribution proposed by Cummins et al. (1990) is a favorable choice as it nests a wide range of distributions defined on $\mathbb{R}^+$. The density of the GB2(a, b, p, q) distribution is

$$f_{\text{GB2}}(y; a, b, p, q) = \frac{|a| y^{ap-1}}{b^{ap} B(p, q)(1 + (y/b)^a)^{p+q}} \tag{2.16}$$

where $B(p, q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+1/a)\Gamma(q-1/a)}$ is the beta function, $\Gamma(z) = \int_0^\infty x^{z-1} \exp(-x)dx$ is the gamma function, $b > 0$ is the scale parameter, and $a, p, q > 0$ are the shape parameters. The mean and variance are given respectively by

$$\text{E}(Y) = \mu = \frac{b\Gamma(p + \frac{1}{a})\Gamma(q - \frac{1}{a})}{\Gamma(p)\Gamma(q)} \text{ and } \text{Var}(Y) = \frac{b^2\Gamma(p + \frac{2}{a})\Gamma(q - \frac{2}{a})}{\Gamma(p)\Gamma(q)} - \left[\frac{b\Gamma(p + \frac{1}{a})\Gamma(q - \frac{1}{a})}{\Gamma(p)\Gamma(q)}\right]^2. \tag{2.17}$$

Let $\boldsymbol{\nu} = (a, p, q)$ denote the set of shape parameters. Parameter a indicates tail thickness, with larger values giving thinner tails and a negative value giving inverse distribution. GB2 distribution captures both positive and negative skewness. The direction of skewness is determined by the relative values of p and q. Expressing the scale parameter b of E(Y)

$$b = \frac{\mu\Gamma(p)\Gamma(q)}{\Gamma(p + \frac{1}{a})\Gamma(q - \frac{1}{a})} \tag{2.18}$$

in terms of μ using (2.17), the density in (2.16) can be alternatively written as

$$f_{\text{GB2}}(y; a, \mu, p, q) = \frac{|a| y^{ap-1}\Gamma(p + q)\left[\Gamma(p + \frac{1}{a})\Gamma(q - \frac{1}{a})\right]^{ap}}{[\mu\Gamma(p)\Gamma(q)]^{ap}\,\Gamma(p)\Gamma(q)\left\{1 + \left[\frac{y\Gamma(p+\frac{1}{a})\Gamma(q-\frac{1}{a})}{\mu\Gamma(p)\Gamma(q)}\right]^a\right\}^{p+q}}. \tag{2.19}$$

Hence, the GB2 distribution for the error ϵ in the model $y = \mu\epsilon$ can be obtained by setting $\mu = 1$ in 2.19. GB2 distribution nests GG ($q \to \infty$), Weibull ($p =$

$1, q \to \infty$), and exponential ($a = 1, p = 1, q \to \infty$) distributions as special cases. See Chan et al. (2018) for details of the GB2 distribution.

Simulating random variables Y from the GB2 distribution for reserve forecasting involves leveraging the transformation of the GB2 distribution from a beta distribution, such that

$$\text{if } U = \frac{(\frac{Y}{b})^a}{1 + (\frac{Y}{b})^a} \sim \text{Beta}(p, q), \quad Y \sim \text{GB2}(a, b, p, q). \tag{2.20}$$

Hence, the simulation procedures are

$$Y = b\left(\frac{U}{1-U}\right)^{1/a} = \frac{\mu\Gamma(p)\Gamma(q)}{\Gamma(p+1/a)\Gamma(q-1/a)}\left(\frac{U}{1-U}\right)^{1/a} \quad \text{where } U \sim \text{Beta}(p, q) \tag{2.21}$$

using 2.17. The inverse method $Y = F^{-1}_{\text{GB2}}(U)$, $U \sim \text{Uniform}(0, 1)$ fails as the distribution function $F_{\text{GB2}}(\cdot)$ has no closed inverse function.

For the implementation of Bayesian inference, the `Stan` package is utilized within the `R` environment. Given that the built-in GB2 distribution is not inherently supported in `Stan`, the distribution is defined by specifying its log density, denoted as `beta2_log` using 2.19 as below

Code Listing 2.1 – Define GB2 function in `R`

```
functions{
real beta2.log(real x, real a, real mu, real p, real q) {
return log(fabs(a))+(a*p-1)*log(x)+lgamma(p+q)-(a*p)*log((
    mu*tgamma(p)*tgamma(q))/(tgamma(p+(1/a))*tgamma(q-(1/a)
    )))-lgamma(p)-lgamma(q)-(p+q)*log(1+pow((x*tgamma(p+(1/
    a))*tgamma(q-(1/a)))/(mu*tgamma(p)*tgamma(q)),a));
} }
```

where `tgamma()` is a gamma function and `lgamma()` is a log-gamma function. Then the model is specified as `y` $\sim$ `beta2(a,mu,p,q)` where `y` and `mu` are the observation vector and mean vector respectively.

2. **Generalized gamma distribution**

The generalized gamma (GG) distribution has two shape parameters with the density:

$$f_{\text{GG}}(y; a, \lambda, p) = \frac{a y^{ap-1} e^{-\left(\frac{y}{\lambda}\right)^a}}{\lambda^{ap}\Gamma(p)} \tag{2.22}$$

where λ is the scale parameter, a and p are the shape parameters ($\boldsymbol{\nu} = (a, p)$), and $\text{E}(Y) = \mu = \frac{\lambda\Gamma(p+1/a)}{\Gamma(p)}$. Expressing the scale parameter

$$\lambda = \frac{\mu\Gamma(p)}{\Gamma(p+1/a)} \tag{2.23}$$

in terms of μ, the density in (2.22) can be written as

$$f_{\text{GG}}(y; a, \mu, p) = \frac{a y^{ap-1} e^{-\left[\frac{y\Gamma(p+1/a)}{\mu\Gamma(p)}\right]^a}}{\left[\frac{\mu\Gamma(p)}{\Gamma(p+1/a)}\right]^{ap}\Gamma(p)}. \tag{2.24}$$

The GG distribution nests Weibull ($p = 1$), gamma ($a = 1$), and exponential ($a = p = 1$) distributions as special cases and lognormal as a limiting distribution ($b = (\sigma^2 a^2)^{1/a}$, $p = (a\mu + 1)/(\sigma^2 a^2)$, $a \to 0$). The GG is also the limiting distribution of GB2 with the scale parameter $\lambda = bq^{-1/a}$ and $q \to \infty$. See Morteza and Ahmadabadi (2010) for more properties of the GG distribution.

To simulate random variables Y from GG distribution for reserve forecast, one can utilize the fact that the GG distribution is transformed from a gamma distribution such that if $U = (\frac{Y}{\lambda})^a \sim \text{Gamma}(p, 1)$, $Y \sim \text{GG}(p, \lambda, a)$. Hence, the simulation procedures are

$$Y = \lambda U^{1/a} = \frac{\mu\Gamma(p)}{\Gamma(p+1/a)} U^{1/a} \text{ where } U \sim \text{Gamma}(p, 1). \tag{2.25}$$

Again, GG distribution is not a built-in distribution in `Stan`. The distribution is defined directly by specifying its log density called `ggamma_log` using (2.24), similar to the case of GB2 distribution. Then, the model is specified as `y` $\sim$ `ggamma(p,mu,a)`.

3. **Weibull distribution**

Weibull distribution has the density and distribution functions given by, respectively

$$f_{\mathrm{W}}(y;a,\lambda)=\frac{ay^{a-1}}{\lambda^a}e^{-(y/\lambda)^a} \quad \text{and} \quad F_{\mathrm{W}}(y;a,\lambda)=1-\exp\left[-\left(\frac{y}{\lambda}\right)^a\right] \tag{2.26}$$

where λ is the scale parameter, a is the shape parameter ($\nu=a$), and $\mathrm{E}(Y)=\mu=\lambda\Gamma(1+1/a)$. Expressing the scale parameter $\lambda=\frac{\mu}{\Gamma(1+1/a)}$ in terms of μ, the density in (2.26) can be written as

$$f_{\mathrm{W}}(y;a,\mu)=\frac{ay^{a-1}\Gamma(1+1/a)^a}{\mu^a}\exp\left\{-\left[\frac{y\Gamma\left(1+\frac{1}{a}\right)}{\mu}\right]^a\right\}. \tag{2.27}$$

The inverse method can be employed to simulate random variables Y from a Weibull distribution since the distribution function in (2.26) has a closed-form inverse function. Therefore, the simulation procedures are as follows:

$$U\sim\text{uniform}(0,1), \quad Y=F_{\mathrm{W}}^{-1}(U)=\lambda[-\ln(1-U)]^{1/a}. \tag{2.28}$$

Since Weibull distribution is a built-in function in `Stan`, the model is specified as `y` $\sim$ `Weibull(a,lam)` where `lam=mu/tgamma(1+1/a)`.

4. **Exponential extension distribution**

AbuJarad and Khan (2018) proposed the EE distribution with the density, distribution, and mean functions given respectively by

$$f_{\mathrm{EE}}(y;a,\lambda) = \frac{a}{\lambda}\left(1+\frac{y}{\lambda}\right)^{a-1}\exp\left[1-\left(1+\frac{y}{\lambda}\right)^a\right], \qquad y,a,\lambda>0, \tag{2.29}$$

$$\begin{aligned} F_{\mathrm{EE}}(y;a,\lambda) &= 1-\exp\left[1-\left(1+\frac{y}{\lambda}\right)^a\right], \\ \mathrm{E}(Y) &= \mu=\lambda\left[e\,\Gamma_u\left(\frac{1}{a}+1,1\right)-1\right], \end{aligned} \tag{2.30}$$

where the upper incomplete gamma function $\Gamma_u(x,y) = \int_y^\infty t^{x-1}e^{-t}dt$, and the shape parameter $\nu = a$. Expressing the scale parameter λ in (2.30) in terms of μ, the pdf in (2.29) becomes

$$f(y;a,\mu) = \frac{a\left[e\Gamma\left(\frac{1}{a}+1,1\right)-1\right]}{\mu}\left(1+\frac{y\left[e\Gamma\left(\frac{1}{a}+1,1\right)-1\right]}{\mu}\right)^{a-1}\exp\left\{1-\left[1+\frac{y[e\Gamma(\frac{1}{a}+1,1)-1]}{\mu}\right]^{a}\right\}. \tag{2.31}$$

Simulating random variables Y from the EE distribution for reserve forecasting can be achieved using the inverse method so that if $U \sim \text{uniform}(0,1)$,

$$Y = F_{\text{EE}}^{-1}(U) = \frac{\mu}{e\Gamma\left(\frac{1}{a}+1,1\right)-1}\left\{\left[1-\log(1-U)\right]^{\frac{1}{a}}-1\right\}$$

substituting $\lambda = \mu\left[e\Gamma\left(\frac{1}{a}+1,1\right)-1\right]^{-1}$ in 2.30. Alternatively, one can use the fact that EE distribution is transformed from a standard exponential distribution with pdf $f_{\text{E}}(u) = e^{-u}$ such that if $U = -\left[1-\left(1+\frac{Y}{\lambda}\right)^{a}\right] \sim \text{Exp}(1)$, $Y \sim \text{EE}(a,\mu)$. Hence, the simulation procedures are

$$Y = \frac{\mu}{e\Gamma\left(\frac{1}{a}+1,1\right)-1}\left[(U+1)^{1/a}-1\right] \quad \text{where } U \sim \text{Exp}(1). \tag{2.32}$$

Again, since the EE distribution is not readily available as a built-in function in `Stan`, one can define the distribution explicitly by establishing its log pdf called `eexp_log` using 2.31. Then, the model is specified as `y ~ eexp(a, mu)`.

2.3.2 Models for comparison

Models including the state-space model in (2.3), the Hoerl curve model in (2.6), and the CSR model in (2.7) attempt to capture the changes in development year pattern across accident years. Among these models, the CSR model (Meyers, 2015) interprets the changes in terms of changing settlement rates and attributes the accelerating claim settlement, for example, due to technological advancement. The proposed CARR model, on the other hand, uses autoregressive parameter γ_1 and persistent parameter

γ_2 to describe the development year pattern changes. To facilitate comparison with the CSR model, the losses $y_{i,j}$ in the CSR model, as defined by (2.13) with mean function (2.7), are assumed to follow some distributions $\mathcal{D}(\mu_{i,j}, \boldsymbol{\nu})$ described in **Data distributions** section. The model is referred as GB2-CSR if the CSR model adopts the GB2 distribution.

The proposed model is also compared with the popular CL method based on cumulative claims

$$z_{i,j} = \sum_{j'=1}^{j} y_{i,j'}, \quad j = 1, \ldots, t-i+1, \quad i = 1, \ldots, t$$

as the sum of incremental losses from the start to the current development year. The method first evaluates the lag-to-lag development factor

$$\varphi_j = \frac{\sum_{i=1}^{t-j+1} z_{i,j}}{\sum_{i=1}^{t-j+1} z_{i,j-1}}, \quad 2 \leq j \leq t \tag{2.33}$$

which is the ratio of the sum of cumulative losses over accident years from one development year j to the previous year $j-1$. Then it applies the factor φ_j to the cumulative loss $z_{i,j-1}$ to estimate the next cumulative loss

$$\hat{z}_{ij} = z_{i,j-1} \times \varphi_j, \quad 2 \leq j \leq t-i+1.$$

By assuming constant development factors φ_j independent of the accident year i and applying the method recursively to the next cumulative loss, the CL method can be used to estimate cumulative losses

$$\hat{z}_{ij} = \hat{z}_{i,t-i+1} \times \prod_{j'=t-i+2}^{j} \varphi_{j'}, \quad i = 2, ..., t, \quad j > t-i+1 \tag{2.34}$$

in the lower triangle as highlighted in blue in Table 2.1 using the latest observable cumulative losses $\hat{z}_{i,t-i+1} = z_{i,t-i+1}$ for accident year i. However, the method fails to estimate future cumulative losses $z_{i,j},\ i,j > t$ when the accident or development years are greater than t (in green in Table 2.1). The problem stems from the unavailability of loss information $z_{i,j},\ i > t$ for future accident and future development factors $\varphi_j,\ j > t$. To solve the problem, the estimation of $z_{t+h,1}, h = 1, 2, \ldots$ and $z_{1,t+h}, h = 1, 2, \ldots$ highlighted in dark green in Table 2.1 is proposed using other models such as

the best-proposed model with additional assumptions. Subsequently, the estimates for $z_{t+h,j},\ j = 2, \ldots$ and $z_{i,t+h},\ i = 2, \ldots$ can be computed using (2.34) replacing t with $t+h$. Lastly, the estimated incremental losses can be obtained by differencing the two consecutive cumulative losses. This modified CL method is called the *trapezoidal* CL method.

2.4 Models implementation

2.4.1 Bayesian approach

The fitting of the elaborated loss reserve models in (2.13) and (2.14) via the likelihood approach can be difficult because the density functions such as (2.19) and (2.31) and their derivatives are complicated. The Bayesian MCMC approach is adopted because it has several advantages over the classical likelihood approach.

Firstly, it incorporates both data information $f(\mathbf{y}_{1:T}|\boldsymbol{\theta})$ ($\mathbf{y}_{1:T}$ is defined in (2.10)) and prior information $\pi(\boldsymbol{\theta})$ in estimating the posterior distribution

$$\pi(\boldsymbol{\theta}|\mathbf{y}_{1:T}) = \frac{f(\mathbf{y}_{1:T}|\boldsymbol{\theta})\ \pi(\boldsymbol{\theta})}{\int f(\mathbf{y}_{1:T}|\boldsymbol{\theta})\ \pi(\boldsymbol{\theta})\ d\boldsymbol{\theta}} \propto f(\mathbf{y}_{1:T}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}).$$

where $\boldsymbol{\theta} = (\boldsymbol{\vartheta}, \boldsymbol{\nu})$ is the vector of all parameters and $\boldsymbol{\vartheta}$ is the vector of regression parameters. The joint density $f(\mathbf{y}_{1:T}|\boldsymbol{\theta})$ is the product of density for each observation $y_{i,j} \in \mathbf{y}_{1:T}$ given by (2.19), (2.24 and 2.27), and (2.31), respectively for the four distributions. The mean function $\mu_{i,j} = g(\eta_{i,j})$ where $\eta_{i,j}$ is defined in (2.14) using the vector of regression parameters $\boldsymbol{\vartheta}$. The specification of prior information $\pi(\boldsymbol{\theta})$ provides a powerful mechanism to incorporate expert knowledge often available to insurance companies. When there is a lack of prior information, non-informative or reference priors can be used. Using non-informative priors leads to objective Bayesian inference (Berger, 2006). Non-informative gamma priors is employed for parameters defined on positive supports and normal priors for unrestricted parameters, both with large variance. These unrestricted parameters refer to the regression parameters in

the mean function with log transformation. The hyperparameters are set or tuned to provide large variance and stable posterior distribution.

Secondly, inference for any parameter of interest can be conducted based on the posterior sample without relying on the asymptotic normality assumption in the likelihood approach. This asymptotic normality assumption is often not applicable due to insufficient sample size in real applications. Thirdly, combining the posterior distributions with the sampling distribution can obtain the entire posterior predictive distribution of future losses for the reserve forecast.

To illustrate the idea of posterior predictive distribution, this study considers the forecast of the future h years of losses as an example. These losses $\{y_{i,j} : (i,j) \in \mathcal{I}_{(t+1):(t+h)}\}$ where $\mathcal{I}_{a:b}$ defined in (2.11) are the diagonal entities of the dash lines in Figure 2.1. The vector of these $M = h(2t+h+1)/2$ losses is written as

$$\mathbf{y}_{(T+1):(T+M)} = (\mathbf{y}^d_{t+1}, \mathbf{y}^d_{t+2}, \ldots, \mathbf{y}^d_{t+h}) = (y_{t+1,1}, \ldots, y_{1,t+1}, y_{t+2,1}, \ldots, y_{1,t+2}, \ldots, y_{1,t+h}, \ldots, y_{t+h,1}) \tag{2.35}$$

in the order of calendar year (instead of accident year for the in-sample data) to facilitate forecasts by calendar years. Loss forecasts y_{T+m}, $m = 1, \ldots, M$ are based on the posterior predictive distribution defined as

$$f(\mathrm{y}_{T+m}|\mathbf{y}_{1:T}) = \int \cdots \int \prod_{s=1}^{m} \left[f(\mathrm{y}_{T+s}|\boldsymbol{\theta}, \mathbf{y}_{(T+1):(T+s-1)}, \mathbf{y}_{1:T})\, d\mathbf{y}_{(T+1):(T+s-1)} \right] \pi(\boldsymbol{\theta}|\mathbf{y}_{1:T}) d\boldsymbol{\theta}. \tag{2.36}$$

The integral over $d\mathbf{y}_{(T+1):(T+s-1)}$ is to integrate out the unobserved $\mathbf{y}_{(T+1):(T+s-1)}$ whereas the integral over $d\boldsymbol{\theta}$ is to incorporate prior information. These integrals can be approximated by the Monte Carlo estimator, constructed from the posterior samples according to

$$\hat{f}(\mathrm{y}_{T+m}|\mathbf{y}_{1:T}) = \frac{1}{L} \sum_{l=1}^{L} f(\mathrm{y}_{T+m}|\boldsymbol{\theta}^{(l)}, \mathbf{y}^{(l)}_{(T+1):(T+m-1)}, \mathbf{y}_{1:T}) \tag{2.37}$$

where L is the number of iterations after burn-in in each MCMC sampler run given the current window of information, and $\mathbf{y}^{(l)}_{(T+1):(T+m-1)}$ and $\boldsymbol{\theta}^{(l)}$ are the l-th draw in

the posterior sample of $\mathbf{y}_{(T+1):(T+m-1)}$ and $\boldsymbol{\theta}$ respectively. Essentially, (2.37) implies that the mean forecast $\hat{\mathrm{y}}_{T+m}$ is the mean of the posterior sample

$$\mathcal{S}_{\mathrm{y},T+m} = \{\mathrm{y}^{(l)}_{T+m} : \mathrm{y}^{(l)}_{T+m} \sim \mathcal{D}(\mu^{(l)}_{T+m}(\mathbf{y}^{(l)}_{(T+1):(T+m-1)}, \mathbf{y}_{1:T}), \boldsymbol{\nu}^{(l)}),\ l = 1, \ldots, L\}$$

where the distribution $\mathcal{D}(\cdot)$ has the density $f(\cdot)$, mean parameter $\mu^{(l)}_{T+m}$ and shape parameters $\boldsymbol{\nu}^{(l)}$. The mean parameter is the function $\mu^{(l)}_{T+m}(\mathbf{y}^{(l)}_{(T+1):(T+m-1)}, \mathbf{y}_{1:T})$ defined in (2.14) or (2.15). It depends on simulated observations $\mathbf{y}^{(l)}_{(T+1):(T+m-1)}$, regression parameters $\boldsymbol{\vartheta}^{(l)}$ and data $\mathbf{y}_{1:T}$. The vector of all parameters is $\boldsymbol{\theta}^{(l)} = (\boldsymbol{\vartheta}^{(l)}, \boldsymbol{\nu}^{(l)})$ at the l-th draw. The distribution $\mathcal{D}(\cdot)$ is one of the four distributions adopted in this study. The posterior predictive sample $\mathrm{y}^{(l)}_{T+m}$ can be simulated using 2.21, 2.25, 2.28, and 2.32, respectively. Apart from the mean forecast, these posterior predictive samples $\mathcal{S}_{\mathrm{y},T+m}$ also provide any u-th ($u \in (0,1)$) level quantile forecast $\hat{\mathrm{y}}_{T+m,u}$ given by the u-th quantile of the posterior sample $\mathcal{S}_{\mathrm{y},T+m}$. In-sample mean or quantile loss estimates are defined similarly based on the posterior samples $\mathcal{S}_{\mathrm{y},t}, t \leq T$ given the data $\mathbf{y}_{1:T}$. The mean of the posterior samples gives parameter estimates, that is, $\hat{\boldsymbol{\theta}} = \frac{1}{L}\sum_{l=1}^{L} \boldsymbol{\theta}^{(l)}$.

Lastly, the models are implemented via the user-friendly Bayesian software `Stan`, which applies the efficient Hamiltonian Monte Carlo (HMC) and No-U-Turn samplers (Hoffman and Gelman, 2014). `RStan` (Team, 2016) utilises the `Stan` program within `R` developed in the `C++` language. The HMC sampler (Duane et al., 1987; Neal, 1994) as a class of MCMC sampling methods converges faster than the conventional samplers such as random-walk Metropolis (Metropolis et al., 1953) and Gibbs sampler (Geman and Geman, 1984) for some complicated models with many parameters. Additionally, for more notable contributions of Bayesian MCMC to the actuarial literature, refer to works by Taylor (2012), Meyers (2015), Venter et al. (2017), Venter (2018) and Venter and Şahın (2018).

2.4.2 Convergence diagnostics

It is salient to ensure the convergence and independence of the MCMCs. To check these two conditions, graphical plots and numerical measures, can be applied. The autocorrelation function (ACF) plot visualizes the autocorrelation for each parameter. If an MCMC is dependent, it should decrease quickly with increasing lag. Hence, a positive long-lagged ACF indicates high autocorrelation. To reduce the autocorrelation, Annis et al. (2017) suggested running longer chains and drawing subsamples from the longer chains. However, if these procedures do not decrease the ACF, the model may be unstable (Vehtari et al., 2021). A negative ACF, on the other hand, indicates antithetic Markov chains. To avoid his problem, one can increase the number of chains. Another graphical plot is the `traceplot` (within the `Rstan` package). A fuzzy-caterpillar-look trace plot indicates convergence and independence of the MCMC (Lee and Wagenmakers, 2014).

Two measures are reported to evaluate the extent of convergence. The first measure is the number of effective samples n_{eff} after allowing for the dependency in the Monte Carlo sample. Samples that display positive autocorrelation often give smaller n_{eff}, whereas those with negative autocorrelation have larger n_{eff}. As a guideline, n_{eff} is expected to be sufficiently large, such as at least 2000.

The second measure proposed by Gelman and Rubin (1992) is a scale reduction measure called $\widehat{R}$. It is the ratio of the variance of a parameter when the data is pooled across all of the chains to the variance within a chain. The criterion used for convergence is $\widehat{R}$ within 1 ± 0.0001 for each parameter. More information for the two criteria can be found in Peters et al. (2010).

Apart from inspecting the two convergence measures, the ACF and trace plots for each model parameter are carefully checked. When the two criteria, namely $n_{\text{eff}} \geq 2000$ and $\widehat{R} \in 1 \pm 0.0001$, are not met (confirmed by the ACF and trace plots), the Monte Carlo sample size (`iter`), including burn-in, will be increased gradually. Depending on model complexity, the sample size is chosen from 5,000 to 60,000 per chain, including the burn-in of 5,000 iterations per chain, and we use four chains.

2.4.3 Model performance measures

Model selection is based on two model performance measures. The DIC consists of two components:

$$\text{DIC} = \bar{D} + p_D$$

which measures the model fit and penalizes model complexity, respectively. The first component is defined as the posterior expectation of deviance

$$\bar{D} = E_{\theta|\text{y}}[D(\boldsymbol{\theta})] = E_{\theta|\text{y}}[-2\ln f(\mathbf{y}|\boldsymbol{\theta})]$$

where the deviance is -2 times the log-likelihood and the second component evaluates the effective number of parameters defined as the difference between the posterior mean of the deviance and the deviance evaluated at the posterior mean $\bar{\boldsymbol{\theta}}$ of the parameters:

$$p_D = \bar{D} - D(\bar{\boldsymbol{\theta}}) = E_{\theta|\text{y}}[-2\ln f(\mathbf{y}|\boldsymbol{\theta})] + 2\ln f(\mathbf{y}|\bar{\boldsymbol{\theta}}).$$

The posterior expectation is estimated based on the posterior sample of the MCMC output.

The predictive performance of the models is also assessed by comparing the observed losses with the predicted losses using the mean squared error (MSE)

$$\text{MSE} = \frac{1}{T}\sum_{i=1}^{t}\sum_{j=1}^{i}(y_{i,j} - \hat{y}_{i,j})^2$$

where $\hat{y}_{i,j} = \mu_{i,j}$ in (2.14) or (2.15) for mean estimate and $\hat{y}_{i,j} = \hat{y}_{i,j,u}$, $u \in (0,1)$ where $y_{i,j,u}$ is the u-th quantile of $y_{i,j}$. In summary, DIC provides adjusted model fit, allowing for model complexity, whereas MSE assesses prediction accuracy, which is more important in loss reserving.

2.5 Empirical studies

2.5.1 Model selection

The data analysis begins with searching for some efficient mean functions adopting the most flexible GB2 error distribution. Table 2.2 presents the 41 mean functions considered in this study. These functions are arranged in two panels (with and without persistence), each with 21 mean functions of accident (A), development (D), and calendar (C) year effects. The 21 mean functions, in seven groups of three each, are identified by a list of parameter symbols (see **Mean functions** section), which indicates the type of parameters included in the mean function. The three members of each group are none, C-ANOVA (δ), and C-ANCOVA (δ_l), showing the exclusion of calendar year effect, inclusion of ANOVA type calendar year effect, and inclusion of ANCOVA type calendar year effect, respectively. Then the seven groups correspond to the exclusion or inclusion of the accident and development year effects, namely, none, A-ANOVA (α), D-ANOVA (β), AD-ANOVA ($\alpha\beta$), D-ANCOVA (β_q), A-ANOVA-D-ANCOVA ($\alpha\beta_q$), and AD-ANCOVA ($\alpha_s\beta_q$). The mean function is presented using these symbols. For example, GB2-$\alpha\beta$ represents the model with parameters α_i and β_j in the mean function and GB2 as the data distribution.

Table 2.2 – List of mean functions expressed using the symbols in (2.14) and (2.15) in Section 2.3.1

No.	Non CARR	CARR	No.	Non CARR	CARR	No.	Non CARR	CARR	No.	Non CARR	CARR
1	–	$\gamma_1\gamma_2$	7	β	$\boldsymbol{\beta\gamma_1\gamma_2}$	13	β_q	$\beta_q\gamma_1\gamma_2$	19	$\alpha_s\beta_q$	$\alpha_s\beta_q\gamma_1\gamma_2$
2	δ	$\delta\gamma_1\gamma_2$	8	$\boldsymbol{\beta\delta}$	$\beta\delta\gamma_1\gamma_2$	14	$\beta_q\delta$	$\beta_q\delta\gamma_1\gamma_2$	20	$\alpha_s\beta_q\delta$	$\alpha_s\beta_q\delta\gamma_1\gamma_2$
3	δ_l	$\delta_l\gamma_1\gamma_2$	9	$\boldsymbol{\beta\delta_l}$	$\boldsymbol{\beta\delta_l\gamma_1\gamma_2}$	15	$\beta_q\delta_l$	$\beta_q\delta_l\gamma_1\gamma_2$	21	$\alpha_s\beta_q\delta_l$	$\alpha_s\beta_q\delta_l\gamma_1\gamma_2$
4	α	$\alpha\gamma_1\gamma_2$	10	$\alpha\beta$	$\boldsymbol{\alpha\beta\gamma_1\gamma_2}$	16	$\alpha\beta_q$	$\alpha\beta_q\gamma_1\gamma_2$		–	
5	$\alpha\delta$	$\boldsymbol{\alpha\delta\gamma_1\gamma_2^*}$	11	$\boldsymbol{\alpha\beta\delta}$	$\alpha\beta\delta\gamma_1\gamma_2$	17	$\alpha\beta_q\delta$	$\alpha\beta_q\delta\gamma_1\gamma_2$		–	
6	$\alpha\delta_l$	$\alpha\delta_l\gamma_1\gamma_2$	12	$\alpha\beta\delta_l$	$\boldsymbol{\alpha\beta\delta_l\gamma_1\gamma_2}$	18	$\alpha\beta_q\delta_l$	$\alpha\beta_q\delta_l\gamma_1\gamma_2$		–	

Note: 1. models in bold face with DIC$<$ 3070 are selected;
2. model with * has no log-transformation

In fitting these 41 mean functions with the most flexible GB2 distribution, the log-transformed mean functions provide consistently better model fit with lower DIC ($<$ 3070) except the mean function $\alpha\delta\gamma_1\gamma_2$. Hence, the mean functions in (2.14) without log transformation are indicated by '*'. It is worth noting that some of these models may not converge or exhibit poor model fits. The model performance measures, DIC and MSE for the best GB2-$\beta\delta_l$ model according to DIC, are reported in Table 2.3. In this model, the calendar year effect replaces the accident year effect, possibly because the former displays a clearer linear trend in Figure 2.2. due to the superior performance in terms of the lowest MSE. Table 2.3 also includes the outcomes of GB2-$\beta\delta_l\gamma_1\gamma_2$ model, incorporating an additional development year persistence effect. A similar exploration is conducted for GG, Weibull, and EE distributions, and results are summarized in Table 2.3. The eight models are ranked according to DIC and MSE. Results show that, in general, models with the persistence effect are more likely to provide better in-sample estimates, according to MSE. Across distributions, GB2 and GG provide the best model performance. Weibull is slightly worse, and EE is the worst.

Table 2.3 – Performance measures and ranks in brackets for the best-fitted model of each distribution without and with development year persistence as well as for the GB2-CSR model

Performance	**GB2**		**GG**		**Weibull**		**EE**		**GB2-CSR**
Measures	$\beta\delta_l$	$\boldsymbol{\beta\delta_l\gamma_1\gamma_2}$	$\beta\delta_l$	$\beta\delta_l\gamma_1\gamma_2$	$\alpha\beta$	$\alpha\beta\gamma_1\gamma_2$	β	$\beta\gamma_1\gamma_2{}^*$	$\alpha\beta(1-\omega)^{(i-1)}$
DIC	**2976 (1)**	3007 (4)	2977 (2)	3000 (3)	3011 (5)	3019 (6)	3132 (9)	3078 (8)	3048 (7)
MSE ($\times$1000)	3594 (6)	**2671 (1)**	3697 (7)	2890 (4)	2887 (3)	2733 (2)	9814 (9)	6077 (7)	3551(5)

Note: 1. models in bold face are the best according to DIC & MSE;
2. model with * has no log transformation

Previously, Nadarajah and Haghighi (2011) and AbuJarad and Khan (2018) recommended EE distribution. The former preferred EE among the three distributions, Gamma, Weibull, and the Exponentiated Exponential in Bayesian inference. AbuJarad and Khan (2018) suggested EE contrasting with exponential and exponentiated exponential distributions for running `Stan`. Regarding the data analysis, in-sample estimates using the EE-$\beta\gamma_1\gamma_2$ model have very high MSE. In contrast, the GB2-

$\beta\delta_l\gamma_1\gamma_2$ model emerges as the preferred model, providing the most accurate in-sample estimates. This selection is determined by the sum of ranks, underscoring the superior overall fit of the GB2-$\beta\delta_l\gamma_1\gamma_2$ model. In addition, the CSR model described in (2.7) and the CL method outlined in Section 2.3.2 are applied for comparative purposes.

Comparing CARR models with the GB2-CSR model adopting the best GB2 distribution, Table 2.3 shows that the GB2-CSR model performs the worst according to DIC and MSE. A plausible explanation is that the CARR models adopt a more relaxed assumption on the development year pattern change across accident years, which is driven by persistence across development years in the form $\gamma_1 \log(y_{i,j-1})+\gamma_2 \log(\mu_{i,j-1})$ using parameters γ_1 and γ_2. Essentially, it uses the previous claim information to forecast claims in the next development year based on development year persistence. This assumed pattern is more data-driven. So, it is different from the explicit pattern $\beta_j(1-\omega)^{(i-1)}$ based on parameter ω in the CSR model, making the CARR model more adaptive to different data. For the CSR model, an increase (decrease) in payment rate during the earlier development period will cause more (less) payments as the accident year increases. However, the converse is not always true, as other hidden factors can give rise to the observed phenomenon. Nevertheless, a slowdown in claim settlement is found in loss data because ω is estimated to be significantly positive (0.0133), in agreement with Figure 2.2a showing slower claim increments for later accident years. Perhaps technological advancements may not yet affect the settlement rate during the data period of 1978-1995.

The estimation accuracy using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model can be visualized by comparing the observed and estimated losses in Figure 2.3. The estimated losses are provided at the mean and 75%, 85%, and 90% quantiles. These losses are indicated by red and blue lines, respectively, for the earlier (1978-1983) and later (1984-1995) accident year periods with two different observed development year patterns as shown in Figure 2.2a. The earlier accident years show a sharper and earlier rise and drop of claims across development years than the later accident years. Although the mean function with one common development year ANOVA model (β) does not differentiate effects between these two periods, the estimated mean and quantile lines capture

well the slower decay trend in the later period. The CL method tends to provide underestimates that do not differentiate between the two periods.

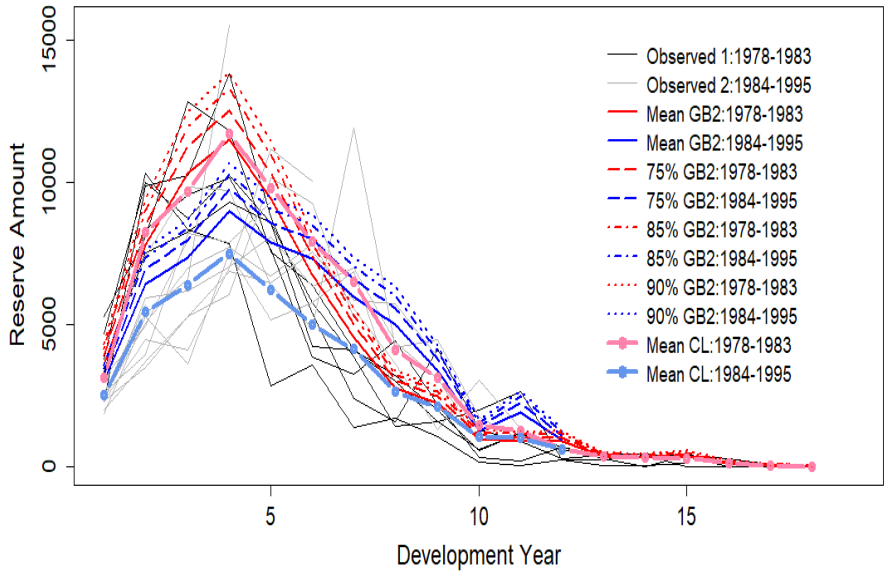


Figure 2.3 – Observed and estimated losses using the mean and quantile estimates of the $\beta\delta_l\gamma_1\gamma_2$-GB2 model as well as the CL method

To further visualize the estimation/prediction accuracy, Figure 2.4 provides the heat maps to distinguish over and underestimation of losses by plotting the differences between estimated and observed losses. The slower rise in the fitted mean in Figure 2.3 gives more under-fits (red) in early development years (1-4) and more over-fits in later development years (5-8) for earlier accident years than later accident years in Figures 2.4a. Figures 2.4a and 2.4b also show that GG distribution tends to provide more underestimates (in red), which can be hazardous to insurance companies, whereas GB2 distribution provides more accurate losses (in grey).

Figures 2.4c, 2.4e, and 2.4d display the accuracy for the 65% quantile of GB2-$\beta\delta_l\gamma_1, \gamma_2$ model, 75% quantile of GB2-$\beta\delta_l\gamma_1, \gamma_2$ model, and the CL method, respectively. The quantile estimates show some overestimation. However, the CL method displays heavier over and underestimation, indicating lower accuracy. Figure 2.4f depicts out-of-sample forecasts, which will be discussed in Section 2.6.3.

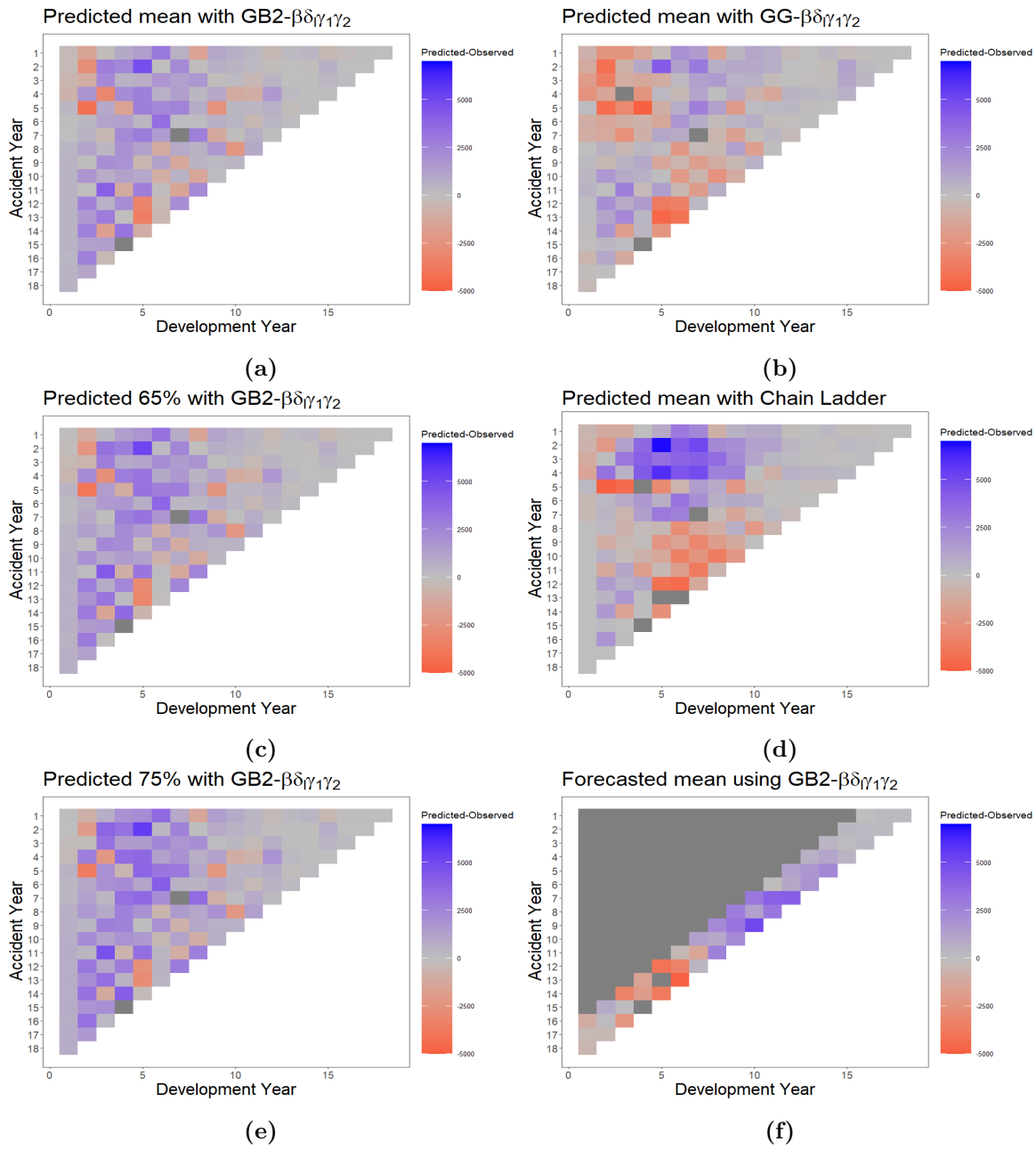


Figure 2.4 – Differences between predicted and observed losses in (a) and (b) for comparing between GB2 and GG distributions using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model; in (c), (e) and (d) for comparing between 65%, 75% quantile of the GB2-$\beta\delta_l\gamma_1\gamma_2$ model and CL method; and in (f) for validating the out-of-sample forecast using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model fitted to data with $t \leq 15$.

2.5.2 Simulation experiments

Validating parameter accuracy

It is challenging to evaluate the accuracy of the parameter estimates from the `Stan` program in `R` as the true values are unknown. In this situation, researchers often conduct simulation experiments. Based on the best GB2-$\beta\delta_l\gamma_1\gamma_2$ model, run-off triangles, each with $T = 171$ observations (yellow cells in Table 2.1), are simulated using the parameter estimates as the true values. Then, the GB2-$\beta\delta_l\gamma_1\gamma_2$ model is fitted to each simulated data set. If a data set gives one or more cases of $n_{\text{eff}} < 1000$, $\widehat{R} \notin 1 \pm 0.001$ or DIC > 3080, the fitted model is considered unstable. Then, the data is replaced with another simulated data, and the model is refitted. These procedures are continued until $S = 100$ sets of parameter estimates are obtained.

Figure 2.5 displays the boxplots of 100 estimates for 21 regression parameters, colored according to the type of regression parameters. Only regression parameters are plotted because, based on research experience, the mean and quantile estimates are less sensitive to the shape parameters. On inspecting these boxplots, it is not surprising that parameters δ_l, γ_1, and γ_2 display higher precision than parameters β_j because the calendar effect is clearly linear. Also, there are more observations to estimate the development year persistence γ_1 and γ_2 than β_j. Moreover, the precision of β_j decreases with j with fewer observations. Table 2.4 reports the average and standard deviation (SD) for these 100 estimates of each regression parameter. The SDs of δ_l, γ_k, and β_j reflect their precision observed in Figure 2.5.

To assess the accuracy of each parameter, two measures are reported. The first measure is the mean absolute deviation (MAD) defined for parameter θ as

$$\text{MAD}(\theta) = \frac{1}{S}\sum_{r=1}^{S} |\hat{\theta}_r - \theta_0|$$

where θ_0 denotes the true parameter value and $\hat{\theta}_r$ is the estimate for the r-th simulated data. The second measure, called coverage (CoV), reports the proportion of the 95% credible intervals that cover the true parameter. The CoV measure is defined as

$$\text{CoV}(\theta) = \frac{1}{S}\sum_{r=1}^{S} I(\theta_0 \in (\hat{\theta}_{r,0.025}, \hat{\theta}_{r,0.975})) \times 100$$

where $\hat{\theta}_{r,u}$ is the $u \in (0,1)$ quantile in the posterior sample of θ for the r-th simulated data and $I(E)$ is the indicator function for an event E. The accuracy of the estimates is reasonable for all parameters according to MAD and CoV. Moreover, the DIC averaged over 100 simulations is 2972. This value is close to 3007 in the real data analysis. In summary, parameter estimates using `Stan` program are reliable.

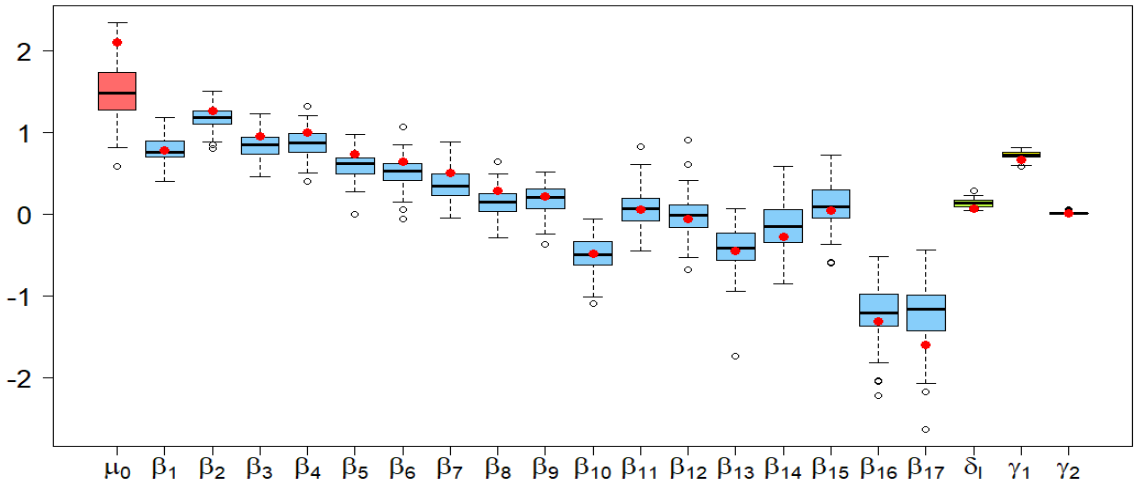


Figure 2.5 – Boxplots of 100 estimates for 21 regression parameters with true values indicated by red dots

Validating CARR model robustness

In this simulation study, the data-generating process is varied in a way so that some effects in (2.15) are switched on and off to compare the robustness of models to different data-generating processes. To investigate the robustness of the CARR model, two models, GB2-$\beta\delta_l\gamma_1\gamma_2$ (model A) and GB2-$\beta\delta_l$ (model B), are considered. Data is simulated from one model and fitted to the other using parameter estimates as true values. A total of $S = 100$ sets of parameter estimates with DIC$<$ 3080 are acquired by generating approximately 300 data sets for Scenario I (from A to B) and 150 data

Parameters	True	Estimate	SD	CoV (%)	MAD	Parameters	True	Estimate	SD	CoV(%)	MAD
μ_0	2.10	1.50	0.42	73	0.61	β_{11}	0.06	0.06	0.21	92	0.18
β_1	0.78	0.79	0.21	100	0.12	β_{12}	−0.06	−0.03	0.23	94	0.19
β_2	1.27	1.18	0.17	96	0.13	β_{13}	−0.45	−0.42	0.25	96	0.20
β_3	0.95	0.84	0.18	94	0.15	β_{14}	−0.28	−0.15	0.27	91	0.25
β_4	1.00	0.87	0.20	92	0.17	β_{15}	0.05	0.10	0.29	96	0.21
β_5	0.74	0.60	0.20	96	0.17	β_{16}	−1.31	−1.18	0.36	94	0.26
β_6	0.64	0.50	0.19	90	0.17	β_{17}	−1.60	−1.21	0.45	91	0.46
β_7	0.50	0.37	0.19	91	0.19	δ_l	0.07	0.13	0.09	99	0.06
β_8	0.28	0.13	0.20	88	0.18	γ_1	0.66	0.72	0.06	78	0.07
β_9	0.21	0.18	0.20	96	0.14	γ_2	0.01	0.01	0.02	100	0.01
β_{10}	−0.48	−0.49	0.20	96	0.17	**Average DIC = 2972**					

Table 2.4 – True parameter values, estimates, standard deviation, coverage percentage and MAD of in-sample fit using GB2-$\beta\delta_l\gamma_1\gamma_2$ model with $T = 171$ observations and $S = 100$ simulated data sets.

sets for Scenario II (from B to A). Results are reported in Table 2.5. It is observed that $\hat{\mu}_0$ is higher for Scenario I but lower for Scenario II due to the non-inclusion and inclusion of persistence effect γ_1, γ_2 respectively in the fitted model. The inclusion of the persistence effect can partially account for the claim level so that $\hat{\mu}_0$ can be lower. These biases affect the accuracy of the first few β_j related to high claim levels. Moreover, the average DIC is 3038 for Scenario I and 2960 for Scenario II, showing the relative advantage in robustness for the in-sample fit of the CARR model even when the data do not display development year persistence. Results for calendar year reserve forecasts are presented in **Validation using 1978-1992 losses** section. It is worth noting that the DIC value of 2960 closely resembles the DIC value of 3007 in Table 2.3 for the loss data.

2.6 Loss reserve forecast

2.6.1 Partial and calendar year reserve forecasts

Given the in-sample data $\mathbf{y}_{1:T}$, the posterior samples of parameters $\boldsymbol{\theta}^{(l)}$ and the forecasts $\mathbf{y}^{(l)}_{(T+1):(T+m-1)}$, $l = 1, \ldots, L$ where L is the posterior sample size, the posterior

Parameters	True	Estimate	SD	CoV(%)	MAD	Parameters	True	Estimate	SD	CoV(%)	MAD
Scenario I: simulating from GB2-$\beta\delta_l\gamma_1\gamma_2$ model and fitting to GB2-$\beta\delta_l$ model											
μ_0	2.10	6.59	0.25	0	8.99	β_{11}	0.06	-0.35	0.26	66	0.44
β_1	0.78	1.05	0.20	76	0.56	β_{12}	-0.06	-0.47	0.28	80	0.84
β_2	1.27	1.69	0.20	34	0.86	β_{13}	-0.45	-0.89	0.31	72	0.92
β_3	0.95	1.82	0.20	0	1.73	β_{14}	-0.28	-1.01	0.33	46	1.47
β_4	1.00	1.90	0.20	0	1.81	β_{15}	0.05	-0.85	0.37	20	1.79
β_5	0.74	1.75	0.21	0	2.04	β_{16}	-1.31	-1.79	0.44	92	1.01
β_6	0.64	1.56	0.22	0	1.83	β_{17}	-1.60	-2.25	0.62	90	1.36
β_7	0.50	1.28	0.22	10	1.54	δ	0.07	0.20	0.09	70	0.29
β_8	0.28	0.90	0.23	30	1.23	γ_1	0.66	-	-	-	-
β_9	0.21	0.57	0.24	70	0.74	γ_2	0.01	-	-	-	-
β_{10}	-0.48	-0.31	0.25	92	0.46	**Average DIC = 3038**					
Scenario II: simulating from GB2-$\beta\delta_l$ model and fitting to GB2-$\beta\delta_l\gamma_1\gamma_2$ model											
μ_0	6.90	3.91	0.84	0	30.88	β_{11}	-0.09	-0.02	0.35	80	3.14
β_1	0.89	2.80	0.83	0	19.81	β_{12}	-0.83	-0.67	0.38	90	2.97
β_2	1.62	1.27	0.23	40	3.88	β_{13}	-1.41	-0.77	0.42	70	5.56
β_3	1.70	1.06	0.30	0	7.54	β_{14}	-1.71	-0.57	0.47	50	6.86
β_4	1.91	1.17	0.32	10	8.22	β_{15}	-0.90	-0.68	0.51	60	6.80
β_5	1.49	0.91	0.35	10	9.08	β_{16}	-1.99	-0.79	0.57	50	8.53
β_6	1.46	0.82	0.35	0	8.38	β_{17}	-1.29	-1.09	0.70	40	11.24
β_7	1.38	0.81	0.35	0	7.55	δ	0.09	0.27	0.25	90	2.46
β_8	0.88	0.38	0.35	10	7.44	γ_1	-	0.32	0.26	-	-
β_9	0.74	0.41	0.34	40	5.54	γ_2	-	0.51	0.32	-	-
β_{10}	0.03	-0.06	0.35	80	4.56	**Average DIC = 2960**					

Table 2.5 – True parameter values, estimates, standard deviation, coverage percentage, and MAD of in-sample fit for each scenario with $T = 171$ observations and $S = 100$ simulated data sets.

samples of the means $\boldsymbol{\mu}^{(l)}_{(T+1):(T+m-1)}$ and the next mean $\mu^{(l)}_{(T+m)}$ can be calculated using the mean functions in (2.14) and (2.15).

Then the posterior predictive distribution in (2.37) for the next loss y_{T+m} is given by

$$\hat{f}(\mathrm{y}_{T+m}|\mathbf{y}^{(l)}_{(T+1):(T+m-1)}, \boldsymbol{\theta}^{(l)}, \mathbf{y}_{1:T}) = \hat{f}(\mathrm{y}_{T+m}|\mu^{(l)}_{T+m}, \boldsymbol{\theta}^{(l)}). \tag{2.38}$$

Then the posterior sample of $\mathrm{y}^{(l)}_{T+m}, l = 1, \ldots, L$ can be obtained from simulation, that is, $\mathrm{y}^{(l)}_{T+m} \sim \hat{f}(\mathrm{y}_{T+m}|\mu^{(l)}_{T+m}, \boldsymbol{\theta}^{(l)})$. The mean of these posterior samples of forecasts

at various cells is reported in Table 2.6. The sum of deep and light blue loss forecasts in the lower run-off triangle provides the partial reserve forecast of 223,905 using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model.

Additionally, $h = 3$ future calendar year loss forecasts $s_{t+1}, \ldots, s_{t+h}$ are presented in (2.12). These sums of future losses per calendar year include future losses denoted by y_{T+m}, $m = 1, \ldots, M$ where $M = h(2t + h + 1)/2$. The vector of these losses is denoted by $\mathbf{y}_{(T+1):(T+M)}$ and defined in (2.35). The posterior predictive samples $\mathcal{S}_{t+k} = \{s^{(l)}_{t+k}, l = 1, \ldots, L\}$, $k = 1, \ldots, h$ of the sum of future loss forecasts by calendar years can be obtained by summing the future losses $\mathrm{y}^{(l)}_{T+m}$ in the posterior predictive sample in calendar year $t + k$ given by

$$s^{(l)}_{t+k} = \sum_{r=1}^{t+k} y^{(l)}_{r,t+k-r+1} = \sum_{m=(k-1)t+(k-1)k/2+1}^{kt+k(k+1)/2} \mathrm{y}^{(l)}_{T+m}. \tag{2.39}$$

Similarly, the mean and quantiles of the posterior predictive samples $\mathcal{S}_{t+k}, k = 1, \ldots, h$ can be obtained for forecasting.

Accident Year	Development Year																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1978	3323	8332	9572	10172	7631	3855	3252	4433	2188	333	199	692	311	0.01	405	293	76	14	12	21	69
1979	3785	10342	8330	7849	2839	3577	1404	1721	1065	156	35	259	250	420	6	1	0.01	1	3	10	
1980	4677	9989	8746	10228	8572	5787	3855	1445	1612	626	1172	589	438	473	370	31	49	16	20		
1981	5288	8089	12839	11829	7560	6383	4118	3016	1575	1985	2645	266	38	45	115	154	73	20			
1982	2294	9869	10242	13808	8775	5419	2424	1579	4149	1296	917	295	428	359	827	371	125	29			
1983	3600	7514	8247	9327	8584	4245	4096	3216	2014	593	1188	691	368	403	826	372	124	30			
1984	3642	7394	9838	9733	6377	4884	11920	4188	4492	1760	944	921	553	489	916	398	131	33			
1985	2463	5033	6980	7722	6702	7834	5579	3622	1300	3069	1370	1101	609	516	942	402	132	31			
1986	2267	5959	6175	7051	8102	6339	6978	4396	3107	903	1100	881	515	456	875	383	128	30			
1987	2009	3700	5298	6885	6477	7570	5855	5751	3871	1627	1399	1013	567	497	915	398	133	30			
1988	1860	5282	3640	7538	5157	5766	6862	2572	2427	1184	1125	879	509	456	877	387	129	30			
1989	2331	3517	5310	6066	10149	9265	5262	3848	2825	1288	1183	910	526	471	889	390	129	30			
1990	2314	4487	4112	7000	11163	10057	6921	4301	3031	1349	1222	928	535	478	911	399	132	31			
1991	2607	3952	8228	7895	9317	7445	5811	3814	2759	1280	1175	901	528	475	897	396	133	32			
1992	2595	5403	6579	15546	10889	8286	6174	3977	2873	1295	1196	934	544	480	924	401	135	33			
1993	3155	4974	7961	9752	8573	7019	5479	3675	2716	1264	1191	910	537	473	902	399	133	31			
1994	2626	5704	7658	9323	8238	6775	5352	3621	2694	1255	1165	906	529	478	901	396	132	31			
1995	2827	6676	8065	9591	8392	6923	5448	3648	2728	1258	1180	910	537	481	914	407	135	32			
1996	2617	6705	8036																		
1997	3072	6731																			
1998	3409																				

Table 2.6 – Mean forecasts using GB2-$\beta\delta_l\gamma_1\gamma_2$ for calendar year reserves (deep blue and green highlights) and partial reserve (lower run-off triangle in deep and light blue highlights).

Forecasts using the trapezoidal CL method are also provided for comparison. The observed cumulative (upper panel) and incremental (lower panel) losses and forecasts

(in blue and green cells) are reported in Table 2.7. The upper panel also reports the development factors φ_j defined in (2.33) in Section 2.3.2. These factors show fast increases in earlier development years, slower increases in the middle (after the peak of incremental losses), and near-zero increases near the end. The incremental losses in the first row and first column of the lower panel in green cells are estimated using the best GB2-$\beta\delta_l\gamma_1\gamma_2$ model. The mean and quantile calendar year reserve forecasts are reported in Table 2.8 for the GB2-$\beta\delta_l\gamma_1\gamma_2$ model, GG-$\beta\delta_l\gamma_1\gamma_2$ model and trapezoidal CL method. The partial reserve forecast is estimated to be 206,383 using the CL method, with individual loss estimates reported in Table 2.7(b). This partial reserve forecast is lower than 223,905 using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model with individual loss estimates reported in Table 2.6.

2.6.2 Comparing calendar year reserve forecasts

Results of the calendar year reserve forecasts in Table 2.8 show that the GB2 distribution provides higher forecasts than the GG distribution for all the mean and quantile estimates of all calendar years. The trapezoidal CL method provides the lowest forecasts. Across calendar years, forecasts using GB2 distribution tend to drop for quantiles less than 75%, stabilize for mean, and increase for quantiles greater than 80%. These trends drop consistently for GG distribution and trapezoidal CL method. Figure 2.6 visualizes these trends of calendar year reserve by graphing the observed (black line), estimated/forecasted mean (red line), and estimated/forecasted quantiles (grey lines) using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model and the trapezoidal CL method (blue line). The forecast period is highlighted with a light blue background. All calendar year reserve forecasts follow some quadratic trends from 1978 to 1989. These quadratic trends tend to move above the observed calendar year losses from 1981. The red (mean) and grey (quantiles) lines using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model tend to converge to the observed calendar year loss in 1990 and move up closely together. On the other hand, the blue line using the trapezoidal CL method tends to stabilize below the observed calendar year losses from 1990. These underestimates using the trapezoidal CL method can be seen in Figure 2.3 and confirmed by the MSE of 4800 thousand

Accident	Development Year																				
Year	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1978	3323	10483	18902	29057	37571	44455	50115	53692	56412	57675	58790	59343	59670	59946	60189	60315	60363	60378	60391	60415	60502
1979	3785	11941	21530	33097	42794	50635	57083	61157	64255	65694	66964	67593	67966	68280	68558	68701	68755	68772	68787	68815	
1980	4677	14755	26603	40897	52880	62568	70535	75569	79398	81176	82745	83523	83983	84371	84714	84892	84958	84980	84998		
1981	5288	16682	30079	46240	59788	70742	79750	85442	89771	91781	93555	94434	94955	95393	95781	95982	96057	96082			
1982	2294	7237	13049	20059	25937	30689	34596	37066	38944	39816	40585	40967	41193	41383	41551	41638	41671	41681			
1983	3600	11357	20477	31480	40703	48160	54293	58168	61115	62483	63691	64290	64644	64943	65207	65343	65394	65411			
1984	3642	11490	20716	31847	41178	48722	54926	58846	61828	63212	64434	65040	65398	65700	65968	66106	66157	66174			
1985	2463	7770	14010	21537	27847	32950	37145	39796	41813	42749	43575	43985	44227	44432	44612	44706	44741	44752			
1986	2267	7152	12895	19823	25631	30328	34189	36629	38485	39347	40108	40485	40708	40896	41062	41148	41180	41191			
1987	2009	6338	11427	17567	22714	26876	30298	32461	34105	34869	35543	35877	36075	36242	36389	36465	36494	36503			
1988	1860	5868	10580	16264	21030	24883	28051	30053	31576	32283	32907	33216	33399	33554	33690	33761	33787	33796			
1989	2331	7354	13259	20383	26355	31184	35154	37663	39572	40458	41240	41628	41857	42050	42221	42310	42343	42354			
1990	2314	7300	13162	20234	26163	30956	34898	37389	39283	40163	40939	41324	41552	41744	41913	42001	42034	42045			
1991	2607	8224	14829	22796	29476	34876	39317	42123	44257	45248	46123	46556	46813	47029	47221	47319	47356	47368			
1992	2595	8187	14761	22691	29340	34716	39136	41929	44053	45040	45911	46342	46597	46813	47003	47102	47138	47150			
1993	3155	9953	17946	27588	35671	42207	47581	50977	53560	54760	55818	56343	56653	56915	57146	57266	57311	57325			
1994	2626	8284	14937	22963	29690	35130	39603	42430	44580	45578	46459	46896	47154	47372	47565	47664	47702	47714			
1995	2827	8919	16080	24720	31963	37819	42635	45678	47992	49067	50015	50485	50763	50998	51205	51313	51353	51366			
1996	2617	8256	14886																		
1997	3072	9691																			
1998	3409																				
φ_j	-	3.1548	1.8030	1.5373	1.2930	1.1832	1.1273	1.0714	1.0507	1.0224	1.0193	1.0094	1.0055	1.0046	1.0041	1.0021	1.0008	1.0003	1.0001	1.0002	1.0003

Accident	Development Year																				
Year	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1978	3323	7160	8418	10156	8514	6884	5660	3577	2720	1263	1115	553	327	276	244	126	47	15	13	24	87
1979	3785	8156	9589	11568	9697	7841	6447	4074	3099	1439	1270	629	372	314	278	143	54	17	15	27	
1980	4677	10078	11849	14294	11983	9689	7967	5034	3829	1778	1569	778	460	388	343	177	66	22	19		
1981	5288	11394	13396	16161	13548	10954	9007	5692	4329	2010	1774	879	520	439	388	200	75	24			
1982	2294	4943	5812	7011	5877	4752	3908	2469	1878	872	770	382	226	190	168	87	33	11			
1983	3600	7757	9120	11002	9223	7458	6132	3875	2947	1369	1208	599	354	299	264	136	51	17			
1984	3642	7848	9227	11131	9331	7545	6204	3920	2981	1385	1222	606	358	302	267	138	52	17			
1985	2463	5307	6240	7527	6310	5102	4195	2651	2016	936	826	410	242	204	181	93	35	11			
1986	2267	4885	5743	6928	5808	4696	3862	2440	1856	862	761	377	223	188	166	86	32	10			
1987	2009	4329	5090	6140	5147	4162	3422	2162	1645	764	674	334	198	167	147	76	29	9			
1988	1860	4008	4712	5684	4765	3853	3168	2002	1523	707	624	309	183	154	137	70	26	9			
1989	2331	5023	5905	7124	5972	4829	3971	2509	1908	886	782	388	229	193	171	88	33	11			
1990	2314	4986	5862	7072	5929	4794	3942	2491	1894	880	776	385	228	192	170	88	33	11			
1991	2607	5617	6604	7967	6679	5401	4441	2806	2134	991	875	434	257	216	191	99	37	12			
1992	2595	5592	6574	7931	6648	5376	4420	2793	2124	987	871	432	255	215	190	98	37	12			
1993	3155	6798	7993	9642	8083	6536	5374	3396	2583	1199	1058	525	310	262	232	120	45	15			
1994	2626	5658	6653	8026	6728	5440	4473	2827	2150	998	881	437	258	218	193	100	37	12			
1995	2827	6092	7162	8640	7243	5856	4815	3043	2314	1075	948	470	278	235	208	107	40	13			
1996	2617	5639	6630																		
1997	3072	6619																			
1998	3409																				

Table 2.7 – Upper panel: (a) Observed and forecasted (in blue and green highlights) cumulative losses using CL *trapezoid* method; Lower panel: (b) Observed and forecasted (in blue and green highlights) incremental losses based on the cumulative losses.

using the trapezoidal CL method compared to 2671 thousand using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model. The trapezoidal CL method's heat map in Figure 2.4d also shows more deep red cells (underestimates) in the diagonals. Moreover, the trends of the red and grey lines in Figure 2.6 display clear lag-1 effects when adopting the development year persistence effect in the model. In summary, the GB2-$\beta\delta_l\gamma_1\gamma_2$ model provides more accurate in-sample estimates of calendar year loss than the trapezoidal CL method.

Table 2.8 – Calendar year reserve forecast for years 1996, 1997, and 1998.

Year	Models	mean	50%	55%	60%	65%	70%	75%	80%	85%	90%
1996	GB2-$\beta\delta_l\gamma_1\gamma_2$	70281	57932	62025	66448	71274	76806	83364	91662	103217	121429
	GG-$\beta\delta_l\gamma_1\gamma_2$	54537	47947	52442	57205	62331	67963	74283	81644	90740	102881
	trapezoidal CL	48078									
1997	GB2-$\beta\delta_l\gamma_1\gamma_2$	70241	50951	55968	61454	67616	74782	83397	94508	109798	133820
	GG-$\beta\delta_l\gamma_1\gamma_2$	51926	44886	49227	53795	58723	64194	70488	77804	86887	98950
	trapezoidal CL	48968									
1998	GB2-$\beta\delta_l\gamma_1\gamma_2$	70071	47224	52389	58117	64675	72306	81632	93755	110465	137198
	GG-$\beta\delta_l\gamma_1\gamma_2$	50704	43491	47651	52132	56989	62379	68538	75756	84714	96844
	trapezoidal CL	50461									

During the forecast period from 1995 to 1998, the red and grey lines continue to rise gradually in 1996 but start to stabilize in 1997 and 1998 except the quantiles higher than 80%. The blue line shows a mild upward trend for the trapezoidal CL method. It is worth noting that the quantile lines diverge due to the increase in forecast variability across the forecast period. The variability of calendar year reserve forecast should increase with calendar year k as the posterior predictive sample $\mathrm{y}^{(l)}_{T+m}$ in (2.38) depends on more posterior predictive samples $\mathbf{y}^{(l)}_{(T+1):(T+m-1)}$, adding more sampling variability to $\mathrm{y}^{(l)}_{T+m}$ and hence the calendar year reserve forecast $s^{(l)}_{t+k}$ in (2.39). Moreover, uncertainty increases in the later accident years with fewer observed losses and in the later development years with more noises or irregularities. These factors add to the variability of calendar year reserve forecasts. Therefore, it is anticipated that the flexible GB2 distribution will capture these heterogeneities and offer more reliable forecasts. To demonstrate its flexibility in producing more reliable forecasts, the properties of the GB2 distribution are examined in the following section. Additionally, an analysis of over and under-reserve is conducted, and experiments are carried out to validate the calendar year reserve forecasts.

2.6.3 Forecast performance studies

Flexibility of GB2 distribution

Heterogeneity is often encountered in insurance data. It frequently results in distributions with thick tails. GB2 distribution provides a flexible variance-to-mean ratio compared with the over-dispersed Poisson CL model with a constant variance-to-mean ratio across cells. Table 2.9 reports estimates of $\mu_{i,j}$, $\text{Var}(Y_{i,j})$ according to (2.17) and the variance to mean ratio $\text{Var}(Y_{i,j})/\mu_{i,j}$ across losses in the first forecasted calendar year 1996. The results indicate great variations in variance and the variance-to-mean ratio. The latter is closely associated with the mean level across development years. The findings suggest that the GB2 distribution permits a greater variance-to-mean ratio as the mean level increases, offering added flexibility for modeling loss data.

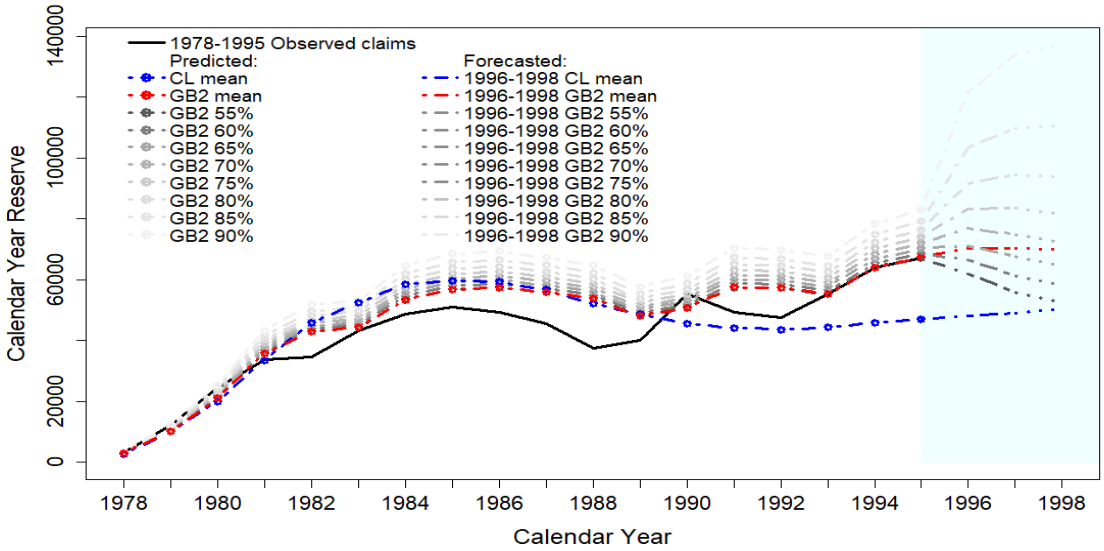


Figure 2.6 – Observed, predicted and forecasted of calendar year reserves at (a) the mean (red line) and quantiles (gray lines) using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model; and (b) the mean (blue line) using the CL *trapezoid* method

Additionally, a comparison is made between the flexibility of the GB2 distribution and the GG distribution within the family to resolve the disparity in calendar year reserve forecasts. This comparison is conducted using the GB2-$\beta\delta_l\gamma_1\gamma_2$ and GG-$\beta\delta_l\gamma_1\gamma_2$

Accident year	Develop year	$\mu_{i,j}$	$\text{Var}(Y_{i,j})$	$\text{Var}(Y_{i,j})/\mu_{i,j}$	Accident year	Develop year	$\mu_{i,j}$	$\text{Var}(Y_{i,j})$	$\text{Var}(Y_{i,j})/\mu_{i,j}$
19	**1**	2644	4,748,071	1795.8	**9**	**11**	1024	712,187	695.5
18	**2**	7150	34,722,130	4856.2	**8**	**12**	1219	1,009,257	827.9
17	**3**	8330	47,128,572	5657.7	**7**	**13**	622	262,770	422.5
16	**4**	10,926	81,080,547	7420.9	**6**	**14**	403	110,307	273.7
15	**5**	13,143	117,323,037	8926.7	**5**	**15**	593	238,838	402.8
14	**6**	8488	48,933,358	5765.0	**4**	**16**	79	4239	53.7
13	**7**	7823	41,566,259	5313.3	**3**	**17**	24	391	16.3
12	**8**	4081	11,311,689	2771.8	**2**	**18**	0.09	0.006	0.067
11	**9**	2368	3,808,533	1608.3	**1**	**19**	8	43	5.38
10	**10**	1559	1650770	1058.9					

Table 2.9 – Mean, variance and variance to mean ratio using GB2-$\beta\delta_l\gamma_1\gamma_2$ model for calendar year 1996.

models in Table 2.8. Cummins et al. (1990) stated that GB2 distribution arises from the structural $\text{GG}(y; a, \theta, p)$ distribution where the scale parameter θ is distributed as $\text{GG}(\theta; -a, b, q)$. The GB2 distribution, being a scale mixture of GG distributions, exhibits additional heterogeneity compared to the GG distribution. The limiting case when q tends to infinity signifies GG distribution corresponds to homogeneity. Refer to Appendix section 2.8(1), which explains the difference between the two distributions in terms of gamma-generating distributions. The three shape parameters a, p, and q for GB2 distribution are estimated to be 8.44, 0.22, and 0.34, respectively, whereas a and p for GG distribution are 4.20 and 0.33, respectively. Hence, part of the difference between the two distributions comes from q being 0.34 and ∞, respectively. Parameter a is lower in GG distribution to compensate for the thin tail from large q. To demonstrate this, a comparison is made between the logarithms of the densities in (2.19) and (2.22), which pertain to the two distributions at three quantiles for each loss in the calendar year 1996. Refer to Appendix section 2.8(2) for detailed calculations. Figure 2.7 plots the log densities at 50%, 75%, and 90% quantiles across development years. The plots show consistently higher log density for GB2 relative to GG distribution, indicating thicker tails for GB2 distributions. These results explain the lower forecasts using GG distributions since their tails are not thick enough to accommodate extreme losses. In summary, the tail property is an important consideration in choosing loss distribution.

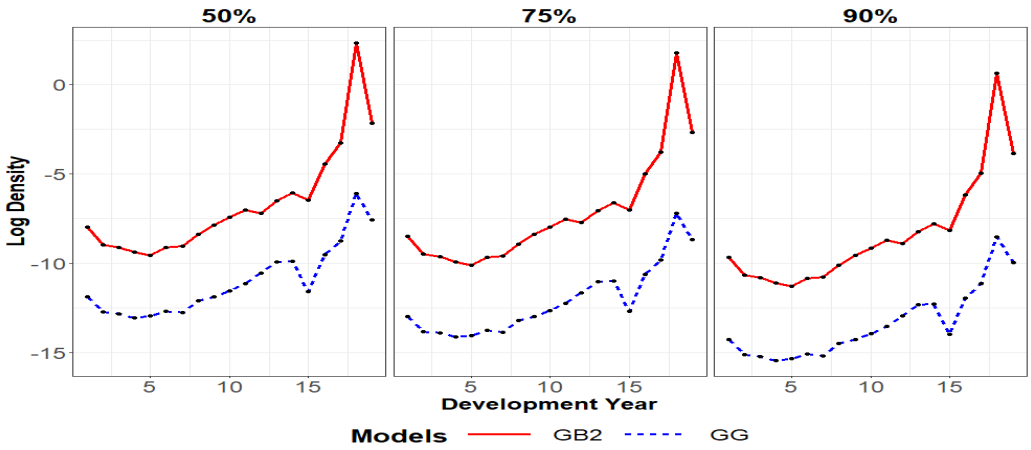


Figure 2.7 – Log density across development years for calendar year 1996 at 50%, 75% and 90% quantiles using GB2-$\beta\delta_l\gamma_1\gamma_2$ and GG-$\beta\delta_l\gamma_1\gamma_2$ models

Under and over loss reserve

Over and under reserves for the calendar year reserve forecasts have different implications for insurance companies. Over-reserved during prosperous financial years may not materially impact the company's current financial condition. However, it reduces the profit and leads to low surplus levels during lean years (Zhang and Browne, 2013). This incident is viewed in the insurance industry as a dangerous complacency. On the other hand, under-reserving may expose the company to the risk of a shortfall in outstanding liabilities and potential bankruptcy. The amount of over and under-estimations of individual losses are graphed in Figures 2.4a to 2.4e for various types of estimates.

McKenzie (2011) and De Myttenaere et al. (2016) proposed to measure the amount of over and under reserve using the mean absolute percentage error (MAPE) defined as

$$\text{MAPE} = \frac{1}{t}\sum_{k=1}^{t}\frac{|\hat{s}_k - s_k|}{s_k} \times 100 \tag{2.40}$$

where t denotes the current year and $\hat{s}_k$ is the estimate of s_k defined in (2.12). Anderson (1971) suggested that the percentage error for a loss reserve that falls within

Calendar Year	Observed	GB2						Chain Ladder
		Mean	55%	60%	65%	70%	75%	
1978	3323	3047^{-}	3075^{-}	3140	3208	3280	3360	2786^{-}
1979	12117	10284^{-}	10344^{-}	10554^{-}	10772^{-}	11002^{-}	11266^{-}	10242^{-}
1980	24591	21176^{-}	21252^{-}	21681^{-}	22141^{-}	22628^{-}	23174^{-}	20071^{-}
1981	33779	35731	35805	36549	37348	38197	39149^{+}	33601
1982	34609	42866^{+}	42945^{+}	43811^{+}	44729^{+}	45719^{+}	46820^{+}	45871^{+}
1983	43230	44341	44440	45282	46171	47138	48242	52412^{+}
1984	48628	53432	53538	54614	55766	56993^{+}	58402^{+}	58458^{+}
1985	51096	56765	56851	57961	59160^{+}	60458^{+}	61930^{+}	59702^{+}
1986	49387	57425^{+}	57504^{+}	58643^{+}	59855^{+}	61183^{+}	62661^{+}	59396^{+}
1987	45645	55949^{+}	56005^{+}	57098^{+}	58270^{+}	59556^{+}	60995^{+}	56643^{+}
1988	37486	53777^{+}	53841^{+}	54890^{+}	56000^{+}	57223^{+}	58598^{+}	52230^{+}
1989	40151	48060^{+}	48129^{+}	49052^{+}	50023^{+}	51106^{+}	52314^{+}	48742^{+}
1990	55304	50878^{-}	50938^{-}	51920^{-}	52969	54127	55426	45591^{-}
1991	49319	57517^{+}	57488^{+}	58717^{+}	60041^{+}	61488^{+}	63128^{+}	44161^{-}
1992	47583	57376^{+}	57348^{+}	58540^{+}	59813^{+}	61217^{+}	62784^{+}	43656^{-}
1993	55305	55312	55249	56419	57682	59064	60612	44413^{-}
1994	63846	64060	63951	65323	66791	68413	70233	45865^{-}
1995	67198	67287	67186	68668	70253	72008	73975	46972^{-}

		GB2		
		Mean	55%	65%
Training	**1978**	2917	2897	3005
	1979	10169	10126	10555
	1980	19597	19432	20272
	1981	32438	32134	33617
	1982	38937	38677	40378
	1983	39960	39717	41311
	1984	48795	48497	50492
	1985	53467	53205	55314
	1986	55239	54992	57126
	1987	55458	55236	57302
	1988	52539	52347	54317
	1989	47296	47123	48819
	1990	50844	50670	52506
	1991	63608	63384	65725
	1992	60936	60693	62869
Validation	**1993**	57968	50915	59009
	1994	65083	49616	61353
	1995	70120	48722	62171

MAPE in (%) and calendar year count t_g in bracket

MAPE (Overall)	13	13	15	16	18	20	20
MAPE_0 (acceptable: -5% to 15%)	4 (7)	4 (7)	6 (8)	7 (8)	7 (7)	7 (6)	1 (1)
MAPE_- (under-reserved: $< -5\%$)	11 (4)	11 (4)	10 (3)	11 (2)	9 (2)	6 (2)	18 (9)
MAPE_+ (over-reserved: $> 15\%$)	23 (7)	23 (7)	26 (7)	27 (8)	28 (9)	30 (10)	24 (8)

Table 2.10 – Top left: (a) Observed, estimated and forecasted calendar year loss using the mean and quantiles of the GB2-$\beta\delta_l\gamma_1\gamma_2$ model as well as using the trapezoidal CL method; Top right: (b) Estimated and forecasted calendar year loss using training data of years 1978-1992; Bottom left: MAPE in percentage with year count in bracket

five percent of the actual loss is reasonable from an actuarial point of view. An asymmetric criterion within the range of -5% to 15% is adopted, favoring the acceptance of over-reserve compared to under-reserve due to the higher risk posed by under-reserving to an insurance company. The assessment of reserves is based on three categories: over-reserve ($\hat{s}_k > 1.15s_k$), acceptable reserve ($0.95s_k \leq \hat{s}_k \leq 1.15s_k$), and under-reserve ($\hat{s}_k < 0.95s_k$). To quantify the level of deviation in these reserves, the MAPE measure in (2.40) is generalized to cover all three cases. The MAPE is defined as follows:

$$\text{MAPE}_g = \frac{1}{t_g}\sum_{k=1}^{t} I_{k,g}\frac{|\hat{s}_k - s_k|}{s_k} \times 100, \quad g = +, 0, - \tag{2.41}$$

where $I_{k,+} = I(\hat{s}_k > 1.15s_k)$, $I_{k,0} = I(0.95s_k \leq \hat{s}_k \leq 1.15\hat{s}_k)$, $I_{k,-} = I(\hat{s}_k < 0.95s_k)$ and the counts $t_g = \sum_{k=1}^{t} I_{k,g}$, $g = +, 0, -$.

The top left panel of Table 2.10 reports the calendar year reserve forecast s_k, indicated with '+' if it is an over-reserve and '−' if it is an under-reserve. The bottom left panel reports MAPE, MAPE_0, MAPE_-, MAPE_+, and the counts t_g in a bracket using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model and the trapezoidal CL method. It is reasonable to see the percentage of under-reserves (MAPE_-) drop and the percentage of over-reserves (MAPE_+) increase with quantile levels. Comparing the loss reserves using the mean of GB2-$\beta\delta_l\gamma_1\gamma_2$ model and the trapezoidal CL method, the former performs better by having lower MAPE, MAPE_-, and MAPE_+ with reasonable numbers of counts. Comparing the loss reserves using the mean and quantiles of the GB2-$\beta\delta_l\gamma_1\gamma_2$ model, the loss reserve using the mean estimates is marginally better than the reserve using the 55% quantile estimates. These 55% quantile estimates provide, in terms, the best loss reserve among other quantile estimates.

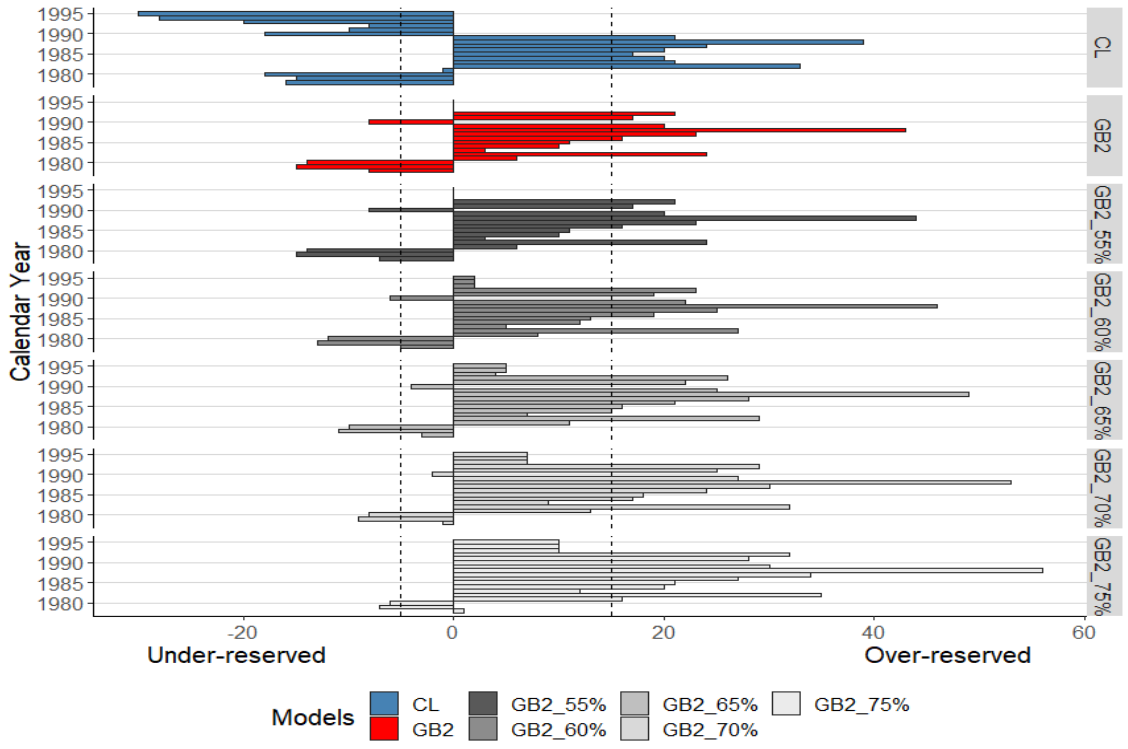


Figure 2.8 – Percentages of under and over reserves across types of estimators and calendar years

Figure 2.8 plots the percentages of under ($<-5\%$) and over-reserves ($>15\%$) indicated by two dash lines at -5% and 15%, respectively, for all estimators in all calendar years. The trapezoidal CL method has substantially more under-reserve than the reserves using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model. The amount of over-reserves is relatively similar to the reserves using the mean estimates of the GB2-$\beta\delta_l\gamma_1\gamma_2$ model. As the quantile level rises, it is clear that the amount of under-reserves drops at the cost of increasing over-reserves. Between the reserves using the mean estimates and 55% quantile estimates, the former is slightly better, as confirmed in Figure 2.8.

Validation using 1978-1992 losses

In order to assess the accuracy of these calendar year reserve forecasts, two experiments are conducted, commencing with the partitioning of the data into a training set with losses in 1978 to 1992 and a validation set with losses in 1993 to 1995. The GB2-$\beta\delta_l\gamma_1\gamma_2$ model is fitted to the training set, which comprises 15 years and $N = 120$ observations, to project losses for the three calendar years 1993, 1994, and 1995. Results of the first experiment using the loss data are reported in the right panel of Table 2.10. The model for the training data has DIC = 2185 and MSE = 3161 thousands. Figure 2.9 graphs the observed, estimated, and forecasted mean and quantiles for the calendar year losses or reserve forecasts. The calendar year reserve forecasts for 1993-1995 are very close to the observed calendar year losses. The amount of over or under-forecast for each loss is shown in the heat map of Figure 2.4f. There are a few over-forecasts in the earlier accident years and under-forecasts in the later accident years, so the sums of these loss forecasts in the diagonals approximate the observed sums. In conclusion, the GB2-$\beta\delta_l\gamma_1\gamma_2$ model provides reasonable calendar year reserve forecasts.

The second experiment continues from **Validating CARR model robustness** section with the 100 simulated data for Scenario I & II. The models are fitted using 15-year losses. Calendar year losses (1978-1992) and reserve forecasts (1993-1995; year 16-18) at the mean, 55% quantile, and 65% quantile are reported in Table 2.11. Scenario I with GB2-$\beta\delta_l$ fitted model has DIC = 1841 whereas Scenario II with GB2-

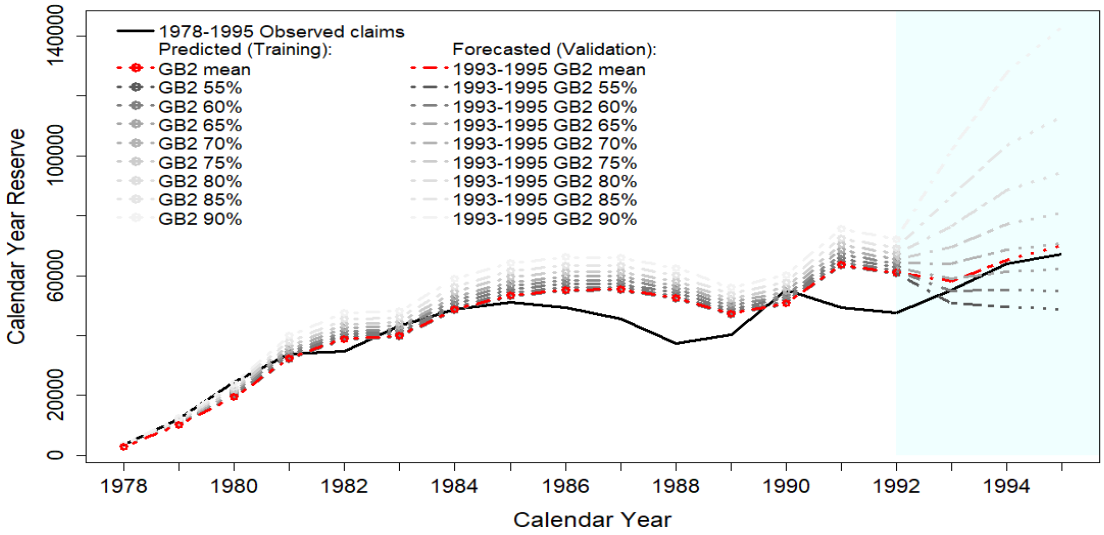


Figure 2.9 – Observed, predicted, and forecasted total losses at the mean (red line) and quantiles (gray lines) using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model. Data in 1978-1992 is for training, and data in 1993-1995 is for validation.

$\beta\delta_l\gamma_1\gamma_2$ fitted model has DIC = 1763. The MSE with three-year validation is better for scenario II using mean and 55% quantiles forecasts to assess the calendar year reserve forecast accuracy. In summary, results again confirm the relative advantage of the CARR model in providing calendar year reserve forecast even when fitting the model to data without persistence.

2.7 Conclusion

This chapter makes four contributions to loss reserving. Firstly, it introduces a novel approach to model changes in the development year pattern by incorporating the persistence of the development year effect using the CARR model. This component is integrated into the conventional ANOVA and ANCOVA models for accident year, development year, and calendar year effects. The observed losses indicate approximately linear for the accident and calendar year effects and quadratic for the development year effect, which is applied to the ANCOVA models.

	Calendar	GB2-$\beta\delta_l\gamma_1\gamma_2$	GB2-$\beta\delta_l$ Fitted			GB2-$\beta\delta_l$	GB2-$\beta\delta_l\gamma_1\gamma_2$ Fitted		
	Year	Simulated	Mean	55%	65%	Simulated	Mean	55%	65%
Training	1978	2807	2811	2838	2886	2575	2491	2511	2543
	1979	8873	9184	9243	9364	8360	8049	8097	8197
	1980	16625	17007	17096	17289	13993	14380	14456	14627
	1981	27380	26519	26639	26914	22196	21692	21796	22055
	1982	37016	34683	34829	35176	28723	28328	28460	28792
	1983	41379	41635	41807	42215	34289	33609	33766	34167
	1984	45984	47321	47517	47977	40202	39182	39361	39828
	1985	53336	51029	51243	51740	42065	43667	43870	44404
	1986	51979	53888	54117	54647	43990	45685	45897	46461
	1987	53084	55120	55356	55906	47117	46462	46679	47256
	1988	57093	56302	56549	57115	47031	49200	49432	50052
	1989	57487	57284	57539	58124	48326	48681	48911	49532
	1990	57839	57992	58255	58857	47646	49302	49535	50175
	1991	58677	58622	58890	59514	49576	49581	49816	50467
	1992	59020	59274	59550	60202	48804	51199	51443	52128
Validation	1993	60221	59920	59874	61988	50582	49433	49769	51938
	1994	64459	60556	60534	62678	49198	49310	49525	52262
	1995	65656	61187	61154	63321	49006	49653	49743	52833
MSE (×1000)	–	–	11765	11933	3915	–	584	4368	8624

Table 2.11 – Simulated calendar year losses (1978-1992), the mean, 55% quantile and 65% quantile calendar year reserve forecasts (1993-95) averaged over 100 simulated data sets with models trained using losses in 1978-1992 and MSEs with validation using losses in 1993-1995.

Secondly, this chapter extends the data distribution to GB2 and compares it with its two members, GG and Weibull distributions, and the distinct EE distribution. The implementation of the model is described in detail, utilizing a Bayesian approach via the user-friendly `Stan` package running in `R`. `Stan` commands for implementing the four distributions and performing the posterior simulation are provided. The simulated posterior samples provide mean and quantile loss estimates and forecasts.

Thirdly, the chapter demonstrates the way to forecast partial reserve and calendar year reserve using run-off triangle data. These losses are the sum of losses in the lower run-off triangle and the diagonals of the trapezoidal data below the run-off triangle, respectively. The application includes three calendar year reserve forecasts. Forecasts

of losses in the trapezoid are more complicated than the forecasts of losses in the lower run-off triangle only. Additional assumptions are introduced to enable the forecasts using the ANOVA models and trapezoidal CL method. To select the best forecasting model, 41 mean functions are considered, with and without the development year persistence effect, and they are applied to the four distributions. Results show that the GB2-$\beta\delta_l\gamma_1\gamma_2$ model is the best for both DIC model-fit and estimation accuracy using MSE. It also outperforms the GB2-CSR model and trapezoidal CL method.

Finally, a series of validation experiments are conducted to assess the reliability of the `Stan` package in model estimation, the robustness of the CARR model, and the accuracy of forecasts. These experiments include three validation tests, two based on simulation. The first experiment simulates run-off triangle losses using parameters in the best GB2-$\beta\delta_l\gamma_1\gamma_2$ model as true values and refits the simulated losses to the model. The MAD and CoV measures show that the parameter estimates achieve high levels of accuracy and credible interval coverage. The second experiment simulates losses from the CARR model (with development year persistence) and fits them to the non-CARR model and vice versa, resulting in two scenarios. Results confirm the robustness of the CARR model in fitting losses without development year persistence. The third experiment divides the observed and simulated losses (with the two scenarios) into training and validation (1993-1995). Results show that the GB2-$\beta\delta_l\gamma_1\gamma_2$ model using mean estimator forecasts calendar year reserves reasonably close to the observed calendar year losses in the validation period. With the simulated losses from the two scenarios, calendar year reserve forecasts from scenario II (simulating from the GB2-$\beta\delta_l$ model and fitting to the GB2-$\beta\delta_l\gamma_1\gamma_2$ model) are still relatively more accurate than the forecasts from scenario I.

In addition, a comparison is made between the amount of over and under calendar year reserve forecasts using the GB2-$\beta\delta_l\gamma_1\gamma_2$ model and CL method. Forecast errors are quantified using four MAPEs in (2.40) and (2.41), revealing that the GB2-$\beta\delta_l\gamma_1\gamma_2$ model with mean estimator delivers the best calendar year reserve forecasts with the lowest level of over and under-reserve. The 55% quantile estimator is the second best and outperforms the trapezoidal CL method. The study also encompasses an explo-

ration of several effect trends comparing distribution choices (GB2/GG/Weibull/EE), estimator types (mean/quantiles), and model classes (ANOVA/ANCOVA/CARR/C-SR/CL), etc, to detect essential features in the loss dynamic. In summary, it is confirmed that the GB2-$\beta\delta_l\gamma_1\gamma_2$ model with mean estimator stands out as the most reliable choice for calendar year reserve forecasting in this data set. The model offers a promising methodological advancement in loss reserve prediction and forecast.

From the forecasting perspective, ANCOVA and CARR models facilitate forecast through trend modeling. For the simple linear calendar year effect trend, the linear function in the ANCOVA model denoted by δ_l captures the calendar year effect well and hence is often selected in our data. However, the quadratic function in the ANCOVA model β_q does not describe the quadratic development year effect better than the more flexible ANOVA model β as shown in Table 2.2. Nevertheless, the ANOVA model has more parameters and is inapplicable beyond the observation period when computing forecasts. The generalized additive model may provide a better functional form to describe the development year pattern. However, the drawback is the substantial increase in regression parameters estimated from the limited observations in the run-off triangle. A compromise is introduced by shifting the ANOVA model to the ANCOVA model, incorporating a linear function in the later development years. This shift aims to create more parsimonious models that facilitate forecasting. Together with the CARR model, it is anticipated that the persistence terms will capture some of the development year effects, particularly those that are less prominent in later development years. Lastly, persistence terms can also be added to the calendar year or accident year effects if the data display such effects. These adjustments are particularly suitable for larger run-off triangles, and warrant further exploration.

2.8 Appendix

1- Both GB2 and GG are distributions generated from gamma distributions. Let Z_1 and Z_2 be gamma distributed random variables with scale parameter 1 and

shape parameter p and q, respectively. Then

$$b\left(\frac{Z_1}{Z_2}\right)^{1/a} \sim \text{GB2}(a,b,p,q) \text{ and } b\,(Z_1)^{1/a} \sim \text{GG}(a,b,p) \tag{2.42}$$

Thus, the tail of Z_1/Z_2 will be thinner (heavier) than that of Z_1 alone as the parameter q is large (near or less than one).

2- To calculate the log of GB2 density for any (i,j) cell at 90% quantile as an example can be computed using the commands 2.2 and applying some built-in R functions where the parameter estimates of a, p and q and b for the (i,j) cell are given by the posterior mean.

Code Listing 2.2 – Define log density of 90% quantile of GB2 distribution in R

```
u90=qbeta(0.90,p,q)
z90=ln(u90/(1-u90))
b=tgamma(p+1/a)*tgamma(q-1/a)/(mu*tgamma(p)*tgamma(q))
y90=b*exp(z90/a)
ld90=beta2.log(y90,a,mu,p,q)
```

where the command `beta2.log` is defined in the commands 2.1. For GG distribution, the commands to provide quantile and log density at the quantile are given below:

Code Listing 2.3 – Define log density of 90% quantile of GG distribution in R

```
y90=qggamma(0.90,alpha=a,scale=lam,psi=p);
ld90=dggamma(y90,alpha=a,scale=lam,psi=p,log=T);
```

where `lam` for the (i,j) cell is given by (2.23).

Chapter 3

Claim prediction and premium pricing for telematics auto-insurance data using Poisson regression with lasso regularization

This chapter considers a different data format for auto insurance. Telematics are increasingly popular devices for collecting individual driving data. These data will differentiate driving behaviors and provide tailor-made premium strategies. Due to this emerging granular data source, machine learning techniques, particularly lasso regression and deep learning neural networks, are developed in this and the next chapter to analyze the telematics data. This chapter is organized as follows: Section 3.1 reviews the literature on developing claim models using statistical and machine learning techniques. Section 3.2 describes the telematics data. Section 3.3 introduces GLMs, including two-stage threshold Poisson (TP), Poisson mixture (PM), and zero-inflated (ZIP) models, lasso regularization, and extensions. Section 3.4 provides empirical studies. Section 3.5 proposes a UBI premium pricing using predicted claims. Finally, Section 3.6 concludes the results.

3.1 Introduction

To evaluate the UBI premium, abundant driving information is first collected using telematics. Then, a wide range of driving behavior variables, called driving variables (DVs), which include, in general, four types of variables, namely driver-related, vehicle-related, driving habit, and driving behavior, can be created and regressed to insurance claims to reveal patterns between driving habits and driving risks. This is sometimes called knowledge discovery (Murphy, 2012). Guillen et al. (2021) suggested fitting Poisson regression models to traditional (driver/vehicle-related and driving habits) and critical incidents (risky driving behaviors) DVs. Results enable splitting an insurance premium into baseline premium and additional charges for *near-miss events*, defined as critical incidents such as hard braking, acceleration, and smartphone use while driving, that can potentially cause an accident. They suggested that these additional charges due to near-miss events can be updated weekly.

To quantify the possible non-linear effects of DVs on claim frequencies, Verbelen et al. (2018) used Poisson and negative binomial (NB) regression with generalized additive models (GAM) on demographic and telematic risk exposure DVs, such as total distance driven, yearly distance, number of trips, distance per trip and distance division according to road type (urban, other, motorways and abroad), time slot, and weekday/weekend. However, while these exposure-type DVs can act as offsets in regression models, they do not reveal true driving behavior. To enable a complete understanding of driving behavior, it is important to extract a large number of DVs that discriminate safety/risky driving and describe claim risk. For example, a simple braking event can be replaced by a complete 'braking story' that differentiates braking according to road characteristics (location, lanes, angles), braking style (abrupt, continuous, repeated, degree of harshness), braking time (time of day, day of week), road type (speed limit, normative speed), weather, previous action (turning, lane changing), and so forth. Moreover, extra environmental and traffic variables can be collected through GPS to enrich information.

However, as the number of DVs that describe driving behavior increases, the data

becomes voluminous, volatile, and noisy. Reducing these variables to a manageable number can reduce the computational cost and, more importantly, tackle the multicollinearity problem among these DVs. This problem is related to the substantial overlap in the predictive power of some of these DVs. For example, a driver living in an area with many traffic lights may use higher g-force brakes, or a senior driver may drive his car more during midday than standard rush hours or late at night. Hence, confusion may arise between location and harsh braking or between age and safer time slots. The consequence of multicollinearity is over-fitting and instability of predictive models.

Machine learning using statistical algorithms is compelling in tackling over-fitting and enhancing the stability and predictability of many predictive models. Machine learning algorithms are often categorized as being supervised or unsupervised. For unsupervised learning, Osafune et al. (2017) provided classifiers of safe and risky drivers based on the frequencies of acceleration, deceleration, and left acceleration using smartphone-equipped sensor data from 800+ drivers. Labeling drivers with at least 20 years of driving experience and no accident records as safe drivers and those with more than two accident records as risky drivers, they validated the classifier with 70% accuracy. Wüthrich (2017) introduced pattern recognition of the 2-dimensional velocity and acceleration (VA) heat maps via K-means clustering. However, neither study provides a claim prediction.

With claim risk information such as claim making, claim frequency, and claim size, supervised machine learning models embedded within GLMs can be constructed to unfold the hidden patterns in big data and predict future claims for premium pricing. Clustering, decision tree/random forest/gradient boosting, and neural networks are popular for machine learning techniques. Others include Poisson GAMs of Gao et al. (2019b) who incorporated 2-dimensional speed-acceleration heat maps in addition to some classical risk factors to predict claim frequencies. To reduce the size of components in the grid, they used the feature extraction techniques explored in Gao and Wüthrich (2018), namely, K-medoids clustering to group drivers with similar heatmaps and principal component analysis (PCA) to reduce the design matrix

dimension. Gao et al. (2019a) provided an in-depth predictive power study of more driving style and habit covariates using Poisson GAM. Makov and Weiss (2016) incorporated decision trees into Poisson prediction models for claim frequency.

To compare these machine learning techniques, Paefgen et al. (2013) found that neural networks outperform logistic regression and decision tree classifiers using 15 predictor variables on claim events. With claim frequencies classified as low, fair, and high, Weerasinghe and Wijegunasekara (2016) compared neural networks with decision trees and multinomial logistic regression models. They found that neural networks achieve the best predictive performance. However, logistic regression was suggested for model interpretability. Huang and Meng (2019) applied logistic regression for claim probability and Poisson regression for claim frequency embedded with support vector machine, random forest, advanced gradient boosting, and neural networks to 7 traditional variables and 30 DVs grouped according to travel habit, driving behavior, and critical incidents. They used stepwise feature selection and provided a survey of UBI pricing models with machine learning.

Another popular approach to tackle multicollinearity and overfitting is to regularize the loss function by penalizing the likelihood with the number of predictors. Ridge regression has merely the shrinkage property, but it does not select some optimal set of DVs that can best describe driving behaviors. Tibshirani (1996) proposed the least absolute shrinkage and selection operator (lasso) regression with L1 penalty for the predictors. After that, the lasso regression framework was extended to advance model fitting and variable selection processes. For example, Zou and Hastie (2005) proposed an elastic net that linearly combines the L1 and L2 penalties of the lasso and ridge methods, and Zou (2006) proposed the adaptive lasso where adaptive weights are used for penalizing different predictor coefficients in the L1 penalty. Furthermore, Park and Casella (2008) provided the Bayesian implementation of lasso regression as the lasso estimates can be interpreted as Bayesian posterior modes are estimates when the priors on the regression parameters are independent double-exponential (Laplace) distributions. Then Bayesian credible intervals of parameters can guide variable selection. Jeong and Valdez (2018) extended the Bayesian lasso framework

of Park and Casella (2008) using conjugate hyperprior distributional assumptions. This extension leads to a new penalty function called log-adjusted absolute deviation, which enables variable selection while maintaining the consistency of the estimator.

When modeling claim frequencies, one problem is the excessive number of zero claims that Poisson or negative binomial regression models may fail to capture. These zero claims do not necessarily mean no accidents during the terms of policies but rather no reported accidents by some bonus-hunting policyholders under a no-claim discount system. To identify factors for zero and non-zero claims (case-control), Winlaw et al. (2019) fitted a logistic regression model with lasso regularization to evaluate the impact of 24 travel behavior variables and found that speeding was the most critical driver behavior linked to crash risk. Alternatively, Tang et al. (2014) applied a ZIP regression model with EM adaptive lasso, combining the EM algorithm and adaptive lasso penalty to model the claim frequencies directly. However, they found that the variable selection result is not very good for the zero-inflation part. They suggested a possible reason is the lower signal-to-noise ratio in the zero-inflation part compared to the Poisson part. Banerjee et al. (2018) proposed a multicollinearity-adjusted adaptive lasso using zero-inflated count regression. They considered two data-adaptive weights: the inverse of the maximum likelihood estimates and the inverse estimates divided by their standard errors.

A generalization to the ZIP model allowing the degenerated zero component to be another Poisson component is the PM model. PM model serves dual purposes of modeling the number of claims and classifying safe and risky drivers. Although Park and Lord (2009) applied the PM model to analyze vehicle crash data as well as Chang and Kim (2012) applied the model in a Bayesian approach to identify accident-prone spots, they did not use lasso regularization. The novelty of this Chapter lies in implementing the PM model with lasso regularization, in particular, to avoid overfitting when analyzing the telematics data with many DVs. Refer to Table A1 of Eling and Kraft (2020) for a systematic summary of various modeling approaches in UBI.

Nevertheless, many of these studies utilized a limited number of DVs to describe a wide range of driver behaviors. Moreover, the overfitting issue, which can lead to unstable

results, was not thoroughly addressed in these studies. Literature on UBI predictive pricing models applying GLMs with machine learning, such as lasso regularization to tackle overfitting, is still relatively limited, particularly in modeling claim frequencies and addressing excessive zero claims and overdispersion due to heterogeneous driving behaviors.

This chapter aims to model the relationship between driving behaviors and claim frequencies to enhance claim prediction, select differential DVs, and classify drivers according to driving behaviors so that differential UBI premiums can be applied to safe and risky drivers. These UBI premiums should be updated regularly to provide continuous feedback to drivers. The contributions of this chapter are as follows:

1. **Predicting claim frequency using telematics data**: Utilizing a telematics dataset with 65 DVs and claim counts from a large number of drivers, this chapter employs three models to capture the effects of driving behavior on claim counts, namely, the two-stage TP, PM, and ZIP regression models. The aim is to identify separate DV sets that effectively differentiate safe and risky driving behaviors.

2. **Enhancing robustness and reducing overfitting**: It is important to ensure robustness and reduce overfitting in analyzing big telematics data. Hence, the above-mentioned three models will incorporate lasso regularization techniques, including adaptive lasso and elastic net. In addition, resampling and cross-validation techniques are employed, contributing to more stable and reliable results.

3. **Classifying drivers based on driving behavior**: The TP regression model classifies drivers in the first stage based on whether the annual predicted claims cross specific thresholds. In the second stage, lasso regularized Poisson regression models are refitted to each driver subgroup, allowing different sets of selected DVs in different subgroups. The PM models simultaneously estimate parameters and classify drivers, leveraging the capability of PM distributions

to capture overdispersion akin to NB distributions. Recognizing the high proportion of zeros in the data, the ZIP models include a structural zero part to capture a subgroup of drivers who do not claim. However, this group does not necessarily indicate safe drivers. Classification performance is evaluated using the area under the receiver operating characteristic (ROC) curve (AUC). It provides valuable insights into the classification effectiveness.

4. **Proposing experience-rating UBI premium pricing**: The chapter explores an experience-rating UBI premium pricing model, extending from the traditional approaches by incorporating classified claim groups (safe or risky) and predicted claims from the best-trained model. These advanced pricing models aim to incentivize safe driving, improve loss reserving practices, and enhance the accuracy of assessing the integrity of reported accidents.

3.2 Telematics data description

The dataset contains $J_0 = 65$ DVs constructed based on the information collected from telematics and GPS for $N = 14157$ drivers. The way to construct these DVs can be ad hoc; for example, event ranges (e.g., the morning rush is 6am to 9am) and binary bins for behavioral variables (e.g., lane change) are first defined. Then explanatory variables, such as counts of lane changing during rush hour when there are three or more lanes, etc., are created from the binary variables and event ranges. These DVs are aggregated over time to obtain certain incidence rates (per km or hour of driving) and scaled to normalize their ranges for better interpretability of their coefficients in the predictive models. These procedures transform the multi-dimensional longitudinal DVs into a single row for each driver, which is the unit of analysis.

The $J_0 = 65$ DVs are presented as column vectors $\boldsymbol{x}_{\bullet j}, j = 1, \ldots, J_0$ (exclude DVs 6, 11, 12, 17, 21, 30, 40, 41, 42, 48, 62 and 70 as they are repeats). The unit column vector $\boldsymbol{x}_{\bullet 0} = \mathbf{1}$ is included for fitting an intercept into various models. The

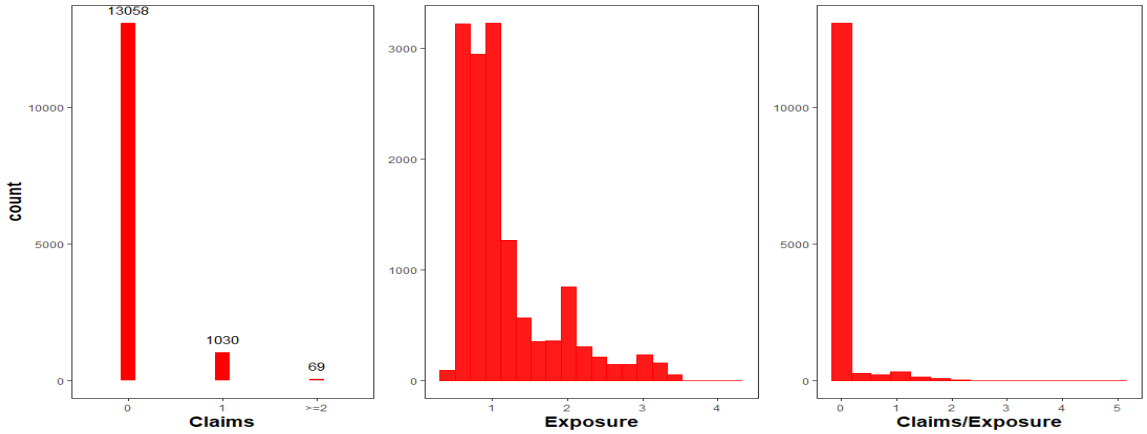


Figure 3.1 – Histogram of claims (left), exposure (mid), and claims per exposure (right)

data also contain two column vectors of claim counts $\boldsymbol{y} = y_{1:N}$ and policy duration or exposure $\boldsymbol{n} = n_{1:N}$ in the year. Ninety-two percent of $\boldsymbol{y}$ are zero. Figure 3.1 displays three histograms for $\boldsymbol{y}$, $\boldsymbol{n}$ and annual claims $\boldsymbol{a} = \{a_i = y_i/n_i, i = 1, \ldots, N\}$ respectively. Their average are $\bar{y} = 0.083, \bar{n} = 1.146$ and $\bar{a} = 0.075$, respectively. The variance of $y_{1:N}$ is 0.089, showing equidispersion, possibly due to the large proportion of zeros. Drivers with 0, 1, and at least 2 claims form three classes $C_b = \{i : y_i = b\}$, with sizes N_b, $b = 0, 1, 2^+$. The proportions $p_b = N_b/N$ are (0.92, 0.07, 0.005), averaged exposure $\bar{n}_b = \sum_{i \in C_b} n_i/N_b$ are (1.13, 1.38, 1.64) and averaged annual claim $\bar{a}_b = \sum_{i \in C_b} a_i/N_b$ are (0, 0.92, 1.71). Also $\bar{y}_{2^+} = \sum_{i \in C_{2^+}} y_i/N_{2^+} = 2.11$. The number of claims is not simply linearly related to exposure (R-squared is 0.014 regressing y_i on n_i), showing some possible impact of DVs $\boldsymbol{x}_{\bullet j}$ on y_i.

In a preliminary analysis, Figure 3.2 visualizes the DVs $j = 1, \ldots, J_0$ by plotting x_{ij} across driver i with colors indicating the number of claims $y_i = b$, $b = 0, 1, 2^+$. In the figure, 7 DVs (1, 2, 7, 8, 10, 14, 28) are sparse, with at most 13 non-zeros. The non-sparsity of DVs is measured by

$$S_j = \frac{\sum_i I(x_{ij} \neq 0)}{N} \times 100\%$$

where $I(E)$ is an indicator function of event E. Table 3.1 shows that these 7 DVs indeed have very low S_j. A refined measure that captures non-zero values to quantify

Figure 3.2 – Value against driver ID with colours showing claim frequency $y_i = 0, 1, \geq 2$ for 65 DVs

the information content of DVs is Shannon's entropy (Shannon, 2001) given by

$$H_j = -\sum_{l=1}^{n_j} P_j(v_{jl}) \log P_j(v_{jl})$$

where n_j denotes the number of distinct values $v_{jl}, l = 1, \ldots, \mathrm{n}_j$ in ascending order out of $\boldsymbol{x}_{\bullet j}$ for the j-th DV, and $P_j(\cdot)$ is the cumulative probability distribution (as a cumulative histogram) for $\boldsymbol{x}_{\bullet j}$ across v_{jl}. While H_j provides no information on the relationship with $\boldsymbol{y}$, the information gain (IG) evaluates the additional information the j-th DV provides about the claims $\boldsymbol{y}$ with three classes $C_b, b = 0, 1, 2^+$). It is defined as

$$\mathrm{IG}_j = H_j - \sum_{b \in \{0,1,2^+\}} p_b H_{jb},$$

where H_j is the entropy of the j-th DVs and H_{jb} is the entropy of the j-th DVs restricted to C_b.

Apart from studying the effect of DVs on $\boldsymbol{y}$, it is clear that multicollinearity between DVs affects the stability of a regression model. Figure 3.3 plots the correlation matrix of $\mathrm{Corr}(\boldsymbol{x}_{\bullet 1:J}, \boldsymbol{y}, \boldsymbol{a})$ and the correlations of 65 DVs with $\boldsymbol{y}$ are written by $\rho_j = \mathrm{Corr}(\boldsymbol{x}_{\bullet j}, \boldsymbol{y})$. The correlation matrix shows that DV up to 16 (except 9) are nearly uncorrelated with each other, the next up to DV 39 are mildly correlated, and the rest are moderately correlated, reflecting some pattern of these DVs. However, they are only weakly correlated with $\boldsymbol{y}$ and $\boldsymbol{a}$, showing low signal content of each DV in predicting $\boldsymbol{y}$ and $\boldsymbol{a}$.

Table 3.1 reports ρ_j, H_j, S_j, and IG_j to quantify the information content of the 65 DVs, flags a DV as "✗" when $H_j < 1$ and $S_j < 1\%$ ("✓" otherwise) and highlights IG_j in bold face when $\mathrm{IG}_j > 0$ indicating information gain. Asterisks are added to the DVs' ID to indicate the two levels of information content, in "✓" and boldface, respectively. Twenty DVs with "✗" are classified as having low information content. Including them in PM and ZIP models leads to unstable results. Subsequently, the analysis involves dropping them, resulting in $J_0' = 65 - 20 = 45$ DVs marked with "✓" in PM and ZIP models. However, all $J_0 = 65$ DVs are considered in the TP models. All DVs are normalized before analyses to ensure efficient modeling.

Figure 3.4 plots the Euclidean distance $d(j, j') = \sqrt{\sum_{i=1}^{N}(x_{ij} - x_{ij'})^2}$ between the (j, j') pair of DVs and performs the hierarchical clustering based on $d(j, j')$. Results show one major cluster of size 54 and two smaller clusters (49**, 36*, 43**, 72*, 56*,

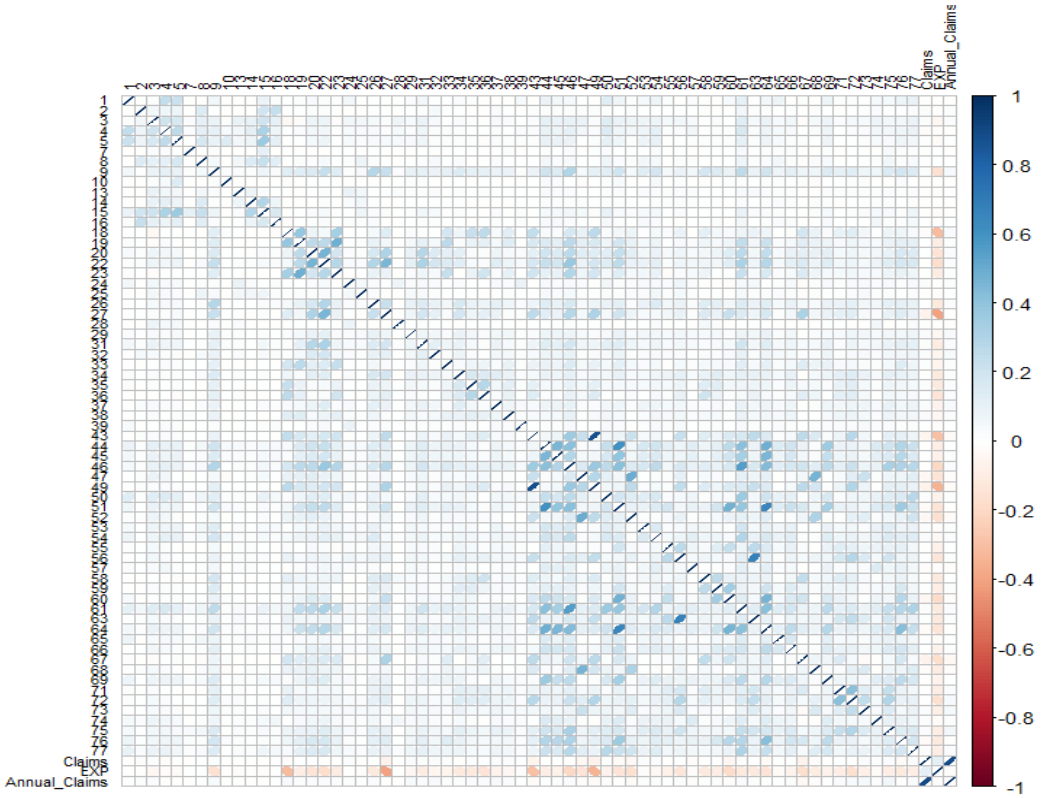


Figure 3.3 – Correlation matrix between 65 DVs, claims, exposure and annual claims

Table 3.1 – Identification of informative DVs. DVs with one asterisk have $H_j \geq 1$ and $S_j \geq 1\%$. DVs with two asterisks have $\text{IG}_j > 0$, indicating information gain.

DVs	ρ_j	H_j	$S_j(\%)$	IG_j	Flag	DVs	ρ_j	H_j	$S_j(\%)$	IG_j	Flag	DVs	ρ_j	H_j	$S_j(\%)$	IG_j	Flag	DVs	ρ_j	H_j	$S_j(\%)$	IG_j	Flag
1	-0.002	0.08	0.012	0	✗	**22***	00.003	89.45	12.991	0	✓	**39**	0.008	0.42	0.076	0	✗	**59***	0.002	45.89	8.312	0	✓
2	0.012	0.02	0.002	0	✗	**23***	-0.001	87.75	12.871	0	✓	**43****	**-0.053**	**99.99**	**13.789**	**0.002**	✓	**60****	**-0.041**	**99.93**	**13.787**	**0.001**	✓
3*	0.018	7.04	1.231	0	✓	**24**	0.017	1.02	0.192	0	✗	**44***	-0.004	19.69	3.795	0	✓	**61***	0.018	35.71	6.472	0	✓
4	-0.011	1.91	0.310	0	✗	**25**	-0.002	0.30	0.046	0	✗	**45***	0.002	18.61	3.678	0	✓	**63***	0.006	32.91	6.462	0	✓
5	0.003	0.79	0.119	0	✗	**26***	0.005	17.50	3.990	0	✓	**46***	-0.003	90.51	13.008	0	✓	**64***	-0.021	61.78	10.149	0	✓
7	-0.002	0.01	0.001	0	✗	**27****	**-0.060**	**99.69**	**13.773**	**0.002**	✓	**47***	-0.035	92.68	13.246	0	✓	**65**	0.0005	1.22	0.295	0	✗
8	0.004	0.10	0.014	0	✗	**28**	-0.004	0.03	0.003	0	✗	**49****	**-0.061**	**99.98**	**13.789**	**0.002**	✓	**66***	0.008	4.41	1.339	0	✓
9*	0.010	28.69	5.666	0	✓	**29***	0.023	4.41	1.288	0	✓	**50***	0.012	6.65	1.247	0	✓	**67****	**-0.060**	**99.54**	**13.766**	**0.002**	✓
10	-0.002	0.01	0.001	0	✗	**31***	0.014	15.93	3.698	0	✓	**51***	-0.025	67.41	10.718	0	✓	**68***	-0.019	76.17	11.953	0	✓
13	-0.003	0.45	0.069	0	✗	**32**	0247	4.41	1.229	0	✗	**52***	-0.039	94.18	13.357	0	✓	**69***	-0.007	7.83	1.895	0	✓
14	-0.006	0.06	0.009	0	✗	**33***	0.011	39.01	7.957	0	✓	**53**	0.015	3.00	0.645	0	✗	**71***	0.006	32.11	6.585	0	✓
15	-0.0001	0.50	0.076	0	✗	**34***	-0.001	21.44	5.114	0	✓	**54***	0.023	5.03	1.161	0	✓	**72***	0.007	41.24	7.861	0	✓
16	0.006	0.24	0.036	0	✗	**35***	0.010	35.54	7.257	0	✓	**55***	-0.002	21.21	4.424	0	✓	**73***	0.023	11.09	2.775	0	✓
18**	**-0.010**	**99.90**	**13.785**	**0.001**	✓	**36***	0.009	54.80	9.701	0	✓	**56***	0.001	34.25	6.654	0	✓	**74**	0.013	1.03	0.222	0	✗
19*	0.003	77.88	12.129	0	✓	**37***	0.024	2.40	0.856	0	✓	**57**	-0.012	1.23	0.229	0	✗	**75***	0.029	10.85	2.669	0	✓
20*	0.006	67.10	10.980	0	✓	**38***	0.022	3.84	1.355	0	✓	**58***	-0.008	61.74	10.378	0	✓	**76***	-0.021	35.50	7.043	0	✓
																		77*	0.011	13.61	3.354	0	✓

63*) and (55*, 59*, 71*, 27**, 58*) respectively with increasing pairwise distance from the major cluster. All spare DVs labeled as noninformative are in the major cluster. These DV features may guide our interpretation of the selected DVs in subsequent

analyses.

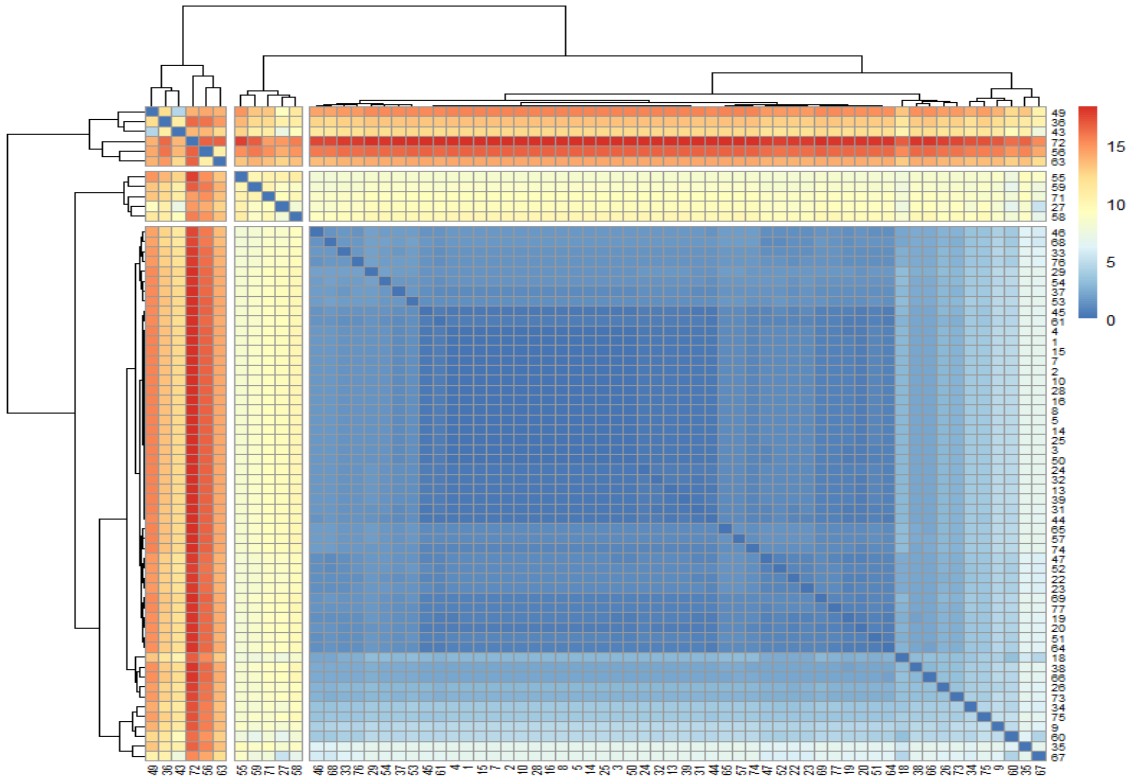


Figure 3.4 – Distance score and hierarchical clustering between 65 DVs

3.3 Methodologies

The predictive models for the claim rate of any driver using GLMs with lasso regularization is derived when the true model is assumed to have a sparse representation of 65 DVs. This section also consider some model performance measures, including AUC.

3.3.1 Regression models

Poisson regression model

The Poisson regression model is commonly applied to count data, like claims. It is defined as

$$Y_i \sim \text{Poisson}(\mu_i), \quad \mu_i = n_i a_i = n_i \exp(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta}) = \exp\Big(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta} + \log(n_i)\Big) \tag{3.1}$$

where $\boldsymbol{Y} = Y_{1:N}$, $\boldsymbol{\beta} = \beta_{0:J_1}$, J_1 is the number of selected DVs in the model, $a_i = \exp(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta})$ estimates the number of claim per year for driver i, and $\log(n_i)$ is the offset parameter in a regression model. Poisson regression assumes equidispersion. For overdispersed data, negative binomial (NB) distribution, such as the Poisson-Gamma mixture, provides extra dispersion. With NB regression, the term "Poisson(μ_i)" in (3.1) is replaced with NB distribution $\text{NB}(\nu, q_i) = \text{NB}\Big(\nu, \mu_i/(\nu - \mu_i)\Big)$ where ν is the shape parameter and q_i is the success probability of each trial. NB distribution converges to Poisson distribution if ν tends to infinity.

Poisson mixture model

The finite mixture Poisson model is another popular model for modeling unobserved heterogeneity. The model assumes G unobserved groups each with probability π_g, $0 < \pi_g < 1, g = 1, \ldots, G$ and $\sum_{g=1}^{G} \pi_g = 1$. Assuming that each driver claim Y_i from the driver group g follows a Poisson distribution with mean μ_{ig}, that is, $Y_i \sim \text{Poisson}(\mu_{ig})$ at probability $\pi_g,\ g = 1, \ldots, G$, the observed data likelihood function is

$$\mathcal{L}_{\text{o}}(\boldsymbol{\theta}) = \prod_{i=1}^{N} \sum_{g=1}^{G} \pi_g f_g(y_i|\mu_{ig}) \tag{3.2}$$

where $f_g(y_i; \mu_{i|g})$ is the density of Poisson distribution with mean function $\mu_{ig} = \exp\Big(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta}_g + \log(n_i)\Big)$, $\boldsymbol{\beta}_g = (\beta_{0:J_1,g})^\top$, and $\boldsymbol{\theta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_G^\top, \pi_1, \ldots, \pi_{G-1})^\top$ is the model parameter vector. For the two group mixture with $G = 2$, the summation

term in (3.2) for driver i becomes

$$f_{\text{PM}}(y_i|\pi, \mu_{i1}, \mu_{i2}) = \pi f_1(y_i|\mu_{i1}) + (1-\pi) f_2(y_i|\mu_{i2}) \tag{3.3}$$

where $\pi = \pi_1$ and $f_g(\cdot|\mu_{ig}), g = 1, 2$ is the probability mass function of Poisson distribution with mean μ_g. The marginal predicted claim is

$$\hat{y}_i = \hat{\mathfrak{z}}_{i1}\mu_{i1} + (1 - \hat{\mathfrak{z}}_{i1})\mu_{i2} \tag{3.4}$$

where μ_{ig} is the predicted claim condition on group g. The expectation-maximisation (EM) algorithm often estimates parameters $\boldsymbol{\theta}$. In the M-step, $\boldsymbol{\theta}$ are estimated by maximising the complete-data likelihood given by

$$\mathcal{L}_{\text{c}}(\boldsymbol{\theta}) = \prod_{i=1}^{N}\prod_{g=1}^{G}[\pi_g f_g(y_i|\mu_{ig})]^{\mathfrak{z}_{ig}} \tag{3.5}$$

where the posterior group membership for driver i is estimated by

$$\mathfrak{z}_{ig} = \frac{\pi_g f_g(y_i|\mu_{ig})}{\sum_{g'} \pi_{g'} f_{g'}(y_i|\mu_{ig'})} \tag{3.6}$$

in the E-step. Besides, the Bayesian method with MCMC sampling is also popular in estimating mixture models.

If there is a high proportion of zero claims, the ZIP model (Lambert, 1992) may be suitable to capture the excessive zeros. The model is a special case of a two-group mixture model that combines a zero-point mass in group 1 with a Poisson distribution in group 2. The zeros may come from a Poisson distribution's point mass (structural zero) or the zero count (natural zero). The model is given by

$$\Pr(Y_i = 0) = \pi_i + (1-\pi_i)\exp(-\mu_i), \text{ and } \Pr(Y_i = y_i) = (1-\pi_i)\frac{\mu_i^{y_i}\exp(-\mu_i)}{y_i!}, \; y_i \geq 1 \tag{3.7}$$

where the two regression models (called zero and count) for the probability π_i of structural zero and the expected counts (including non-structural zero) μ_i, respectively,

are given by

$$\pi_i = \frac{\exp\left(\boldsymbol{x}_{i\bullet}\boldsymbol{\psi} + \log(n_i)\right)}{1 + \exp\left(\boldsymbol{x}_{i\bullet}\boldsymbol{\psi} + \log(n_i)\right)}, \text{ and } \mu_i = \exp\left(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta} + \log(n_i)\right). \tag{3.8}$$

The logistic regression parameters $\boldsymbol{\psi} = (\psi_0, \ldots, \psi_{J_\psi})^\top$ is a vector of J_ψ selected DVs to estimate the probability of extra zero. The vector of model parameters is $\boldsymbol{\theta} = (\boldsymbol{\psi}^\top, \boldsymbol{\beta}^\top, \pi)^\top$.

3.3.2 Regularization techniques

A stepwise procedure is often used to search for a good subset of DVs. However, it often suffers from high variability, local optimum, and ignorance of uncertainty in the searching procedures (Fan and Li, 2001). Lasso regularization offers an alternative approach to select variables for parsimonious models. It is further extended to adaptive lasso and elastic net.

Lasso

The lasso (L) regularization with the L1 penalty provides a simple way to enforce sparsity in variable selection by shrinking some coefficients β_j to zero. This aligns with our aim to select important DVs, that is, those with coefficients that have not shrunk to zero. Essentially, the lasso estimates $\boldsymbol{\beta}_{\lambda,\mathrm{L}}$ with a given penalty $\lambda > 0$ minimize the penalized least square loss function

$$\mathrm{SSR}_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{N}\left[y_i - \exp\left(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta} + \log(n_i)\right)\right]^2 + \lambda\sum_{j=1}^{J}|\beta_j| \tag{3.9}$$

where the first term (denoted by SSE_0) has an order $p = 2$, the penalty function $\varrho(\boldsymbol{\beta}) = \sum_{j=1}^{J}|\beta_j|$ has an order $\alpha = 1$ for lasso regression, and λ is the weight of the penalty function $\rho(\boldsymbol{\beta})$. The parameter estimates which minimize (3.9) condition on

λ are given by

$$\hat{\beta}_{j,\lambda} = \beta_{j,\text{LS}} \max\left(0, 1 - \frac{N\lambda}{|\beta_{j,\text{LS}}|}\right) \tag{3.10}$$

where $\boldsymbol{\beta}_{\text{LS}} = (\beta_{1,\text{LS}}, \ldots, \beta_{J,\text{LS}})$ minimize SSE_0 when $\lambda = 0$.

As λ increases, the term $1 - \frac{N\lambda}{|\beta_{j,\text{LS}}|}$ becomes negative and so $\hat{\beta}_{j,\lambda}$ will shrink to zero. Then, the term $\lambda \sum_{j=1}^{J} |\hat{\beta}_{j,\lambda}|$ in (3.9) will drop as more $\hat{\beta}_{j,\lambda} = 0$ but SSE_0 increases as $\boldsymbol{\beta}_\lambda$ get further away from $\boldsymbol{\beta}_{\text{LS}}$ so that one can choose a $\lambda_{\min}$ that minimizes (3.9) to obtain $\boldsymbol{\beta}_{\lambda,\text{L}}$. Alternatively, one can perform a K-fold CV and choose $\lambda_{\min}$ that provides the best overall model fit for all K validated samples. Apart from SSE_0, other model fit criteria, including the sum of absolute errors (SAE) and deviance $\mathcal{D} = -2\sum_{i=1}^{N} \log f(y_i; \mu_i)$, can replace SSE_0 in (3.9). Different criterion may suggest different optimal $\lambda_{\min}$ and hence the estimates $\boldsymbol{\beta}_{\lambda_{\min}}$. However, Meinshausen and Bühlmann (2006) showed the conflict of optimal prediction and consistent variable selection in lasso regression. Moreover, whether lasso regression has an oracle procedure is debatable. An estimating procedure is an oracle if it can identify the right subset of variables and has an optimal estimation rate so that estimates are unbiased and asymptotically normal.

Adaptive lasso

Zou (2006) introduced adaptive lasso (A) that deals with three issues, namely, inconsistent selection of coefficients, lack of oracle property, and unstable parameter estimate when working with high dimensional data. Städler et al. (2010) also proclaimed these issues and addressed some bias problems of the (one-stage) lasso, which may shrink important variables too strongly. In addition, the L1 penalty method has a collinearity problem in high-dimensional data, severely degrading the performance of lasso (Zou and Zhang, 2009). The two-stage adaptive L1 penalty is a modification of lasso when each coefficient β_j is given its weight $\mathfrak{w}_j$. The adaptive lasso coefficients

$\boldsymbol{\beta}_{\lambda,\boldsymbol{w},\mathrm{A}}$ are estimated by minimizing the loss function

$$\mathrm{SSR}_{\lambda,\mathfrak{w}}(\boldsymbol{\beta}) = \sum_{i=1}^{N}\Big[y_i - \exp\Big(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta} + \log(n_i)\Big)\Big]^2 + \lambda\sum_{j=1}^{J}\mathfrak{w}_j|\beta_j| \tag{3.11}$$

where $\mathfrak{w} = (\mathfrak{w}_1, \ldots, \mathfrak{w}_J)$ are the adaptive data-driven weights. The weights control the rate as each coefficient is shrunk towards 0. It is clear that smaller coefficients will leave the model before larger coefficients. Hence, adaptive lasso penalizes more those coefficients with lower initial estimates $\hat{\beta}_{j,\mathrm{R}}$.

Zou (2006) recommended that the weights $\mathfrak{w}_j = |\hat{\beta}_{j,\mathrm{R}}|^{-\gamma}$ where $\hat{\beta}_{j,\mathrm{R}}$ is an initial estimate from ridge regression, and $\gamma > 0$ is a tuning parameter to ensure that the adaptive lasso has oracle properties. The weights are rescaled so that their sum equals the number of DVs inputted. By assigning a higher penalty to small coefficients and vice verse, adaptive lasso can consistently select the true model and provide unbiased estimates (Courtois et al., 2021). Städler et al. (2010) suggested the tuning parameter $\gamma = 1$ for low claim threshold and $\gamma = 2$ for high claim threshold. Tuning parameter $\gamma = 2$ is adopted as the best choice to estimate weights $\mathfrak{w}_j$ in the subsequent adaptive lasso models. Compared with $\gamma = 0.5$ and 1, $\gamma = 2$ tends to give more differential weights $\mathfrak{w}_j$ when penalizing β_j to minimize $\mathrm{SSR}_{\lambda,\mathfrak{w}}$ in (3.11) and obtain the adaptive lasso estimates $\boldsymbol{\beta}_{\lambda,\mathfrak{w},\mathrm{A}}$.

Elastic-net

Zou and Zhang (2009) argued that the L1 penalty can perform poorly when multi-collinearity problems are common in high-dimensional data. They proposed elastic-net (E), which takes a weighted average of two penalties: ridge ($\alpha = 0$) and lasso ($\alpha = 1$). The elastic-net coefficients $\boldsymbol{\beta}_{\lambda,\alpha,\mathrm{E}}$ are estimated by minimising the loss function

$$\mathrm{SSR}_{\lambda,\alpha}(\boldsymbol{\beta}) = \sum_{i=1}^{N}\Big[y_i - \exp\Big(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta} + \log(n_i)\Big)\Big]^2 + \lambda\Big[\frac{1-\alpha}{2}\sum_{j=1}^{J}{\beta_j}^2 + \alpha\sum_{j=1}^{J}|\beta_j|\Big]. \tag{3.12}$$

Adaptive elastic-net

Adaptive elastic-net (N) combines adaptive lasso with elastic-net regularization to estimate coefficients $\boldsymbol{\beta}_{\lambda,\mathfrak{w},\alpha,\mathrm{N}}$ by minimizing the loss function

$$\mathrm{SSR}_{\lambda,\alpha,\mathfrak{w}}(\boldsymbol{\beta}) = \sum_{i=1}^{N}\Big[y_i - \exp\Big(\boldsymbol{x}_{i\bullet}\boldsymbol{\beta} + \log(n_i)\Big)\Big]^2 + \lambda\Big[\frac{1-\alpha}{2}\sum_{j=1}^{J}\mathfrak{w}_j{\beta_j}^2 + \alpha\sum_{j=1}^{J}\mathfrak{w}_j|\beta_j|\Big]. \tag{3.13}$$

which combine (3.11) and (3.12). When implementing the elastic-net regularization, the mixing parameter $\alpha \in (0,1)$ balances the two penalties with $\alpha > 1/3$, indicating a heavier lasso penalty. However, the optimal α needs to be searched over to identify the best α with the lowest root MSE (RMSE), MSE, or R-squared.

Remark: To implement these regularization techniques to the mixture models in Section 3.3.1 using the EM approach, the penalized log-likelihood is applied (Bhattacharya and McNicholas, 2014; Banerjee et al., 2018). For the case of adaptive elastic-net regularization, the penalized log-likelihood is given by

$$\ell_{\lambda,\alpha,\mathfrak{w}}(\boldsymbol{\theta}) = -\log\mathcal{L}_{\mathrm{c}}(\boldsymbol{\theta}) + \lambda_1\Big[\frac{1-\alpha}{2}\sum_{j=1}^{J}{\beta_{j1}}^2 + \alpha\sum_{j=1}^{J}\mathfrak{w}_{j1}|\beta_{j1}|\Big] + \lambda_2\Big[\frac{1-\alpha}{2}\sum_{j=1}^{J}{\beta_{j2}}^2 + \alpha\sum_{j=1}^{J}\mathfrak{w}_{j2}|\beta_{j2}|\Big] \tag{3.14}$$

where $\mathcal{L}_c(\boldsymbol{\theta})$ is given by (3.5) and $\mathfrak{w}_{j1}$ and $\mathfrak{w}_{j2}$ are the data-driven adaptive weights. Refer to Appendix section 3.7.1(1) for implementing all lasso regularization procedures under Poisson regression.

The search for α in (3.13) and (3.14) is conducted for different models summarized in Table 3.2. For example, model TPAL-2 refers to a stage 2 TP model when adaptive lasso (A) regularization is applied in stage 1 and lasso (L) is applied in stage 2. Different stage 1 TP, stage 2 (under TPL-1 and TPA-1) TP with threshold τ (to split predicted a_i into low and high groups), PM and ZIP models are considered. Each model is executed for five α values (0.100, 0.325, 0.550, 0.775, 1.000), and the optimal α, yielding the lowest RMSE with $K = 10$ folds CV, is identified. To ensure robustness in the search, the procedure is repeated $R = 100$ times for each model using R=100 70% subsamples $\mathcal{S}_{1:R}$. The proportions of times out of R subsamples

that each α is the best according to RMSE are reported in Table 3.3. Results show that low $\alpha = 0.1$ should be adopted for stage 1 TP models, the low group of most stage 2 TP models, PME model, and PMN model, whereas higher $\alpha = 0.775, 1$ for the higher group of stage 2 TP models. Refer to Appendix section 3.7.1(2) for the implementation details using `caret` package in `R`.

Table 3.2 – Model names for TP, PM, and ZIP models with different lasso regularization

Stage 1 threshold Poisson		**Stage 2 threshold Poisson**			**Poisson mixture**		**Zero-inflated**	
TPL-1	Lasso	**TPLL-2**	**TPAL-2**	Lasso	**PML**	Lasso	**ZIPL**	Lasso
TPE-1	Elastic-net	**TPLE-2**	**TPAE-2**	Elastic-net	**PME**	Elastic-net		
TPA-1	Adaptive lasso	**TPLA-2**	**TPAA-2**	Adaptive lasso	**PMA**	Adaptive lasso	**ZIPA**	Adaptive lasso
TPN-1	Adaptive elastic-net	**TPLN-2**	**TPAN-2**	Adaptive elastic-net	**PMN**	Adaptive elastic-net		

Table 3.3 – Proportions of times out of $R = 100$ subsamples $\mathcal{S}_{1:R}$ with 70% subsampling rate that favors each of the five α in the grid search for optimum α in elastic net regularization for stage 1 TP, stage 2 TP, and PM models. The optimal choices of α according to RMSE with $K = 10$ fold CV are indicated with "*" for the highest proportions and used in the corresponding elastic net model.

		α	**0.100**	**0.325**	**0.550**	**0.775**	**1.000**	α	**0.100**	**0.325**	**0.550**	**0.775**	**1.000**
		Stage 1 threshold Poisson model											
		TPE-1	0.38*	0.32	0.28	0.02	-	**TPN-1**	0.35*	0.26	0.39	-	-
		Stage 2 threshold Poisson model											
$\tau_{0.08}$	**Low**	**TPLE-1**	0.04	0.73*	0.23	-	-	**TPAE-1**	0.38	0.51*	0.11	-	-
	High		-	0.02	0.30	0.58*	0.10		0.01	0.04	0.17	0.67*	0.11
	Low	**TPLN-1**	0.02	0.78*	0.19	-	-	**TPAN-1**	0.35	0.52*	0.13	-	-
	High		-	0.04	0.34*	0.53	0.09		-	0.04	0.19	0.61*	0.16
$\tau_{0.09}$	**Low**	**TPLE-1**	0.79*	0.19	0.02	-	-	**TPAE-1**	0.76*	0.19	0.05	-	-
	High		-	-	0.06	0.40	0.54*		-	0.04	0.25	0.46*	0.25
	Low	**TPLN-1**	0.42*	0.36	0.22	-	-	**TPAN-1**	0.74*	0.2	0.06	-	
	High		-	-	0.22	0.57*	0.21		-	0.06	0.19	0.49*	0.26
$\tau_{0.10}$	**Low**	**TPLE-1**	0.43*	0.42	0.15	-	-	**TPAE-1**	0.87*	0.1	0.03	-	-
	High		-	0.01	0.22	0.5*	0.27		-	-	0.17	0.51*	0.32
	Low	**TPLN-1**	0.84*	0.14	0.02	-	-	**TPAN-1**	0.86*	0.09	0.05	-	-
	High		-	-	0.06	0.36	0.58*		-	0.01	0.14	0.54*	0.31
$\tau_{0.11}$	**Low**	**TPLE-1**	0.89*	0.09	0.02	-	-	**TPAE-1**	1.00*	-	-	-	-
	High		-	-	0.07	0.17	0.76*		-	0.01	0.15	0.37	0.47*
	Low	**TPLN-1**	0.84*	0.14	0.02	-	-	**TPAN-1**	1.00*	-	-	-	
	High		-	-	0.06	0.36	0.58*		-	0.01	0.10	0.40	0.49*
		Poisson mixture											
		PME	0.92*	0.08	-	-	-	**PMN**	0.92*	0.08	-	-	-

3.3.3 Model performance measures

Model performance can be evaluated based on the aims and model assumptions. The goodness of model fit, prediction accuracy, and driver classification are the main criteria linked to different metrics.

1. *Bayesian information criterion* (BIC) is a popular model fit measure that takes into account the model assumptions by the deviance term -2 times log-likelihood and adds a parameter penalty term, which uses the log of sample size as weight. *Akaike information criterion* (AIC) can also be used when the parameter penalty term uses 2 as the weight. Some packages also use deviance (without parameter penalty) to select models. Models with lower values for either measure are better.

2. For prediction accuracy, the popular *mean square error* (MSE) is adopted for comparing predicted with observed claims, defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mu_i)^2$$

   Some packages also report a *mean absolute error* (MAE) for robustness consideration. As the number of claims is capped by 5, the claim data has no extreme outliers. For either measure, lower values indicate better prediction accuracy. The goal is to provide accurately predicted claims as risk scores in the premium calculation; MSE is used as an important measure and assigns double weight.

3. *Correlation* ρ reveals dependency between observed and predicted annual claims (instead of claims in MSE). A higher correlation shows better performance.

4. To quantify classification performance, the difference between observed group membership and predicted group membership should be quantified. However, the group membership of each driver is not observed, so it is approximated by the K-mean clustering using elected DVs. With this estimated binary response vector, AUC for ROC curve (Fawcett, 2006) visualizes classification performance

by constructing confusion matrix conditions on the classifier (e.g., estimated annual claim a_i in (3.1) for TP-2 and safe group probabilities $\mathfrak{z}_{i1}$ in (3.6) for PM) cutoff. For each confusion matrix, the true positive rate (TPR) (sensitivity) and the false positive rate (FPR) (1-specificity) can be calculated, and the ROC curve plots the TPR against the FPR as the discrimination cutoff for the classifier varies. TPR (FPR) is expected to be as large (small) as possible. If TPR = 1 can be achieved by a smaller cutoff, such a binary classifier is more efficient. It has a greater AUC, which is the probability that a randomly chosen member of the positive class has a lower estimated probability of belonging to the negative class than a randomly chosen member of the negative class. Refer to Appendix section 3.7.1(5) for the detailed implementation.

To distinguish between low claim (safe driver) and high claim (risky driver) groups, a binary classifier is employed. The "observed" groups are estimated by K-mean clustering using the DVs selected by each model, and they serve as proxies to the true groups. The K-mean clustering minimizes the total within-cluster variation for any cluster $\mathcal{C}_g$ defined as the sum of squared Euclidean distances

$$d(\mathcal{C}_g) = \sum_{i \in \mathcal{C}_g} (\boldsymbol{x}_{i\bullet} - \bar{\boldsymbol{x}}_g)^\top (\boldsymbol{x}_{i\bullet} - \bar{\boldsymbol{x}}_g), \;\; g = 1, 2$$

where $\boldsymbol{x}_{i\bullet}$ is vector of J selected DVs from some chosen models for driver $i \in \mathcal{C}_g$ and $\bar{\boldsymbol{x}}_g$ is the mean value of $\boldsymbol{x}_{i\bullet}$ in cluster $\mathcal{C}_g$.

These four measures are applied to assess the performance of a set of models $\mathbb{M}$. To facilitate model selection, models $\mathcal{M} \in \mathbb{M}$ are ranked in *descending* order of preference for each performance measure $\mathfrak{m}_{\mathcal{M},l}, l = 1, \ldots, 4$ denoting BIC, MSE, ρ and AUC respectively to obtain the ranks $\mathcal{R}_{\mathcal{M},l} = \underset{\mathcal{M} \in \mathbb{M}}{\text{rank}}(\mathfrak{m}_{\mathcal{M},l})$. Then the weighted sum of ranks for model $\mathcal{M} \in \mathbb{M}$ in model selection is

$$\mathcal{R}_{\mathcal{M}} = \sum_{l=1}^{4} \mathsf{w}_l \times \mathcal{R}_{\mathcal{M},l} \tag{3.15}$$

where w_l is the weight for performance measure l to reflect their relative importance

in evaluating the overall performance. Models with higher $\mathcal{R}_{\mathcal{M}}$ are better. In the application, the weights $\mathsf{w}_{1:4} = (1, 2, 1, 1)$ are considered, where MSE is prioritized as a more important performance measure. This weighted measure emphasizes more on accurate claims prediction which is crucial for UBI premium calculation, as discussed in Section 3.5. Even though the classification of drivers is also very important, the measure of classification efficiency depends on the true proxy groups with uncertainties.

3.4 Empirical studies

The proposed models, namely the two-stage TP, PM, and ZIP models with different regularization and model claims select predictive DVs and classify drivers. Resampling is also performed to to enhance the robustness of the results.

For the distribution choices, section 3.3.1 introduces Poisson and NB regression. In comparison with Poisson regression for equidispersed data, NB regression can capture overdispersion. The equidispersion assumption is tested with null hypothesis, H_0 : $\text{Var}(Y_i) = \mu_i$ and alternative hypothesis H_1 : $\text{Var}(Y_i) = \mu_i + \iota g(\mu_i)$ where $g(\cdot) > 0$ is a transformation function (Cameron and Trivedi, 1990) and $\iota > 0$ ($\psi \leq 0$) indicates overdispersion (underdispersion). Refer to Appendix section 3.7.1(4) for implementation details. For model TPL-1, $\iota = 0.0369$ ($p-$value $= 0.0482$) and for model TPA-1, $\iota = 0.0369$ ($p-$value $= 0.0477$) which are marginally significant. This result agrees with the remark in Section 3.2 that the sample variance of y_i is only marginally greater than the mean. Hence, it concludes that if overdispersion exists, it may not be strong enough to necessitate NB regression. Moreover, TP, PM, and ZIP models can capture some overdispersion by splitting according to threshold and mixture components. Hence, the focus of the following analyses will be primarily on Poisson regression.

3.4.1 Two-stage threshold Poisson model

This model fits Poisson regression twice at stages 1 and 2. At stage 1, regularized Poisson regression models are trained through resampling and applied to predict claims for all drivers. Then drivers are classified according to thresholds τ of predicted annual claims into low ($\hat{a}_i < \tau$; tentatively called safe drivers) and high (risky drivers) predicted claim groups. At stage 2, two regularized Poisson regression models are applied to each group, and the selected DVs are recorded. Different lasso regularization techniques are applied in each stage, and the optimal model is chosen based on specific model selection criteria. Details are provided below.

Stage 1: Simulate subsamples to run each TP regression model and select the mostly selected DVs

To ensure model robustness and reduce overfitting, regularized Poisson regression is repeated $R = 100$ times with 70% simulated subsamples, selecting DVs for each repeat in stage 1. For a given repeat r, the optimal $\lambda_{\min,r}$ is chosen with $K = 10$ (default setting) folds CV. Then, the DVs most frequently selected are identified using a weighted count I_j^{T1} in 3.19. Details are given below.

1. **Subsamples are drawn.** Subsamples

$$\mathcal{S}_r = \{(\boldsymbol{x}_{i\bullet}, n_i, y_i), i \in \mathcal{I}_r\}, \quad r = 1, \ldots, R, \text{ with index set } \mathcal{I}_r$$

each containing $N_r = 0.7N = 9910$ drivers are drawn. The K-fold CV further splits $\mathcal{S}_r$ into K nonoverlapping parts

$$\mathcal{S}_{rk} = \{(\boldsymbol{x}_{i\bullet}, n_i, y_i), i \in \mathcal{I}_{rk}\}, \quad k = 1, \ldots, K \text{ with index set } \mathcal{I}_{rk}$$

of roughly equal size $N_k = N_r/K \approx 991$ such that $\mathcal{I}_{rk_1} \cap \mathcal{I}_{rk_2} = \phi, k_1 \neq k_2$ and $\bigcup_{k=1}^{K} \mathcal{I}_{rk} = \mathcal{I}_r$. Then the training sets are

$$\mathcal{S}_{rk}^{\mathrm{T}} = \mathcal{S}_r \setminus \mathcal{S}_{rk} = \{(\boldsymbol{x}_{i\bullet}, n_i, y_i), i \in \mathcal{I}_{rk}^{\mathrm{T}}\} \text{ with index set } \mathcal{I}_{rk}^{\mathrm{T}} = \mathcal{I}_r \setminus \mathcal{I}_{rk}.$$

2. **Optimum λ in (3.9) is searched.** Searching from $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_M)$ for some M using MSE as a regularized CV test statistic over K test sets, the optimal tuning parameter

$$\lambda_{r,\min} = \underset{\lambda_m \in \boldsymbol{\lambda}}{\operatorname{argmin}} \, \mathrm{MSE}_r(\lambda_m) = \underset{\lambda_m \in \boldsymbol{\lambda}}{\operatorname{argmin}} \frac{1}{N_r} \sum_{k=1}^{K} \sum_{i \in \mathcal{I}_{rk}} \left[y_{ki} - \exp\left(\boldsymbol{x}_{i\bullet} \boldsymbol{\beta}_{\lambda_m, rk} + \log(n_i) \right) \right]^2$$

minimizes the $\mathrm{MSE}_r(\lambda_m)$ over K test sets $\mathcal{S}_{rk}$ where the mean in (3.1) is

$$\mu^{rk}_{\lambda_m i} = \exp\left(\boldsymbol{x}_{i\bullet} \boldsymbol{\beta}_{\lambda_m, rk} + \log(n_i) \right). \tag{3.16}$$

from the training set $\mathcal{S}^{\mathsf{T}}_{rk}$ and $\lambda_m \in \boldsymbol{\lambda}$ in (3.10). Condition on $\lambda_{r,\min}$, $\boldsymbol{\beta}_r = (\beta_{r1}, \ldots, \beta_{rJ})$ is estimated based on the subsample $\mathcal{S}_r$. Apart from MSE, MAE, and deviance can also be used as the regularized CV test statistic. Figure 3.5a to 3.5c plot $\mathrm{MSE}_r(\lambda_m)$, $\mathrm{MAE}_r(\lambda_m)$ and Poisson deviance respectively with SE against $\log(\lambda_m)$ showing how these measures drop to a minimum at $\lambda_{r,\min}$ for the first subsample ($r = 1$). Figure 3.5d shows how β_{rj} shrinks to zero as λ increases. Across the three measures, optimal $\lambda_{r,\min}$ using Poisson deviance is selected according to the RMSE of predicted claims for all subsamples.

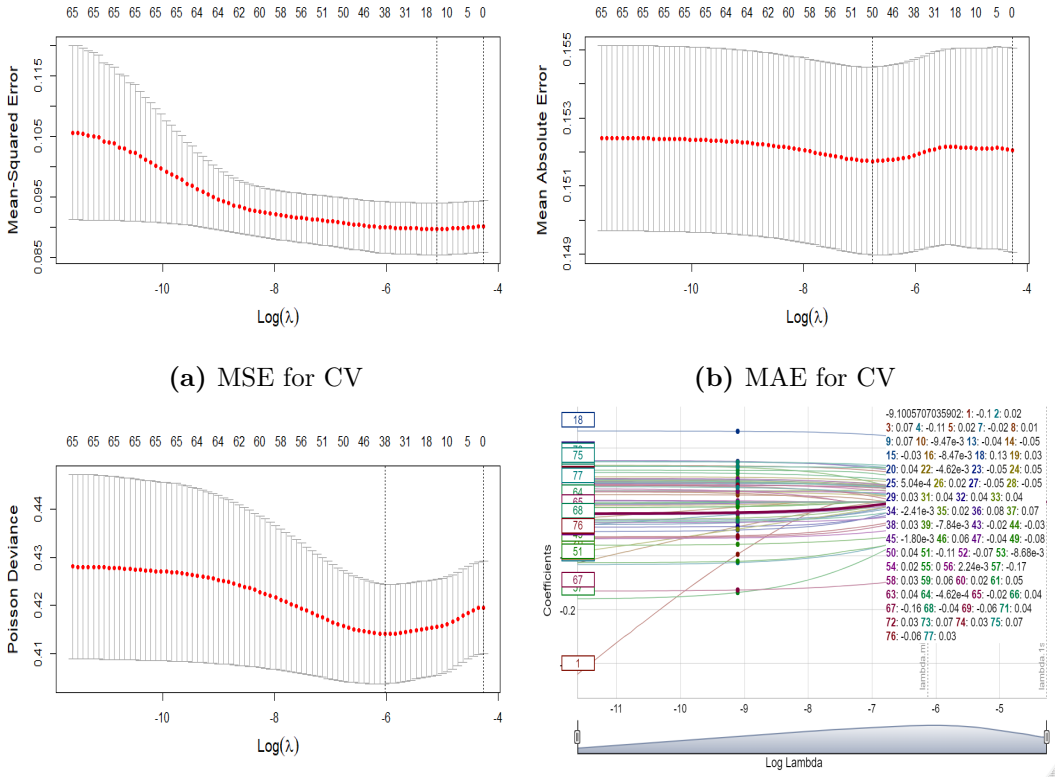
3. **Parameter estimates are averaged over R repeats.** Then the averaged non-zero (that is, selected at least once) parameter estimate is

$$\beta_j = \frac{1}{R_j} \sum_{r \in \mathcal{I}^r_j} \beta_{rj}, \quad j \in \mathcal{I}^{\beta}, \tag{3.17}$$

where the index set of J_1 DVs selected at least once over R subsamples is

$$\mathcal{I}^{\beta} = \{ j : \exists r = 1, \ldots, R, \beta_{rj} \neq 0 \}, \tag{3.18}$$

and $\mathcal{I}^r_j = \{ r : \beta_{rj} \neq 0 \}$ with size R_j is the index set of subsample $\mathcal{S}_r$ in which $\beta_{rj} \neq 0$. Then $\boldsymbol{\beta} = (\beta_{j \in \mathcal{I}^{\beta}})$ contains J_1 selected DVs in stage 1. These averaged coefficient β_j for each selected DV is reported in Table 3.9 (Appendix section 3.7.2) for TPL-1 and TPA-1 using the optimal $\lambda_{\min}$ and Poisson deviance as the



(a) MSE for CV **(b)** MAE for CV

(c) Poisson deviance for CV **(d)** Coefficients path

Figure 3.5 – (a-c) Three CV criteria across $\log \lambda_m$ to find $\lambda_{\min}$ (d) Coefficient β_j across $\log \lambda$ for stage 1 TP model using lasso regularization based on the first subsample $(r = 1)$

optimal test statistic. Using deviance, DV 10 is not even selected once for the TPL-1 model.

4. **Frequently selected DVs are further selected.** Since some DVs are rarely selected, DVs are further selected according to the selection frequency measure for DV j, which is the weighted selection counts over R_j subsamples given by

$$I_j = \sum_{r \in \mathcal{I}_j^r} \frac{1}{\text{RMSE}_r}, \quad j \in \mathcal{I}^\beta \tag{3.19}$$

where the weighted is the inverse of RMSE_r for subsample $\mathcal{S}_r$. Superscripts T1, T2, M, and Z are added to I_j when applied to stage-1 TP, stage-2 TP, PM, and ZIP models, respectively. This weighted selection counts $\boldsymbol{I} = (I_{j \in \mathcal{I}^\beta})$

using regularized test statistics MSE, MAE, and deviance are also reported in Table 3.9 (Appendix section 3.7.2) together with average coefficients $\boldsymbol{\beta}$. Table 3.4 shows that TPL-1 and TPA-1 models are selected according to model performance measures AIC, BIC, and MSE. Results of TPL-1 model in Table 3.9 (Appendix section 3.7.2) show that 12 DVs in deep gray highlight with $I_j < 0.2 \max_{j \in \mathcal{I}^{\beta}}(I_j) = 62$ are taken as rarely selected and so are excluded resulting in $J^{\mathrm{T1}} = 65 - 1 - 12 = 52$ DVs. These J^{T1} DVs can be interpreted as frequently selected DVs or simply selected DVs.

Poisson regression model with the $J^{\mathrm{T1}} = 52$ selected DVs is refitted to all drivers. Parameter estimates $\boldsymbol{\beta}^{\mathrm{T1}}$ are reported in Table 3.9 (Appendix section 3.7.2) under β_j^{T1}. To visualize these coefficients, Figure 3.8 plots the heat map of $\boldsymbol{\beta}^{\mathrm{T1}}$ for all T1 models. There are $J^{\mathrm{T1}} = 52$ selected DVs for the TPL-1 model and $J^{\mathrm{T1}} = 44$ DVs for the TPA-1 model. Let $J_{\mathrm{S}}^{\mathrm{T1}}$ be the number of significant β_j^{T1} with $p-$value < 0.05. For the TPA-1 model, Table 3.9 (Appendix section 3.7.2) shows that $J_{\mathrm{S}}^{\mathrm{T1}} = 13$ out of $J^{\mathrm{T1}} = 44$ selected DVs are significant. Further discussion of numerical results is in Section 3.4.1. Predicted claims $\hat{y}_i = \mu_i$ are calculated using the fitted means in (3.16) and $\boldsymbol{\beta}^{\mathrm{T1}}$. Then annual claims $a_i = \hat{y}_i / n_i$ are calculated.

Stage 2: Classify drivers according to predicted annual claims and thresholds. Then Poisson lasso regression is fitted to each group

Annual claims a_i are used to classify drivers into low and high claim groups according to $a_i < \tau_h$ and $a_i \geq \tau_h$, respectively. Four thresholds $\boldsymbol{\tau} = \{\tau_h\} = (0.08, 0.09, 0.10, 0.11)$ are employed, and the ratios $\mathcal{R}_h$ for the low claim group are (0.70, 0.79, 0.85, 0.90) respectively. To improve model robustness and reduce overfitting, subsamples $\mathcal{S}_{1:R}$ of size $N_r = 0.7N$ are drawn again and each $\mathcal{S}_r$ is further spitted into two groups

$$\mathcal{G}_{\mathrm{L},rh}^{\mathrm{T2}} = \{y_i \in \mathcal{S}_r : a_i < \tau_h\} \quad \text{and} \quad \mathcal{G}_{\mathrm{H},rh}^{\mathrm{T2}} = \{y_i \in \mathcal{S}_r : a_i \geq \tau_h\}, \ h = 1, \ldots, 4 \tag{3.20}$$

where T2 indicates the stage-2 of TP model. Then, Poisson lasso regression is applied to each $\mathcal{G}^{\text{T2}}_{\text{L},rh}$ and $\mathcal{G}^{\text{T2}}_{\text{H},rh}$. Let the index sets $\mathcal{I}^{\beta}_{\text{L}h}$ and $\mathcal{I}^{\beta}_{\text{H}h}$ for non-zero coefficients (that is, selected at least once from $\mathcal{S}_r$) be defined similar to $\mathcal{I}^{\beta}$ in (3.18) for $h = 1, \ldots, 4$. Then $\boldsymbol{\beta}_{\text{L}h} = (\beta_{\text{L}h, j \in \mathcal{I}^{\beta}_{\text{L}h}})$ and $\boldsymbol{\beta}_{\text{H}h} = (\beta_{\text{H}h, j \in \mathcal{I}^{\beta}_{\text{H}h}})$ are averaged parameter estimates for the low and high claim groups respectively obtained similar to $\boldsymbol{\beta}$ defined (3.17), $\boldsymbol{I}^{\text{T2}}_{\text{L}h}$ and $\boldsymbol{I}^{\text{T2}}_{\text{H}h}$ are importance measures based on the $\text{RMSE}_{r,\text{L}h}$ and $\text{RMSE}_{r,\text{H}h}$ defined similarly to $\boldsymbol{I}$ in (3.19), and $J^{\text{T2}}_{\text{L}h}$ and $J^{\text{T2}}_{\text{H}h}$ are the number of frequently selected DVs out of $\boldsymbol{\beta}_{\text{L}h}$ and $\boldsymbol{\beta}_{\text{H}h}$ with $I_{\text{L}hj} > 43$ (62×0.70 and $\mathcal{R}_1 = 0.70$ for $\tau_{0.08}$), 49 ($\tau_{0.09}$), 53 ($\tau_{0.10}$) and 56 ($\tau_{0.11}$) and $I_{\text{H}hj} > 19$ (62×0.30), 13, 9 and 6 respectively (and dropped otherwise). Poisson regression models with various selected DVs are refitted to the low and high claim groups for each h. Table 3.10 (Appendix section 3.7.2) reports parameter estimates $\boldsymbol{\beta}^{\text{T2}}_{\text{L}h}, \boldsymbol{\beta}^{\text{T2}}_{\text{H}h}$ of the best model (TPLA-2 for $\tau = 0.08, 0.09$ and TPLN-2 for $\tau = 0.10, 0.11$) when the stage one model is TPL-1 (from $J^{\text{T1}} = 52$ selected DVs) or TPA-1 ($J^{\text{T1}} = 39$) (read Section 3.4.1 for details). To visualize these coefficients, Figure 3.9 presents the heat map for models in low and high claim groups.

Numerical results

Table 3.4 shows that models TPL-1 and TPE-1 provide a similar selected number of DVs J^{T1}. The same applies to models TPA-1 and TPN-1 with adaptive weights. As feature selection is important, the best model is selected for each group of (TPL-1, TPE-1) and (TPA-1, TPN-1) and is highlighted in Table 3.4. Table 3.5 reports the mean claim mean, SD, skewness, and kurtosis for predicted claims μ_i and annual claims a_i cross-classified by observed claim $y_i = 0, 1, \geq 2$. These predicted claims, as well as predicted annual claims, are plotted in Figure 3.6 to visualize their distributions. Figure 3.7 further splits the drivers into low and high claim groups according to four thresholds using the predicted annual claims from TPA-1 and visualizes the relationship between observed and predicted annual claims. The non-linear pattern in these scatter plots is attributed partially to the impact of driving behavior revealed by the DVs.

For the stage two models, Figure 3.9 shows that the DVs which are mostly selected

Table 3.4 – Model performance measures for the stage 1 TP and ZIP models with $R = 100$ subsample of size $N_r = 9910$ each. The DVs for TP models are selected from $J_0 = 65$ DVs whereas DVs for the ZIP model are selected from $J_0' = 45$ DVs.

Model	J^{T1}	AIC	BIC	MSE	Model	J^{T1}	AIC	BIC	MSE
Stage 1 Threshold Poisson models									
TPL-1	**52**	**8020**	**8421**	**0.0866**	**TPA-1**	**39**	**7998**	**8300**	**0.0866**
TPE-1	57	8029	8468	0.0866	**TPN-1**	39	8000	8301	0.0866
Zero-inflated Poisson models									
ZIPL	J_0^{Z}= 0 J_c^{Z}= 44	8028	8368	0.0868	**ZIPA**	J_0^{Z}= **4** J_c^{Z}= **45**	**8155**	**8258**	**0.0873**

Table 3.5 – Summary of predicted claims μ_i and predicted annual claims a_i for cross classified with observed claim $y_i = 0, 1, \geq 2$ using TPL-1 model ($J_1 = 52$ frequently selected variables) and TPA-1 model ($J_1 = 39$).

Observed	$y = 0$				$y = 1$				$y =\geq$**2**			
N	13058				1030				69			
Predicted	μ_i		c_i		μ_i		c_i		μ_i		c_i	
Model	**TPL-1**	**TPA-1**	**TPL-1**	**TPA-1**	**TPL-1**	**TPA-1**	**TPL-1**	**TPA-1**	**TPL-1**	**TPA-1**	**TPL-1**	**TPA-1**
Mean	0.0802	0.0803	0.0727	0.0728	0.1149	0.1145	0.0887	0.0886	0.1427	0.1417	0.0899	0.0893
SD	0.0557	0.5527	0.0360	0.0359	0.0844	0.0805	0.0753	0.0768	0.0903	0.0886	0.0324	0.0318
Min	7.02e-06	5.36e-06	8.00e-06	6.12e-06	0.0091	0.0094	0.0091	0.0094	0.0268	0.0267	0.0416	0.0406
Max	1.3189	1.1989	0.9098	0.9402	1.3668	1.0339	1.6342	1.7291	0.5203	0.4957	0.2017	0.2021
Skewness	3.15	2.92	4.41	4.52	5.11	3.87	12.29	13.20	1.86	1.78	1.14	1.14
Kurtosis	32.79	26.62	55.94	59.62	59.60	34.00	220.65	250.22	7.93	7.46	4.41	4.51

and significant in the low claim groups are: 4, 9*, 18**, 24, 32, 37*, 47*, 57, 67**, 73* and 74 while the least are: 2, 7, 15, 16, 23*, 46* and 53. For the high claim group, they are 4, 29*, 52*, and 67** in which DV 67** is selected for both groups. The information content of each DV is indicated by asterisks. See Table 3.1 for details. More DVs are selected for the low claim group with thresholds $\tau_{2:4} = 0.09, 0.10, 0.11$ and these selected DVs are relatively more informative. Two DVs, 4 and 67, are selected in both low and high-level claim groups. For model selection, Table 3.6 summarizes the model performance measures for 32 models with 8 models for each threshold. The two criteria, BIC and AUC, in Table 3.6 are averaged over the two groups using the ratio $\mathcal{R}_h$ in Tables 3.10 (Appendix section 3.7.2). The number $J_{\text{L}h}^{\text{T2}}, J_{\text{HS}h}^{\text{T2}}$ of selected $\beta_{\text{L}hj}^{\text{T2}}$ and $\beta_{\text{L}hj}^{\text{T2}}$ and the number $J_{\text{LS}h}^{\text{T2}}, J_{\text{HS}h}^{\text{T2}}$ of significant $\beta_{\text{L}hj}^{\text{T2}}$ and $\beta_{\text{L}hj}^{\text{T2}}$ with $p-$value < 0.05 are also reported in Table 3.6. Those models with 0 $J_h^{\text{T2}}, J_{\text{S}h}^{\text{T2}}$ are dropped for either

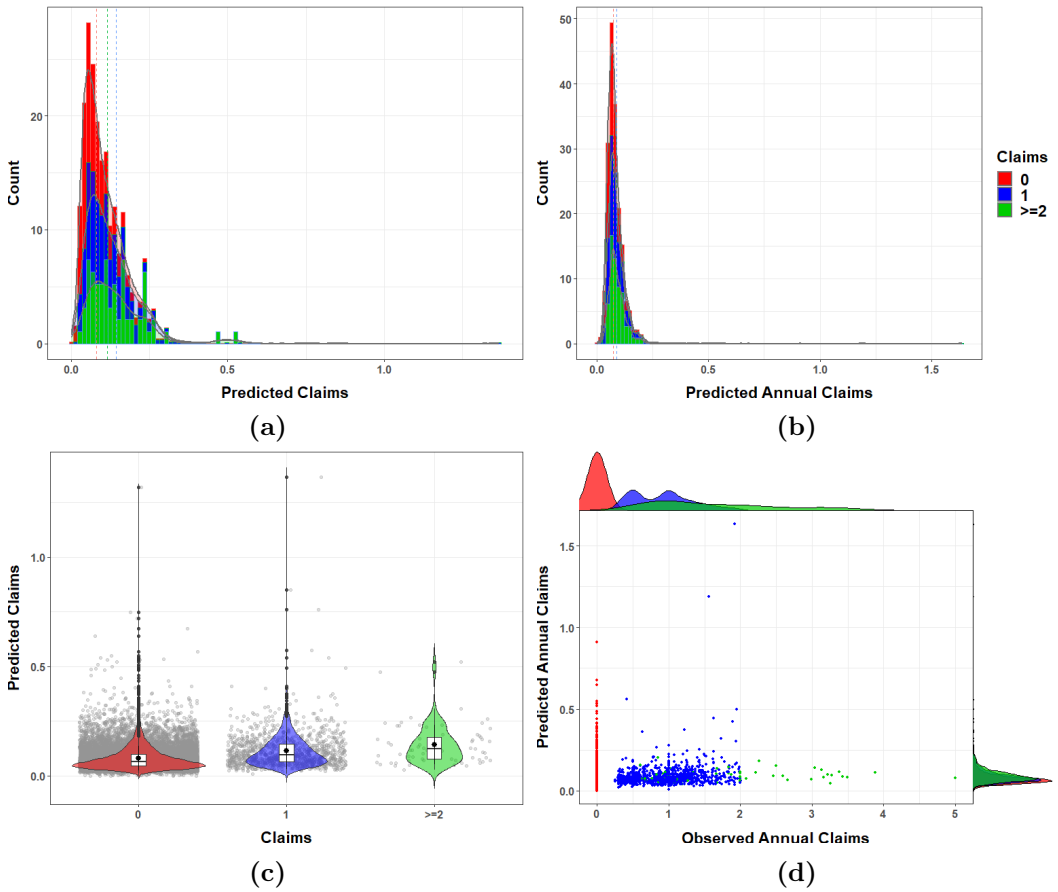


Figure 3.6 – Compare observed and predicted claims and annual claims using the TPL-1 model. (a) Stacked histogram with density curve for predicted claims μ_i; (b) Stacked histogram with density curve for predicted annual claims a_i, (c) Violin with boxplots for predicted claims $\hat{\mu}_i$ by y_i, and (d) Scatter plot of predicted annual claims $\hat{a}_i$ against observed annual claims a_i with marginal densities and colors to indicate observed claims y_i 0, 1, ≥ 2

groups and the best model is chosen $\mathcal{M}$ according to weighted sum of ranks $\mathcal{R}_{\mathcal{M}}$ in (3.15) with weights $\mathbf{w}= (1, 2, 1, 1)$ (see Section 3.3.3). The best stage two model is TPLA-2 for $\tau = 0.08, 0.09$ and TPLN-2 for $\tau = 0.10, 0.11$. For the four selected models, Figures 3.12 plots the ROC curve and reports AUC. The results are compared in Section 3.4.4 with PM and ZIP models.

Table 3.6 – Number of DVs selected, $J^{\mathrm{T2}}_{\mathrm{L}h}, J^{\mathrm{T2}}_{\mathrm{H}h}, J^{\mathrm{M}}_{\mathrm{L}}, J^{\mathrm{M}}_{\mathrm{H}}$ ($\beta^{\mathrm{T2}}_{\mathrm{L}hj}, \beta^{\mathrm{T2}}_{\mathrm{H}hj}, \beta^{\mathrm{M}}_{\mathrm{L}j}, \beta^{\mathrm{M}}_{\mathrm{H}j} \neq 0$ and $I^{\mathrm{T2}}_{\mathrm{L}hj}, I^{\mathrm{T2}}_{\mathrm{H}hj}$, $I^{\mathrm{M}}_{\mathrm{L}j}, I^{\mathrm{M}}_{\mathrm{H}j} > 62$), the number of significant selected DVs, $J^{\mathrm{T2}}_{\mathrm{LS}h}, J^{\mathrm{T2}}_{\mathrm{HS}h}, J^{\mathrm{M}}_{\mathrm{LS}}, J^{\mathrm{M}}_{\mathrm{HS}}$, and performance measures, BIC, MSE, ρ and AUC using all data for the stage two TP models and PM model. Models with zero selected DVs for the low or high-claim group are excluded. Boldfaced and yellow highlighted measures and the weighted sum of rank $\mathcal{R}$ have top rank for each threshold. The model with the top $\mathcal{R}$ is selected from each of the TP-2 given τ and PM models.

		J^{T2}_h		$J^{\mathrm{T2}}_{\mathrm{S}h}$		Ranking importance of measures				Assigned weightage				Weighted
										w_1=1	w_2=2	w_3=1	w_4=1	sum
		Low	High	Low	High	BIC	MSE	ρ	AUC	BIC	MSE	ρ	AUC	$\mathcal{R}$
τ_i	Models	Stage 2 threshold Poisson model												
$\tau_{0.08}$	TPLL-1	0	2	0	2	-	-	-	-	-	-	-	-	-
	TPLE-1	0	1	0	1	-	-	-	-	-	-	-	-	-
	TPLA-1	16	22	0	3	8286$^{(4)}$	**0.0863**$^{(4)}$	**0.1336**$^{(5)}$	0.5990$^{(4)}$	4.0	8.0	5.0	4.0	**21.0**
	TPLN-1	23	19	1	1	8326$^{(3)}$	0.0864$^{(2)}$	0.1261$^{(2)}$	**0.6119**$^{(5)}$	3.0	4.0	2.0	5.0	14.0
	TPAL-1	5	0	2	0	-	-	-	-	-	-	-	-	-
	TPAE-1	17	2	4	1	**8132**$^{(5)}$	0.0867$^{(1)}$	0.1176$^{(1)}$	0.5958$^{(3)}$	5.0	2.0	1.0	3.0	11.0
	TPAA-1	25	20	4	4	8347$^{(2)}$	**0.0863**$^{(4)}$	0.1324$^{(4)}$	0.5757$^{(1)}$	2.0	8.0	4.0	1.0	15.0
	TPAN-1	27	19	3	4	8356$^{(1)}$	**0.0863**$^{(4)}$	0.1323$^{(3)}$	0.5781$^{(2)}$	1.0	8.0	3.0	2.0	14.0
$\tau_{0.09}$	TPLL-1	26	3	4	2	**8204**$^{(8)}$	0.0864$^{(1.5)}$	0.1237$^{(1)}$	0.6186$^{(7)}$	8.0	3.0	1.0	7.0	19.0
	TPLE-1	39	4	7	2	8321$^{(5)}$	0.0863$^{(3.5)}$	0.1282$^{(4)}$	0.6111$^{(5)}$	5.0	7.0	4.0	5.0	21.0
	TPLA-1	38	14	6	1	8398$^{(3)}$	**0.0862**$^{(6.5)}$	0.1332$^{(7)}$	0.6129$^{(6)}$	3.0	13.0	7.0	6.0	**29.0**
	TPLN-1	41	11	8	1	8400$^{(2)}$	**0.0862**$^{(6.5)}$	0.1291$^{(5)}$	**0.6276**$^{(8)}$	2.0	13.0	5.0	8.0	28.0
	TPAL-1	35	3	10	3	8281$^{(7)}$	0.0863$^{(3.5)}$	0.1263$^{(2.5)}$	0.5972$^{(4)}$	7.0	7.0	2.5	4.0	20.5
	TPAE-1	38	3	10	2	8311$^{(6)}$	0.0864$^{(1.5)}$	0.1263$^{(2.5)}$	0.5936$^{(3)}$	6.0	3.0	2.5	3.0	14.5
	TPAA-1	30	15	9	0	8341$^{(4)}$	**0.0862**$^{(6.5)}$	0.1292$^{(6)}$	0.5921$^{(2)}$	4.0	13.0	6.0	2.0	25.0
	TPAN-1	34	18	9	1	8404$^{(1)}$	**0.0862**$^{(6.5)}$	**0.1338**$^{(8)}$	0.5890$^{(1)}$	1.0	13.0	8.0	1.0	23.0
$\tau_{0.10}$	TPLL-1	49	3	12	0	8400$^{(4)}$	0.0862$^{(2.5)}$	0.1281$^{(2)}$	**0.6284**$^{(8)}$	4.0	5.0	2.0	8.0	19.0
	TPLE-1	51	5	13	0	8434$^{(2)}$	0.0861$^{(5)}$	0.1319$^{(4)}$	0.6276$^{(7)}$	2.0	10.0	4.0	7.0	23.0
	TPLA-1	38	18	12	0	8427$^{(3)}$	0.0860$^{(7)}$	0.1329$^{(5)}$	0.6218$^{(6)}$	3.0	14.0	5.0	6.0	28.0
	TPLN-1	42	23	11	0	8507$^{(1)}$	**0.0859**$^{(8)}$	**0.1398**$^{(8)}$	0.6170$^{(5)}$	1.0	16.0	8.0	5.0	**30.0**
	TPAL-1	38	6	11	0	8327$^{(7)}$	0.0862$^{(2.5)}$	0.1313$^{(3)}$	0.5944$^{(2)}$	7.0	5.0	3.0	2.0	17.0
	TPAE-1	38	5	11	0	**8321**$^{(8)}$	0.0863$^{(1)}$	0.1271$^{(1)}$	0.5933$^{(1)}$	8.0	2.0	1.0	1.0	12.0
	TPAA-1	35	17	9	0	8396$^{(6)}$	0.0861$^{(5)}$	0.1353$^{(7)}$	0.6005$^{(3)}$	6.0	10.0	7.0	3.0	26.0
	TPAN-1	35	17	9	0	8397$^{(5)}$	0.0861$^{(5)}$	0.1341$^{(6)}$	0.6012$^{(4)}$	5.0	10.0	6.0	4.0	25.0
$\tau_{0.11}$	TPLL-1	50	6	13	1	8431$^{(4)}$	0.0861$^{(2)}$	0.1310$^{(1)}$	**0.6214**$^{(8)}$	4.0	4.0	1.0	8.0	17.0
	TPLE-1	51	12	13	0	8492$^{(3)}$	0.0860$^{(4.5)}$	0.1391$^{(5)}$	0.6200$^{(7)}$	3.0	9.0	5.0	7.0	24.0
	TPLA-1	41	24	13	0	8503$^{(2)}$	0.0859$^{(6.5)}$	0.1429$^{(7)}$	0.6183$^{(6)}$	2.0	13.0	7.0	6.0	28.0
	TPLN-1	43	28	12	0	8554$^{(1)}$	**0.0857**$^{(8)}$	**0.1457**$^{(8)}$	0.6083$^{(5)}$	1.0	16.0	8.0	5.0	**30.0**
	TPAL-1	38	8	15	1	**8344**$^{(8)}$	0.0861$^{(2)}$	0.1347$^{(2)}$	0.6013$^{(2)}$	8.0	4.0	2.0	2.0	16.0
	TPAE-1	38	9	15	0	8353$^{(7)}$	0.0861$^{(2)}$	0.1357$^{(3)}$	0.6008$^{(1)}$	7.0	4.0	3.0	1.0	15.0
	TPAA-1	34	18	15	1	8396$^{(6)}$	0.0859$^{(6.5)}$	0.1366$^{(4)}$	0.6031$^{(4)}$	6.0	13.0	4.0	4.0	27.0
	TPAN-1	35	20	15	1	8422$^{(5)}$	0.0860$^{(4.5)}$	0.1403$^{(6)}$	0.6029$^{(3)}$	5.0	9.0	6.0	3.0	23.0
		Poisson mixture												
	PML	45	18	4	3	**8551**$^{(4)}$	0.0832$^{(1)}$	0.2345$^{(1)}$	**0.6237**$^{(4)}$	4.0	2.0	1.0	4.0	11.0
	PME	45	35	2	4	8704$^{(2)}$	0.0825$^{(2)}$	0.2581$^{(2)}$	0.6216$^{(3)}$	2.0	4.0	2.0	3.0	11.0
	PMA	39	40	6	5	8692$^{(3)}$	**0.0805**$^{(4)}$	0.2972$^{(3)}$	0.6003$^{(1)}$	3.0	8.0	3.0	1.0	**15.0**
	PME	45	44	1	7	8781$^{(1)}$	0.0811$^{(3)}$	**0.3002**$^{(4)}$	0.6073$^{(2)}$	1.0	6.0	4.0	2.0	13.0

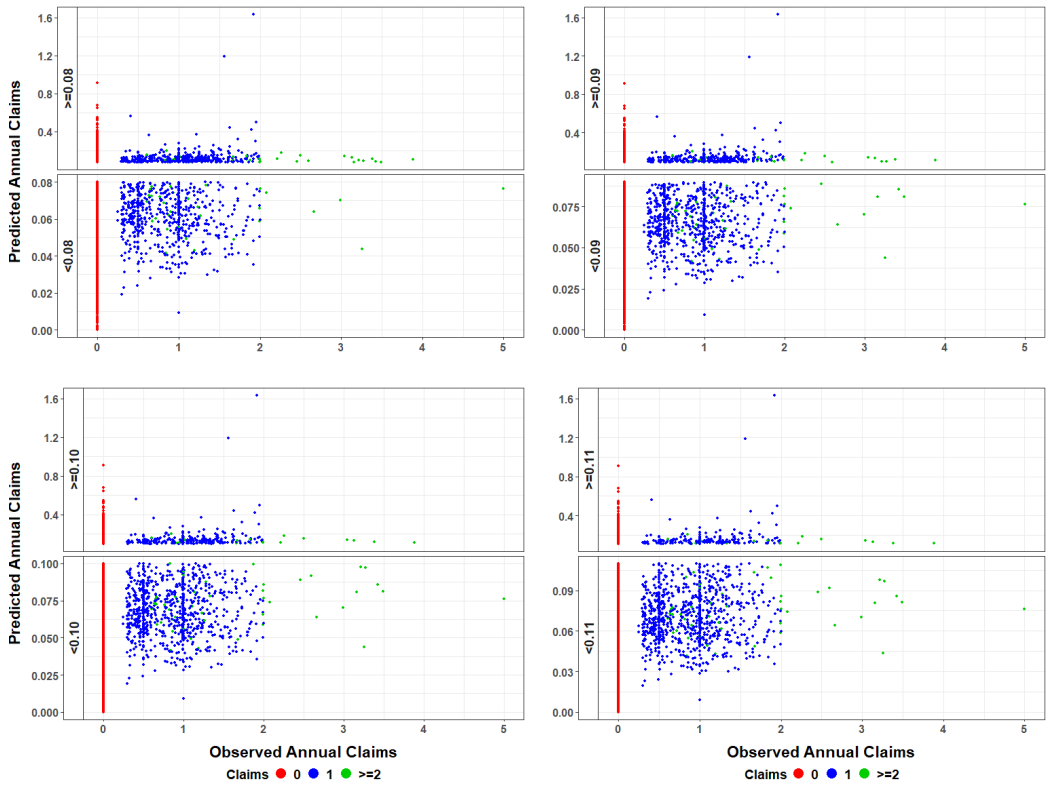


Figure 3.7 – Scatter plots of observed annual claims a_i against predicted annual claims $\hat{a}_i$ using TPA-1 model cross-classified with low and high claim groups by the four thresholds with color indicating claims $y_i = 0, 1, \geq 2$.

3.4.2 Poisson mixture model

Four lasso regularized PM models, namely PML, PMA, PME, and PMN, are considered for driver classification. To enhance the robustness of the results, a resampling procedure is performed, obtaining $R = 100$ subsamples, each with a size of $N_r = 9910$ (70% resampling). Parameters are selected from the $J_0' = 45$ more informative DVs to provide stable results (see Section 3.2). In each subsample $\mathcal{S}_r$, the regularized Poisson mixture model is estimated using $K = 10$ folds CV. Then drivers are classified into $\mathcal{G}^{\mathrm{M}}_{\mathrm{L},r}$ and $\mathcal{G}^{\mathrm{M}}_{\mathrm{H},r}$ according to $\hat{\mathfrak{z}}_{ig}$ in (3.6) ≥ 0.5 or < 0.5 respectively. Let the index sets $\mathcal{I}^{\beta}_{\mathrm{L}}$ and $\mathcal{I}^{\beta}_{\mathrm{H}}$ for non-zero coefficient (that is, selected at least once from $\mathcal{S}_r$) be defined similar to $\mathcal{I}^{\beta}$ in (3.18). Then $\boldsymbol{\beta}_{\mathrm{L}} = (\beta_{\mathrm{L},j\in\mathcal{I}^{\beta}_{\mathrm{L}}})$ and $\boldsymbol{\beta}_{\mathrm{H}} = (\beta_{\mathrm{H},j\in\mathcal{I}^{\beta}_{\mathrm{H}}})$ are averaged parameter estimates for the low and high claim groups respectively obtained similar to $\boldsymbol{\beta}$ defined in (3.17), $\boldsymbol{I}^{\mathrm{M}}_{\mathrm{L}}$ and $\boldsymbol{I}^{\mathrm{M}}_{\mathrm{H}}$ are importance measures based on the $\mathrm{RMSE}^{\mathrm{M}}_{r,\mathrm{L}}$

and $\text{RMSE}^{\text{M}}_{r,\text{H}}$ defined similar to $\boldsymbol{I}$ in (3.19), $\mathcal{R}^{\text{M}}$ is the average ratio of the low group size over $R = 100$ subsamples, and J^{M}_{L} and J^{M}_{H} are the number of selected DVs out of $\boldsymbol{\beta}_{\text{L}}$ and $\boldsymbol{\beta}_{\text{H}}$ with $I^{\text{M}}_{\text{L}j} > 43$ (62×0.69 and $\mathcal{R}^{\text{M}} = 0.69$ for PML), 45 (62×0.73 for PMA), and $I^{\text{M}}_{\text{H}j} > 19$ (62×0.31 for PML), 17 (62×0.27 for PMA) similar to J^{T1}. Results show that some subsamples have too low sample size (< 200) for the high claim group or too low differences (< 0.005) of observed annual claims between the two groups or both. Both criteria suggest ineffective grouping and recommend elimination. Consequently, the actual number of subsamples is 172 and 113 for the PML and PMA models, respectively, to collect 100 subsamples based on the two criteria.

To obtain overall parameter estimates $\boldsymbol{\beta}^{\text{M}}_{\text{L}}$ and $\boldsymbol{\beta}^{\text{M}}_{\text{H}}$, the selected DVs are refitted to PM model again. Table 3.11 (Appendix section 3.7.2) reports $\boldsymbol{\beta}^{\text{M}}_{\text{L}}$ and $\boldsymbol{\beta}^{\text{M}}_{\text{H}}$ of PML and PMA models together with $\boldsymbol{I}^{\text{M}}_{\text{L}}$ and $\boldsymbol{I}^{\text{M}}_{\text{H}}$. To select the best PM model, Table 3.6 reports performance measures BIC, MSE, ρ and AUC as well as the number of selected DVs J^{M}_{L}, J^{M}_{H} and the number of selected significant DVs J^{M}_{LS}, J^{M}_{HS} for each PM model. Results show that, in general, PM models tend to have more selected and significant DVs for the high claim group than the TP models.

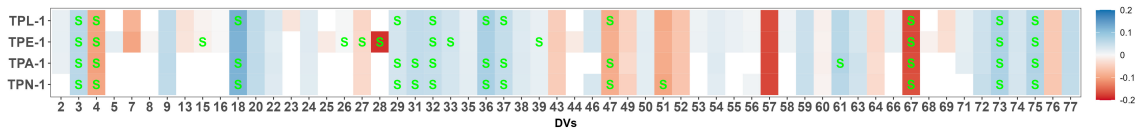


Figure 3.8 – Heat map of coefficients $\boldsymbol{\beta}^{\text{T1}}$ for the selected DVs of stage 1 TP regression model with significance indicated by "S"

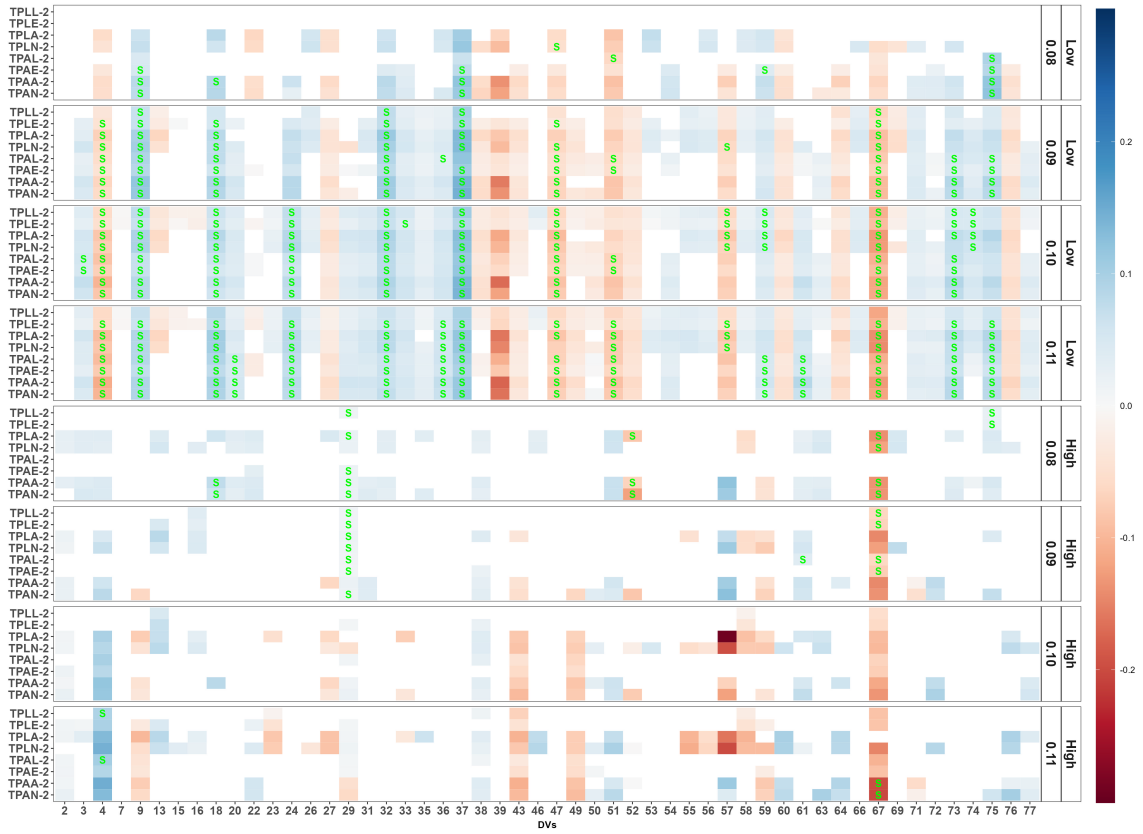


Figure 3.9 – Heat map of parameter estimates $\boldsymbol{\beta}_{\mathrm{L}h}^{\mathrm{T2}}, \boldsymbol{\beta}_{\mathrm{H}h}^{\mathrm{T2}}, h = 1, \ldots, 4$ for the stage 2 TP model with significant coefficients denoted by "S"

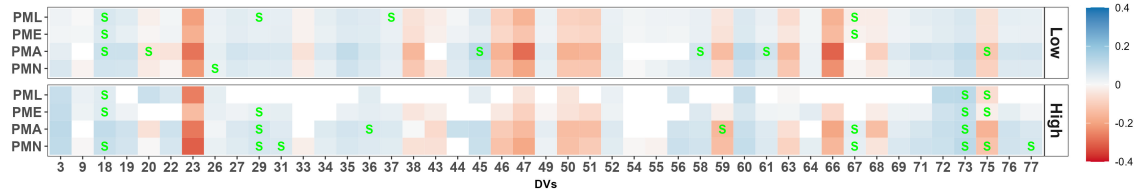


Figure 3.10 – Heat map of parameter estimates $\beta_{\mathrm{L}j}^{\mathrm{M}}$ and $\beta_{\mathrm{H}j}^{\mathrm{M}}$ for PM model by low and high claim groups with significant coefficients indicated by "S"

According to sum of ranks $\mathcal{R}_{\mathcal{M}}$ in (3.15), PMA model is selected. Figure 3.10 plots the heat map of parameter estimates of the two groups for the four PM models. For the selected PMA model, Table 3.11 (Appendix section 3.7.2) shows that there are $J_{\mathrm{L}}^{\mathrm{M}}$= 39 DVs selected for the low claim group and $J_{\mathrm{H}}^{\mathrm{M}}$ = 40 DVs selected for the high claim group of which 6 (18, 20, 45, 58, 61, 75) and 5 (29, 36, 59, 67, 73) DVs are significant respectively. Across all 4 models, the two sets of mostly

selected and significant variables are (**18****,20*,26*,29*,37*,45*,58*,61*,**67****,75) and (18**,**29***,31*,36*,59*,**67****,73*,75*,77*) respectively for the low and high claim groups. These two sets of significant DVs are quite different from those of TP models as only 2 DVs from each group (in boldface) are also selected by TP models. Again, DV 67 is selected by both groups, the same as the case of TP models. Refer to Appendix section 3.7.1(5) for implementing PM models.

3.4.3 Zero-inflated Poisson lasso regression model

Since 92% of claims are zero, the ZIP model in (3.7) and (3.8) (Lambert, 1992; Zeileis et al., 2008) is employed to capture the structural zero portion of claims. This is done to test whether the structural zero claim group should be included in modeling claims. Same as the TP and PM models, subsamples $\mathcal{S}_{1:R}$ are drawn $R = 100$ times, each with $N_r = 9910$ drivers, and ZIP lasso regression is applied to each $\mathcal{S}_r$ to robustify the selection of DVs. Procedures are similar to the cases of TP and PM models. Same as the PM model, DVs are selected from $J_0' = 45$ more informative DVs to provide stable results.

Let $\mathcal{I}_0^\beta$ and $\mathcal{I}_c^\beta$ be the index sets of non-zero parameter estimates (that is, selected at least once from $\mathcal{S}_r$) in the zero and count models respectively defined similar to $\mathcal{I}^\beta$ in (3.18). Then, averaged parameter estimates $\boldsymbol{\beta}_0 = (\beta_{0,j\in\mathcal{I}_0^\beta})$ for the zero model and $\boldsymbol{\beta}_c = (\beta_{c,j\in\mathcal{I}_c^\beta})$ for the count model are obtained similar to $\boldsymbol{\beta}$ in (3.17), $\boldsymbol{I}_0^Z$ and $\boldsymbol{I}_c^Z$ are importance measures based on the $\text{RMSE}_{0,r}^Z$ and $\text{RMSE}_{c,r}^Z$ respectively defined similarly to $\boldsymbol{I}$ in (3.19), and J_0^Z and J_c^Z are the number of selected DVs out of $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_c$ with $I_{0,j}^Z > 62$ and $I_{c,j}^Z > 62$ similar to J^{T1}. Next, the J_0^Z and J_c^Z selected DVs are refitted to the ZIP model for all data to obtain overall parameter estimates $\boldsymbol{\beta}_0^Z$ and $\boldsymbol{\beta}_c^Z$. Parameters $\boldsymbol{\beta}_0, \boldsymbol{\beta}_c$ averaged before refit, parameters $\boldsymbol{\beta}_0^Z, \boldsymbol{\beta}_c^Z$ after refit and the importance measures $\boldsymbol{I}_0^Z, \boldsymbol{I}_c^Z$ are reported in Table 3.11 (Appendix section 3.7.2).

Table 3.4 reports the number of selected DVs J_0^Z and J_c^Z, and performance measures AIC, BIC, and MSE for ZIPL and ZIPA models following the regularization choices from the two chosen TPL-1 and TPA-1 models. Between the two ZIP models, the

ZIPA model is chosen because it has non-zero DVs selected for the zero component and a lower BIC. However, the MSE is the highest among all models in Table 3.6, indicating low predictive power.

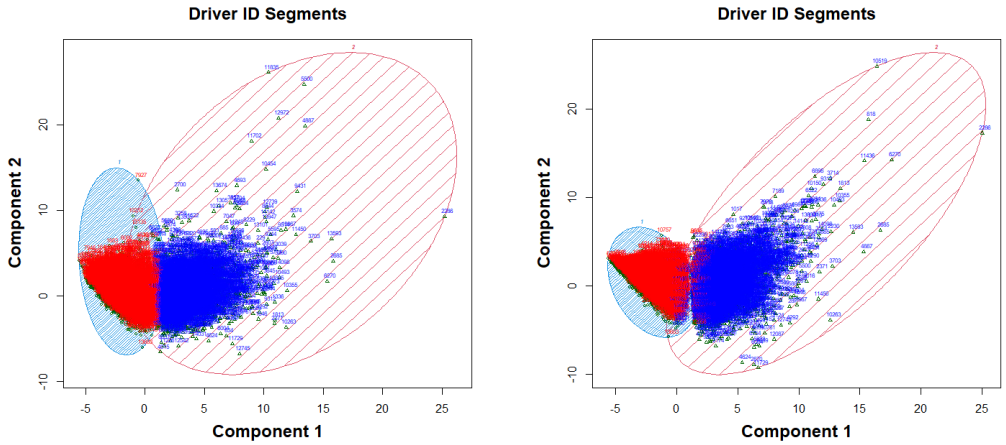
More importantly, the zero model estimates the probability of structural zero among all zeros. Still, it does not guide the classification of safe and risky drivers because safe drivers can claim less but not necessarily none, and risky drivers can claim none by luck or for no claim bonus. Hence, drivers classified to the structural zero claim group are not necessarily safe drivers. As a result, the ZIPA model is excluded from model comparison and selection. Refer to Appendix 3.7.1(6) for implementing ZIP models.

3.4.4 Model comparison and selection

The performance of TP and PM models is compared in terms of claim prediction, predictive DVs selection, and driver classification. Table 3.6 displays the performance of all 32 TP models and 4 PM models. The best TP model for each threshold and the best PM model are selected according to $\mathcal{R}$ in (3.15). Among the selected TPLA-2 ($\tau = 0.08, 0.09$), TPLN-2 ($\tau = 0.10, 0.11$), and PMA models, Table 3.7 shows that the best model is PMA.

Apart from numerical measures for claim prediction and DVs selection, driver classification performance can be visualized using ROC curves. Section 3.3.3 introduces the AUC according to 3 classes of drivers with $y_i = 0, 1, \geq 2$ or all drivers. K-means clustering is initially applied to predict two clusters using $J^{\mathrm{T1}} = 52$ selected DVs (from the TPL-1 model) for all stage 2 TP models and $J_0' = 45$ informative DVs for the PM model as the unobserved true groups of safe and risky drivers. Figure 3.11 plots the two clusters using the first two principal components (PCs) that explain 17.44% and 22.11% of variance respectively using selected DVs for the two cases. Results show that the first PC can separate the two clusters well for both cases. For the stage 2 TP (PM) model, $N_1 = 9450$ (9372) claims are assigned to the low claim cluster accounting for 67% (66%) of drivers using K-mean clustering. These two low group

cluster proportions are not far away from the estimated proportions $\mathcal{R}_h = 0.7$ to 0.89 in Tables 3.10 (Appendix section 3.7.2) using the TPL-1 model as well as $\mathcal{R}^{\mathrm{M}} = 0.73$ using PMA model.



(a) **(b)**

Figure 3.11 – K-mean clustering analysis to segment drivers into low claim cluster in red with blue shade ellipse and high claim cluster in blue with red shade ellipse using (a) $J_1^{\mathrm{T}} = 52$ selected DVs from TPL-1 (b) $J^{\mathrm{M}} = 45$ informative DVs for PM models.

Figure 3.12 plots the ROC curves with AUC values using classifier $\hat{y}_i/n_i$ for the best stage 2 TP models under each threshold as well as classifier $\hat{\mathbf{z}}_{i1}$ in (3.6) for the best PMA model. The zigzag patterns indicate a small sample size of drivers with $y \geq 2$. Table 3.6 reports overall the AUC values, the weighted averages of the three AUC values in Figure 3.12. Table 3.7 shows that model TPLN-2 when $\tau = 0.10$ is the best classifier of low and high claim groups while PMA shows relatively lower classifying power. Figure 3.13 plots five ROC curves for all drivers using the models in Figure 3.12. Again, TPLN-2 when $\tau = 0.10$ is the best classifier. However, the accuracy of the results depends on whether K-means clustering can estimate the true latent groups well.

It is natural to anticipate that some zero-claim drivers might be risky, and non-zero claim drivers might be safe. This disagreement reflects DV's impact on assessing driving risk apart from the claim information. Zero disagreement (ZD) is defined as the proportion, out of all drivers, of those zero claim drivers classified as risky and

Table 3.7 – Model performances for stage 2 TP and PM models. The assigned weight of rank values for BIC, MSE, ρ, and AUC are calculated with the product of measures importance (1:2:1:1).

Models	Performance measures and ranks				Assigned weights				Weighted sum
					w_1 =1	w_2 =2	w_3 =1	w_4 =1	
	BIC	MSE	ρ	AUC	BIC	MSE	ρ	AUC	$\mathcal{R}$
$\tau_{0.08}$: TPLA	**8286**$^{(5)}$	0.0863$^{(1)}$	0.1336$^{(2)}$	0.5990$^{(1)}$	5	2	2	1	10
$\tau_{0.09}$: TPLA	8398$^{(4)}$	0.0862$^{(2)}$	0.1332$^{(1)}$	0.6129$^{(4)}$	4	4	1	4	13
$\tau_{0.10}$: TPLN	8507$^{(3)}$	0.0859$^{(3)}$	0.1398$^{(3)}$	**0.6170**$^{(5)}$	3	6	3	5	17
$\tau_{0.11}$: TPLN	8554$^{(2)}$	0.0857$^{(4)}$	0.1457$^{(4)}$	0.6083$^{(3)}$	2	8	4	3	17
PMA	8692$^{(1)}$	**0.0805**$^{(5)}$	**0.2972**$^{(5)}$	0.6003$^{(2)}$	1	10	5	2	**18**

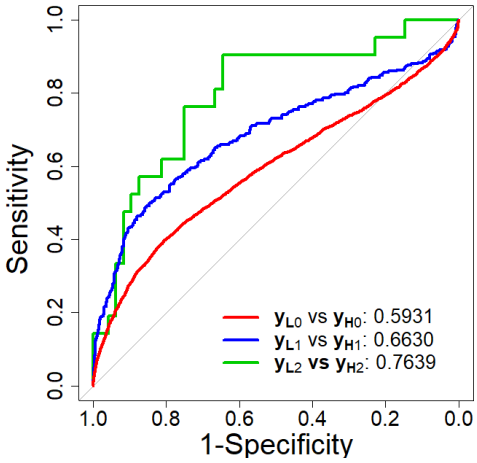
non-zero claim drivers classified as safe. This ZD is 3.2% for the best PMA model. This small ZD is due to the low MSE showing agreement between predicted and observed claims.

3.5 UBI experience-rating premium

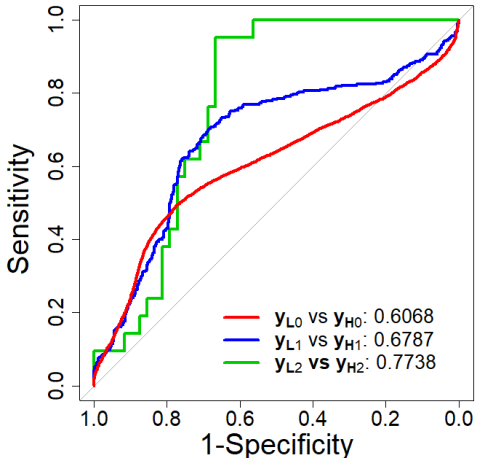
In the context of general insurance, a common approach for assessing risk in the typical short-tail portfolio involves multiplying predicted claims frequency by claims severity to determine the risk premium. This derived risk premium is subsequently factored into the profit margin, alongside operating expenses, to determine the final premium charged to customers. This section focuses on claims frequency, and the premium calculation discussed herein, operated under the assumption of constant claim severity. Consequently, the premium calculation is based on predicting claims frequency.

The traditional experience-rating method prices premiums using historical claims and offers the same rate for drivers within the same risk group (low/high or safe/risky). If individual claim history is available, premiums can be calculated using individual claims relative to overall claims, both from historical records. However, although this extended historical experience-rating method can capture individual differences of risk within a group, it still fails to reflect drivers' recent driving risk. The integration

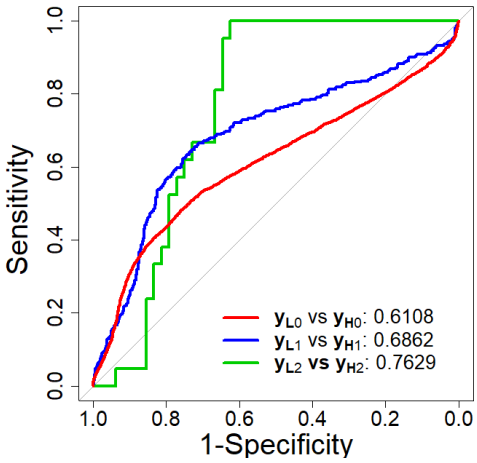
(a) $\tau_{0.08}$: TPLA-2



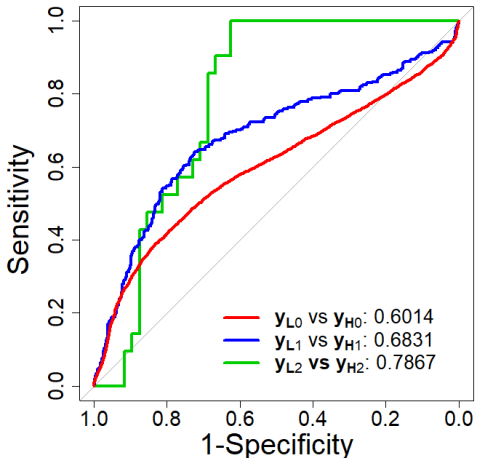
(b) $\tau_{0.09}$: TPLA-2



(c) $\tau_{0.10}$: TPLN-2



(d) $\tau_{0.11}$: TPLN-2



(e) PMA

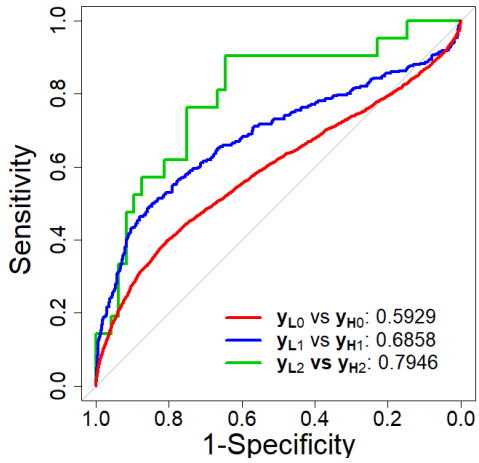


Figure 3.12 – ROC curve and AUC using K-mean clusters to estimate "observed" safe and risky claim groups and using predicted annual claims $(\hat{y}_i/n_i)$ from the four best stage 2 TP and predicted group memberships $\hat{\mathfrak{z}}_{i1}$ from PMA model as classifiers for each class of drivers with $y_i = 0, 1, \geq 2$.

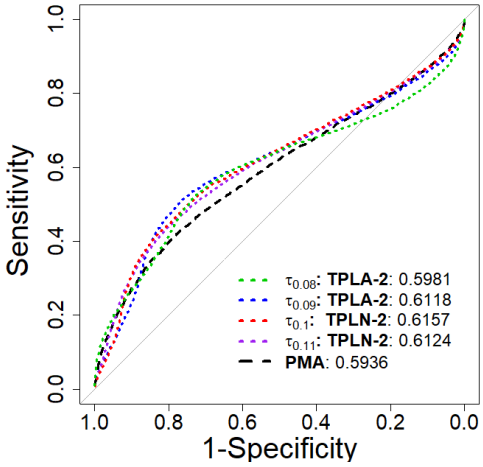


Figure 3.13 – ROC curve and AUC using K-mean clusters to estimate "observed" safe and risky claim groups and predicted annual claims from the four best stage 2 TP and one PM model as predictors for all drivers

of telematic data enables tailoring pricing to *current* individual risks. This enhanced method is termed the *UBI experience-rating* method, leveraging premium pricing as a strategic approach to refine the methodology.

Given a new driver i was classified to claim group g with index set $\mathcal{C}_g$ of all drivers in this group and $i \in \mathcal{C}_g$. Let P_{it} be his premium for year t, $\mathsf{L}_{i,t-1}$ be the historical claim/loss in year $t-1$, $\mathsf{L}^g_{t-1} = \sum_{i' \in \mathcal{C}_g} \mathsf{L}_{i',t-1}$ be the total claim/loss from the claim group g that driver i was classified to, and $P^g_{t-1} = \sum_{i' \in \mathcal{C}_g} P_{i',t-1}$ be the total premium from the claim group g. Moreover, suppose the best PMA model was trained using the sample of drivers. Let $\boldsymbol{x}_{i\bullet,t}$ be the observed DVs for driver i at time t, $g = 1$ safe group ($g = 2$ risky group) be the classified group if the group indicator $\hat{\mathfrak{z}}_{i1} > 0.5$ (otherwise), $\hat{y}_{i,t}$ in (3.4) be the predicted claim frequency given $\boldsymbol{x}_{i\bullet,t}$, $\hat{y}^g_t = (\sum_{i' \in \mathcal{C}_g} \hat{y}_{i',t})/N_g$ be the average predicted claim frequencies from the claim group g that driver i was classified to and N_g is the size of group g.

Using the proposed *UBI experience-rating* method, the premium P_{it} for driver i in year t is given by

$$P_{it,\varkappa} = (1 + \mathsf{R}^{\Delta}_{i,t}) \times \bar{P}^g_t \times n_{it} \times F + \bar{P}^*_t \times n_{it} \times (1 - F) \tag{3.21}$$

where $\bar{P}_t^g$ is the group average annual premium in period t from the group data, $\bar{P}_t^*$ is the average annual premium from all data or some other data source, F is the credibility factor (Dean, 1997), n_{it} is the exposure of driver i, $\mathsf{R}_{i,t-1}^{\Delta}$ is the individual adjustment factor to the overall group loss ratio given by

$$\mathsf{R}_{i,t}^{\Delta} = \mathsf{R}_{i,t-1}^{\Delta,\mathrm{H}} + \varkappa \mathsf{R}_{i,t}^{\Delta,\mathrm{UB}}, \tag{3.22}$$

which is the sum of *historical* loss rate change adjustment $\mathsf{R}_{i,t-1}^{\Delta,\mathrm{H}}$ and weighted *UBI predicted* loss rate change adjustment $\mathsf{R}_{i,t-1}^{\Delta,\mathrm{UB}}$ and $\varkappa \in [0,1]$ is the *UBI policy* parameter to determine how much UBI adjustment is applied to $\mathsf{R}_{i,t-1}^{\Delta,\mathrm{UB}}$ in $\mathsf{R}_{i,t}^{\Delta}$ when updating the premium to account for current driving behavior. The *historical* loss rate change $\mathsf{R}_{i,t-1}^{\Delta,y}$, historical individual loss ratio $\mathsf{R}_{i,t-1}$, and historical group loss ratio R_{t-1}^{g} are respectively

$$\mathsf{R}_{i,t-1}^{\Delta,\mathrm{H}} = \frac{\mathsf{R}_{i,t-1} - \mathsf{R}_{t-1}^{g}}{\mathsf{R}_{t-1}^{g}}, \quad \mathsf{R}_{i,t-1} = \frac{\mathsf{L}_{i,t-1}}{P_{i,t-1}}, \quad \text{and} \quad \mathsf{R}_{t-1}^{g} = \frac{\mathsf{L}_{t-1}^{g}}{P_{t-1}^{g}}. \tag{3.23}$$

The *UBI predicted* loss rate change $\mathsf{R}_{i,t}^{\Delta,\mathrm{UB}}$, *UBI predicted* individual loss ratio $\mathsf{R}_{i,t}^{y}$, and *UBI predicted* group loss ratio $\mathsf{R}_{t}^{y,g}$ are respectively

$$\mathsf{R}_{i,t}^{\Delta,\mathrm{UB}} = \frac{\mathsf{R}_{i,t}^{y} - \mathsf{R}_{t}^{y,g}}{\mathsf{R}_{t}^{y,g}}, \quad \mathsf{R}_{i,t}^{y} = \frac{\hat{y}_{i,t}}{P_{i,t}}, \quad \text{and} \quad \mathsf{R}_{t}^{y,g} = \frac{\hat{y}_{t}^{g}}{P_{t}^{g}}.$$

The credibility factor F is the weight of the best linear combination between the premium estimate $(1 + \mathsf{R}_{i,t}^{\Delta}) \times \bar{P}_t^g$ using the sample data to the premium estimate $\bar{P}_t^*$ using all data or data from another source to improve the reliability of the premium estimate P_{it}. The credibility factor increases with the business size and, hence, the number of drivers in the sample. Dean (1997) provided some methods to estimate F and suggested full credibility $F = 1$ when the sample size N is large enough such as above 10000 in an example. As this requirement is fulfilled for the telematic data with size $N = 14157$ and all data are used to estimate the chosen PMA model, full credibility $F = 1$ is applied. In cases where insured vehicles are less in the sample, the credibility factor F may vary, and external data sources may be used to improve

the reliability of the premium estimate. Furthermore, with the selected PMA model capable of classifying drivers, the premium calculation can focus on the categorized driver groups, leading to a more accurate premium calculation.

An example is provided to illustrate the experience-rating method and its extension to UBI. Consider driver i is classified as a safe driver ($g = 1$) in a driving test and seeks to purchase auto insurance for the upcoming period ($n_{it} = 1$). As summarized in Table 3.8, the annual premium for the safe group is $\bar{P}_t^1 = \$300$ and for the risky group is $\bar{P}_t^2 = \$500$. Driver i has recorded $\mathsf{L}_{i,t-1} = 0.2$ annual claim frequency and paid annual premium $P_{i,t-1} = \$500$ before. The safe group has recorded average $\mathsf{L}_{t-1}^1 = 0.1$ annual claim frequency and paid annual premium $P_{t-1}^1 = \$310$ per driver before. The risky group has recorded average $\mathsf{L}_{t-1}^2 = 0.3$ claims/loss and paid $P_{t-1}^2 = \$510$ annual premium per driver before. Driver i has more claims than the average of a safe group before. According to this historical claim frequencies, driver i is expected to exhibit a relatively higher risk profile compared to the average of the safe group. Therefore, it is appropriate for him to pay a higher premium.

To illustrate the UBI experience-rating method, additional assumptions about the predicted annual claim frequencies for driver i are added to the last row of Table 3.8. Assume that driver i has predicted annual claim frequency $\hat{y}_{i,t-1} = 0.15$ before, that of the safe group is $\hat{y}_{t-1}^1 = 0.105$ and that of the risk group is $\hat{y}_{t-1}^2 = 0.305$. This suggests that driver i operates his vehicle more safely than his historical claims indicate. This information is summarised in Table 3.8.

Table 3.8 – Assumptions summary in a case study in thousand dollars

	Driver i (safe)	Safe group	Risky group
Average annual premium $\bar{P}_t^g$	-	0.3	0.5
Historical annual premium $P_{i,t-1}, P_{t-1}^g$	0.5	0.31	0.51
Historical annual claims $L_{i,t-1}, L_{t-1}^g$	0.2	0.1	0.3
Predicted annual claim frequencies $\hat{y}_{i,t-1}, \hat{y}_{t-1}^g$	0.15	0.105	0.305

Taking the policy parameter $\varkappa = 1$, the UBI experience-rating premium is given by

$$P_{it,1} = (1 + \mathsf{R}^{\Delta}_{i,t}) \times \bar{P}^{g}_{t} \times n_{it} \times F = (1 + 0.1260) \times 300 \times 1 \times 1 = \$337.80$$

using (3.21) where the *historical* loss rate change $\mathsf{R}^{\Delta}_{i,t-1}$, the historical loss ratio $\mathsf{R}_{i,t-1}$ for driver i, the historical loss ratio for safe group R^{1}_{t-1}, the *UBI predicted* loss rate change $\mathsf{R}^{\Delta,\mathrm{UB}}_{i,t-1}$, the UBI predicted loss ratio $\mathsf{R}^{y}_{i,t}$ for driver i, and the UBI predicted loss ratio $\mathsf{R}^{y,1}_{t}$ for the safe group are respectively

$$\mathsf{R}^{\Delta}_{i,t} = \mathsf{R}^{\Delta,\mathrm{H}}_{i,t-1} + 1 \times \mathsf{R}^{\Delta,\mathrm{UB}}_{i,t} = 0.2403 - 0.1143 = 0.1260, \tag{3.24}$$

$$\mathsf{R}^{\Delta,\mathrm{H}}_{i,t-1} = \frac{\mathsf{R}_{i,t-1} - \mathsf{R}^{1}_{t-1}}{\mathsf{R}^{1}_{t-1}} = \frac{0.4 - 0.3225}{0.3225} = 0.2403,\ \mathsf{R}_{i,t-1} = \frac{\mathsf{L}_{i,t-1}}{P_{i,t-1}} = \frac{0.2}{0.5} = 0.4,\ \mathsf{R}^{1}_{t-1} = \frac{\mathsf{L}^{1}_{t-1}}{P^{1}_{t-1}} = \frac{0.1}{0.31} = 0.3225 \tag{3.25}$$

$$\mathsf{R}^{\Delta,\mathrm{UB}}_{i,t} = \frac{\mathsf{R}^{y}_{i,t} - \mathsf{R}^{y,1}_{t}}{\mathsf{R}^{y,1}_{t}} = \frac{0.3 - 0.3387}{0.3387} = -0.1143,\ \mathsf{R}^{y}_{i,t} = \frac{\hat{y}_{i,t}}{P_{i,t-1}} = \frac{0.15}{0.5} = 0.3,\ \mathsf{R}^{y,1}_{t} = \frac{\hat{y}^{1}_{t}}{P^{1}_{t}} = \frac{0.105}{0.31} = 0.3387. \tag{3.26}$$

So, the premium for driver i using the UBI experience-rating method is \$337.80. This premium is higher than the premium $\bar{P}^{1}_{t} = \$300$ for the safe group because the loss ratio for driver i is higher relative to the overall ratio in the safe group using historical claims. However, his current loss ratio due to current safe driving reduces the adverse effect due to the higher historical claims.

However, it is acknowledged that not all insured vehicles are equipped with telematic devices, potentially leading to data gaps in telematics insights. In response to this challenge, the *UBI policy* parameter $\varkappa$ in (3.22) can be set to 0. This adaptation to the UBI pricing model in (3.21) also allows the application to newly insured drivers with only historical records (traditional demographic variables). This premium, referred to as *historical experience-rating* premium for driver i during period t is represented as

$$P_{it,0} = (1 + \mathsf{R}^{\Delta,\mathrm{H}}_{i,t-1}) \times \bar{P}^{1}_{t} \times n_{i} \times F = (1 + 0.24031) \times 300 \times 1 \times 1 = \$372.09$$

where the *historical* loss rate change $\mathsf{R}^{\Delta}_{i,t-1}$ is given by (3.25). This loss rate change

can capture individual differences within a claim group using historical claims but fails to reflect the recent driving risk. Hence, this premium is higher than the UBI experience-rating premium calculated using historical and current driving experience. Thus, the historical experience-rating method is unable to provide immediate compensation/reward for safe driving.

Moreover, the UBI premium can track driving behavior more frequently and closely using regularly updated claim class and annual claim frequency prediction $\hat{y}_{i,t}$. The updating period can be reduced to monthly or even weekly to provide more instant feedback using the live telematic data. In summary, the proposed UBI experience-rating premium provides a correction of loss rate change $\mathsf{R}_{i,t}^{\Delta}$ of the experience-rating only premium using the sum of both *historical* loss rate change $\mathsf{R}_{i,t-1}^{\Delta,\mathrm{H}}$ and *UBI predicted* loss rate change $\mathsf{R}_{i,t}^{\Delta,\mathrm{UB}}$. The proposed PMA model enables the more immediate prediction of annual claim frequencies $\hat{y}_{i,t}$ using live telematic data. Consequently, the UBI premium can be updated more frequently, offering incentives for safe driving. The proposed UBI experience-rating premium introduces an incremental innovation to business processes, allowing for a gradual transition to the new UBI regime. This transition involves adjusting the UBI policy factor $\varkappa$ in Equation (3.22) so that $\varkappa$ can incrementally increase from 0 to 1 if driver i desires his premium to gradually reflect his current driving behavior.

3.6 Conclusions

The claim data from 14157 drivers has equi-dispersion and 92% of zero. The two-stage TP, PM, and ZIP regressions are proposed with lasso, elastic net, adaptive lasso, and adaptive elastic regularization to predict annual claims, select predictive DVs, and classify drivers into low claim (safe driver) and high claim (risky driver) groups. To ensure robustness, 100 resampling iterations of 70% drivers are performed for all TP, PM, and ZIP models. However, regularized ZIP regression does not improve model fit despite the high proportion of zeros in the data. Besides, very few DVs are selected for the structural zero claim group, which also has weak interpretations regarding

safe and risky driving. This finding agrees with Tang et al. (2014) and suggests a possible reason due to the lower signal-to-noise ratio in the structural zero part than in the Poisson part. Empirical results demonstrate that adaptive lasso with elastic net regularization provides better performance for TP models.

Lastly, based on the best PMA model, a UBI experience-rating method is proposed for premium pricing. This method rewards drivers with lower premiums based on recent good driving rather than claim history, which does not necessarily reflect good driving. Moreover, the premium is only revised annually for the traditional premium pricing. The UBI premium pricing system can update premiums more frequently based on recent driving and provide instant rewards. Hence, it encourages good and less driving, reduces cross-subsidizing of risky drivers, and allows better loss reserving for auto insurance companies.

For future work, the number of driver groups can be extended to three or more to capture more driving styles and predict better claim liability. For example, between the two extremes of safe and risky driver groups, a middle driver group can exist, exhibiting driving behavior distinct from either extreme. Such an extension can increase models' predictive power and monitor closer different driving behaviors. Similar mixture models and regularization techniques can still be applied for modeling the multiple mixture components. However, the identification and interpretation of these groups can be more challenging. For example, the third group can be merely a mix of the other two groups or an honest distinct group. Moreover, the label-shifting problem can be more severe for the mixture model with multiple groups.

Unlike lasso regularization, which focuses on selecting important DVs, neural networks take a unique approach. They leverage hidden layers to encapsulate the intricate dynamics of DVs, incorporating various weights and interaction terms. This novel method represents an exciting exploration in telematics data analysis, and its intriguing details will be unveiled in the upcoming Chapter 4.

3.7 Appendix

3.7.1 Working Implementation in R

1. This study utilizes R commands `glm` to fit Poisson regression and `glmnet` to fix Poisson regression with lasso regularization (Zeileis et al., 2008). The latter command begins with adopting the R function `sparse.model.matrix` as
   `data_feature <- sparse.model.matrix(~., dt_feature)`
   The argument `penalty.factor` in `cv.glmnet` is used for adaptive lasso. However, `glmnet` package does not provide a $p-$value for selected DVs. Therefore, the $p-$value is extracted by refitting the model with the selected DVs using `glm` procedure.

2. The $R = 100$ simulated data sets are employed in stages 1 and 2 of TP and PM models to investigate the optimal α in the elastic net regularization. A 10-fold cross-validation strategy is established. The `caret` package in `R` is utilized, employing the `train()` function with the method set to "glmnet" for fitting the elastic net. Refer to commands 3.1 below.

Code Listing 3.1 – Finding optimal α in R

```
XX = model.matrix(Claims ~ . -EXP-1,data=stage1)
  YY = stage1$Claims
  OFF = log(stage1$EXP)
  Fit_stage1 <- caret::train(
    x = cbind(XX,OFF), y = YY,
    method = "glmnet", family = "poisson",
    tuneLength = 10,
    trControl = trainControl(method="cv", number = 10,
       repeats = 100)
  )
```

3. The calculation of the AUC is performed using the `roc()` function from the `pROC` package. Additionally, the `latex2exp` package facilitates the creation of

ROC plots.

4. The `AER` package in `R` using built-in command `dispersiontest()` is utilized to perform the hypothesis test with the alternative hypothesis $H_1 : \text{Var}(Y_i) = \mu_i + \Psi \times \text{trafo}(\mu_i)$ where the transformation function $\text{trafo}(\mu_i) = \mu_i$ (by default, `trafo = NULL`) corresponds to Poisson model with $\text{Var}(Y_i) = (1 + \Psi)\mu_i$. If the dispersion $1 + \Psi$ is greater than 1, it indicates overdispersion.

5. PM regression model is estimated using the `R` package `FLXMRglmnet()` allows fitting mixtures of GLMs where the coefficients are penalized using lasso, adaptive lasso, and elastic net by reusing the functionalities from package `glmnet` (Leisch, 2004). When the PM model is refitted using the selected DVs, the `R` package `flexmix`, which supports cross-validation, prediction, and evaluation of various goodness-of-fit measures, is used. With the selected DVs for low and high claim groups, `FLXMRglmfix()` refits the model and provides the significance of coefficients and predicted values.

6. ZIP regression model is estimated using the `zipath()` function for lasso and elastic-net regularization and `ALasso()` function for adaptive lasso regularization from the `mpath` and `AMAZonn` packages. The optimal lambda minimum is searched via 10-fold cross-validation with `cv.zipath()` and applied to both fitted models, ZIPL and ZIPA, for $R = 100$ subsamples, each with 70% data. Full data are refited to the PM model based on the selected DVs using Poisson `zeroinf`.

3.7.2 Parameter estimate of all models

Table 3.9 – Parameter estimates $\boldsymbol{\beta}$ in (3.17) for stage-1 TP models before refit with $R = 100$ subsamples of 70% data using Poisson `glmnet`, $\boldsymbol{\beta}^{\mathrm{T1}}$ after refitting to full data using Poisson `glm` on the selected DVs with $I_j^{\mathrm{T2}} > 62$ (otherwise dropped as indicated in grey highlight), and selection criteria $\boldsymbol{I}^{\mathrm{T1}}$ in (3.19). There are $J^{\mathrm{T1}} = 52$ DVs selected for TPL-1 and $J^{\mathrm{T1}} = 39$ DVs selected for TPA-1 under columns β_j^{T1}. The bold with yellow highlighted under β_j^{T1} are significant.

TPL-1

	glmnet with 100 repeats				glm		glmnet with 100 repeats				glm		glmnet with 100 repeats				glm
Measures	MSE	MAE	Deviance		Poisson	Measures	MSE	MAE	Deviance		Poisson	Measures	MSE	MAE	Deviance		Poisson
DVs	I_j^{T1}			β_j	β_j^{T1}	DVs	I_j^{T1}			β_j	β_j^{T1}	DVs	I_j^{T1}			β_j	β_j^{T1}
1	-	34	7	-0.0029	-	**28**	3	126	61	-0.0031	-	**55**	-	119	68	0.0082	0.0134
2	89	232	228	0.0176	0.0159	**29**	227	276	279	0.0279	**0.0360**	**56**	37	136	123	0.0149	0.0061
3	180	317	337	0.0402	**0.0619**	**31**	251	324	337	0.0409	0.0535	**57**	139	310	334	-0.0446	-0.1696
4	140	307	334	-0.0409	**-0.0987**	**32**	302	327	341	0.0474	**0.0513**	**58**	24	147	109	0.0115	0.0095
5	3	109	61	0.0085	-	**33**	133	273	266	0.0256	0.0393	**59**	68	252	229	0.0282	0.0448
7	-	136	116	-0.0021	-0.0830	**34**	-	85	20	-0.0073	-	**60**	95	198	191	-0.0243	-0.0091
8	7	95	37	-0.0011	-	**35**	146	245	242	0.0222	0.0168	**61**	255	317	320	0.0426	0.0626
9	272	310	320	0.0417	0.0546	**36**	262	320	334	0.0576	**0.0797**	**63**	98	242	235	0.0219	0.0346
10	-	17	-	-	-	**37**	292	327	334	0.0424	**0.0518**	**64**	30	194	160	-0.0264	-0.0348
13	10	140	72	-0.0154	-0.0253	**38**	184	290	289	0.0238	0.0263	**65**	-	99	41	-0.0084	-
14	-	105	14	-0.0031	-	**39**	71	232	228	0.0122	0.0160	**66**	41	164	133	0.0164	0.0173
15	14	119	68	-0.0190	-0.0065	**43**	78	204	204	-0.0381	-0.0505	**67**	329	330	341	-0.1400	**-0.1706**
16	3	113	65	0.0001	-0.0004	**44**	3	112	41	-0.0113	-	**68**	17	102	61	-0.0135	-
18	316	327	341	0.0969	**0.1254**	**45**	3	78	48	0.0172	-	**69**	17	188	140	-0.0166	-0.0320
19	-	85	20	0.0021	-	**46**	31	194	164	0.0210	0.0361	**71**	78	231	224	0.0172	0.0185
20	173	310	323	0.0363	0.0563	**47**	177	314	330	-0.0611	**-0.0918**	**72**	187	303	297	0.0319	0.0418
22	41	205	177	0.0133	0.0309	**49**	95	245	252	-0.0341	-0.0448	**73**	302	327	341	0.0587	**0.0743**
23	-	133	75	-0.0051	-0.0235	**50**	72	228	218	0.0157	0.0205	**74**	102	242	242	0.0216	0.0350
24	136	272	262	0.0213	0.0236	**51**	116	289	306	-0.0517	-0.0944	**75**	336	330	341	0.0621	**0.0659**
25	-	119	58	-0.0087	-	**52**	150	307	324	-0.0397	-0.0623	**76**	48	239	235	-0.0236	-0.0565
26	58	160	139	0.0129	0.0024	**53**	65	174	157	0.0156	0.0107	**77**	157	307	324	0.0324	0.0549
27	17	160	129	-0.0205	-0.0402	**54**	163	262	272	0.0209	0.0208						

TPA-1

	glmnet with 100 repeats				glm		glmnet with 100 repeats				glm		glmnet with 100 repeats				glm
Measures	MSE	MAE	Deviance		Poisson	Measures	MSE	MAE	Deviance		Poisson	Measures	MSE	MAE	Deviance		Poisson
DVs	I_j^{T1}			β_j	β_j^{T1}	DVs	I_j^{T1}			β_j	β_j^{T1}	DVs	I_j^{T1}			β_j	β_j^{T1}
1	3	41	24	-0.0712	-	**28**	-	79	-	-	-	**55**	-	41	20	0.0030	-
2	61	95	68	0.0360	0.0149	**29**	228	279	276	0.0311	**0.0357**	**56**	14	99	61	0.0311	-
3	160	317	310	0.0512	**0.0608**	**31**	217	316	313	0.0514	**0.0536**	**57**	78	306	327	-0.0797	-0.1726
4	89	296	310	-0.0630	**-0.1035**	**32**	319	337	340	0.0552	**0.0503**	**58**	-	38	3	0.0297	-
5	-	61	10	0.0482	-	**33**	78	248	214	0.0352	0.0363	**59**	31	204	139	0.0374	0.0478
7	-	24	-		-	**34**	-	38	17	-0.0201	-	**60**	58	143	126	-0.0288	-0.0093
8	-	34	13	-0.0067	-	**35**	102	190	177	0.0258	0.0199	**61**	163	283	282	0.0595	**0.0702**
9	187	300	289	0.0517	0.0579	**36**	248	334	330	0.0703	**0.0775**	**63**	41	194	150	0.0358	0.0422
10	-	-	-	-	-	**37**	285	334	330	0.0513	**0.0530**	**64**	27	143	89	-0.0551	-0.0317
13	-	48	20	-0.0144	-	**38**	160	231	218	0.0298	0.0271	**65**	-	17	10	-0.0100	-
14	-	62	-	-	-	**39**	7	116	71	0.0132	0.0163	**66**	17	109	55	0.0221	-
15	3	86	31	-0.0200	-	**43**	48	235	204	-0.0577	-0.0510	**67**	336	340	340	-0.1752	**-0.1686**
16	-	14	3	-0.0088	-	**44**	-	44	7	-0.0294	-	**68**	-	85	55	-0.0201	-
18	333	340	340	0.1212	**0.1205**	**45**	3	72	44	0.0293	-	**69**	7	99	34	-0.0334	-
19	10	65	17	0.0146	-	**46**	10	130	51	0.0410	-	**71**	37	129	102	0.0291	0.0204
20	112	286	272	0.0426	0.0567	**47**	170	327	327	-0.0773	**-0.0913**	**72**	156	269	248	0.0479	0.0443
22	20	143	85	0.0300	0.0282	**49**	58	194	187	-0.0470	-0.0367	**73**	289	340	337	0.0710	**0.0733**
23		55	14	-0.0171	-	**50**	27	129	92	0.0259	0.0206	**74**	51	228	167	0.0301	0.0341
24	58	188	147	0.0237	0.0230	**51**	51	282	262	-0.0718	-0.0918	**75**	316	340	333	0.0748	**0.0709**
25	-	34	3	-0.0843	-	**52**	136	306	303	-0.0493	-0.0618	**76**	41	225	184	-0.0446	-0.0565
26	21	68	38	0.0201	-	**53**	10	71	54	0.0182	-	**77**	116	290	273	0.0457	0.0554
27	10	153	105	-0.0349	-0.0359	**54**	109	176	183	0.0259	0.0256						

Table 3.10 – Parameter estimates $\boldsymbol{\beta}^{\mathrm{T2}}_{\mathrm{L}h}, \boldsymbol{\beta}^{\mathrm{T2}}_{\mathrm{H}h}$ for the stage-2 TP models with $R = 100$ subsamples of 70% data. Parameters are based on $J^{\mathrm{T1}} = 52$ DVs in stage 1 and J^{T}_{2} refers to the number of frequently selected DVs with $I_{\mathrm{L}hj} > 43$ ($\tau_{0.08}$), 49 ($\tau_{0.09}$), 53 ($\tau_{0.10}$) and 56 ($\tau_{0.11}$); and $I_{\mathrm{H}hj} > 19$, 13, 9 and 6 respectively which differ across threshold. Significant $\beta^{\mathrm{T2}}_{\mathrm{L}hj}, \beta^{\mathrm{T2}}_{\mathrm{H}hj}$ are in bold face with yellow highlighted.

$\tau_{0.08}$: TPLA-2						$\tau_{0.09}$: TPLA-2						$\tau_{0.10}$: TPLN-2						$\tau_{0.11}$: TPLN-2					
Groups	Low			High			Low			High			Low			High			Low			High	
$\mathcal{R}_h$	0.70			0.30			0.79			0.21			0.85			0.15			0.90			0.10	
I_{hj}	43			19			49			13			53			9			56			6	
J^{T}_{2}	17			22			38			14			42			23			43			28	
DVs	$I_{\mathrm{L}hj}$	$\beta^{\mathrm{T2}}_{\mathrm{L}hj}$	DVs	$I_{\mathrm{H}hj}$	$\beta^{\mathrm{T2}}_{\mathrm{H}hj}$	DVs	$I_{\mathrm{L}hj}$	$\beta^{\mathrm{T2}}_{\mathrm{L}hj}$	DVs	$I_{\mathrm{H}hj}$	$\beta^{\mathrm{T2}}_{\mathrm{H}hj}$	DVs	$I_{\mathrm{L}hj}$	$\beta^{\mathrm{T2}}_{\mathrm{L}hj}$	DVs	$I_{\mathrm{H}hj}$	$\beta^{\mathrm{T2}}_{\mathrm{H}hj}$	DVs	$I_{\mathrm{L}hj}$	$\beta^{\mathrm{T2}}_{\mathrm{L}hj}$	DVs	$I_{\mathrm{H}hj}$	$\beta^{\mathrm{T2}}_{\mathrm{H}hj}$
2	-	-	**2**	42	0.0277	**2**	-	-	**2**	19	0.0169	**2**	-	-	**2**	18	0.0108	**2**	-	-	**2**	5	0.0250
3	8	0.0093	**3**	31	0.0356	**3**	52	0.0331	**3**	-	-	**3**	116	0.0515	**3**	5	0.0235	**3**	118	0.0403	**3**	-	-
4	49	-0.0567	**4**	42	0.0366	**4**	254	**-0.0714**	**4**	43	0.0466	**4**	356	**-0.0748**	**4**	76	0.0864	**4**	357	**-0.0833**	**4**	129	0.1443
7	-	-	**7**	-	-	**7**	-	-	**7**	-	-	**7**	-	-	**7**	-	-	**7**	-	-	**7**	-	-
9	95	0.0636	**9**	3	-0.0081	**9**	350	**0.0976**	**9**	3	-0.0416	**9**	348	**0.0953**	**9**	10	-0.0421	**9**	357	**0.0859**	**9**	15	-0.0528
13	4	-0.0882	**13**	48	0.0410	**13**	66	-0.0648	**13**	87	0.0824	**13**	243	-0.0513	**13**	91	0.0814	**13**	278	-0.0545	**13**	83	0.0667
15	11	-0.0010	**15**	6	-0.0205	**15**	26	-0.0193	**15**	-	-	**15**	44	-0.0643	**15**	-	-	**15**	22	-0.0286	**15**	8	0.0355
16	11	-0.0584	**16**	17	0.0252	**16**	33	-0.0353	**16**	68	0.0424	**16**	22	-0.0077	**16**	55	0.0426	**16**	11	-0.0097	**16**	15	0.0219
18	46	0.0820	**18**	36	0.0608	**18**	210	**0.0827**	**18**	3	0.0867	**18**	323	**0.0764**	**18**	-	-	**18**	343	**0.1031**	**18**	-	-
20	15	-0.0056	**20**	45	0.0351	**20**	40	0.0428	**20**	-	-	**20**	207	0.0437	**20**	-	-	**20**	171	0.0363	**20**	3	0.0768
22	72	-0.0633	**22**	89	0.0407	**22**	15	-0.0469	**22**	8	0.0272	**22**	36	-0.0138	**22**	5	0.0174	**22**	47	-0.0218	**22**	18	0.0432
23	11	0.0060	**23**	6	0.0196	**23**	33	0.0084	**23**	3	-0.0091	**23**	101	0.0292	**23**	8	-0.0442	**23**	139	0.0280	**23**	40	-0.0761
24	35	0.0645	**24**	11	0.0316	**24**	195	0.0727	**24**	3	0.0456	**24**	348	**0.0904**	**24**	-	-	**24**	339	**0.0844**	**24**	-	-
26	113	0.0577	**26**	6	-0.0608	**26**	199	0.0516	**26**	11	-0.0282	**26**	185	0.0377	**26**	3	-0.0094	**26**	154	0.0270	**26**	8	-0.0363
27	61	-0.0502	**27**	34	0.0476	**27**	95	-0.0422	**27**	5	-0.0264	**27**	214	-0.0478	**27**	10	-0.0320	**27**	75	-0.0287	**27**	33	-0.0863
29	12	-0.0749	**29**	95	**0.0211**	**29**	48	-0.0497	**29**	32	**0.0179**	**29**	163	0.0476	**29**	11	-0.0494	**29**	157	0.0358	**29**	13	0.0148
31	12	0.0563	**31**	17	0.0219	**31**	74	0.0495	**31**	3	0.0463	**31**	287	0.0555	**31**	5	0.0323	**31**	286	0.0549	**31**	-	-
32	46	0.0614	**32**	61	0.0320	**32**	332	**0.1115**	**32**	-	-	**32**	345	**0.0893**	**33**	8	-0.0312	**32**	332	**0.0801**	**32**	-	-
33	23	0.0492	**33**	-	-	**33**	147	0.0523	**33**	-	-	**33**	309	0.0569	**33**	-	-	**33**	264	0.0492	**33**	5	-0.0615
35	-	-	**35**	25	0.0343	**35**	85	0.0277	**35**	-	-	**35**	127	0.0265	**35**	-	-	**35**	61	0.0227	**35**	5	0.0954
36	69	0.0575	**36**	3	0.0776	**36**	137	0.0411	**36**	-	-	**36**	264	0.0536	**36**	-	-	**36**	318	**0.0591**	**36**	3	-0.0304
37	243	0.1047	**37**	14	0.0276	**37**	354	**0.1337**	**37**	-	-	**37**	355	**0.1239**	**37**	-	-	**37**	357	**0.1153**	**37**	-	-
38	38	-0.0502	**38**	22	0.0304	**38**	78	-0.0596	**38**	16	0.0401	**38**	127	-0.0437	**38**	39	0.0386	**38**	25	-0.0237	**38**	18	0.0239
39	87	-0.0652	**39**	-	-	**39**	251	-0.0819	**39**	-	-	**39**	327	-0.1015	**39**	3	0.0148	**39**	132	-0.1482	**39**	-	-
43	27	-0.0380	**43**	6	0.0150	**43**	59	-0.0501	**43**	13	-0.0375	**43**	65	-0.0265	**43**	26	-0.0868	**43**	82	-0.0289	**43**	83	-0.0808
46	4	-0.0310	**46**	67	0.0310	**46**	15	0.0088	**46**	5	0.0376	**46**	26	0.0188	**46**	-	-	**46**	36	0.0345	**46**	10	0.0868
47	60	-0.0564	**47**	6	-0.0103	**47**	225	-0.0711	**47**	3	-0.0745	**47**	341	**-0.0804**	**47**	-	-	**47**	321	-0.0713	**47**	-	-
49	15	-0.0375	**49**	-	-	**49**	59	-0.0326	**49**	11	-0.0564	**49**	54	-0.0219	**49**	75	-0.0759	**49**	193	-0.0381	**49**	26	-0.0744
50	15	-0.0394	**50**	11	0.0320	**50**	29	-0.0543	**50**	-	-	**50**	87	-0.0308	**50**	13	0.0291	**50**	64	-0.0244	**50**	22	0.0360
51	152	-0.0853	**51**	150	0.0625	**51**	206	-0.0769	**51**	8	0.0623	**51**	214	-0.0555	**51**	-	-	**51**	314	**-0.0776**	**51**	7	0.0969
52	4	-0.0013	**52**	45	**-0.0826**	**52**	151	-0.0387	**52**	8	-0.0763	**52**	268	-0.0416	**52**	16	-0.0202	**52**	293	-0.0462	**52**	7	-0.0014
53	152	0.0733	**53**	-	-	**53**	95	0.0407	**53**	3	0.0167	**53**	51	-0.0297	**53**	10	0.0507	**53**	110	0.0380	**53**	5	0.0094
54	34	0.0424	**54**	6	0.0210	**54**	56	0.0202	**54**	3	0.0047	**54**	47	0.0256	**54**	3	0.0332	**54**	232	0.0392	**54**	-	-
55	11	0.0340	**55**	-	-	**55**	158	0.0483	**55**	16	-0.0460	**55**	152	0.0354	**55**	13	-0.0540	**55**	228	0.0491	**55**	33	-0.1115
56	49	0.0443	**56**	3	-0.0516	**56**	122	0.0412	**56**	3	-0.0275	**56**	196	0.0327	**56**	13	-0.0398	**56**	132	0.0401	**56**	20	-0.0636
57	15	-0.0626	**57**	-	-	**57**	214	-0.0672	**57**	92	0.0740	**57**	337	**-0.0673**	**57**	49	-0.1974	**57**	346	**-0.0739**	**57**	45	-0.2011
58	49	0.0432	**58**	78	-0.0545	**58**	74	0.0290	**58**	57	-0.0548	**58**	91	0.0349	**58**	96	-0.0789	**58**	100	0.0280	**58**	98	-0.0941
59	65	0.0571	**59**	-	-	**59**	207	0.0566	**59**	33	-0.0480	**59**	294	**0.0557**	**59**	91	-0.0799	**59**	285	0.0552	**59**	50	-0.0947
60	50	-0.0413	**60**	8	0.0345	**60**	55	-0.0442	**60**	-	-	**60**	239	-0.0514	**60**	10	0.0615	**60**	221	-0.0514	**60**	10	0.0957
61	8	0.0465	**61**	31	0.0258	**61**	26	0.0135	**61**	32	0.0445	**61**	149	0.0485	**61**	-	-	**61**	146	0.0504	**61**	8	0.0182
63	8	-0.0378	**63**	67	0.0422	**63**	4	-0.0054	**63**	-	-	**63**	15	-0.0068	**63**	16	0.0675	**63**	50	0.0219	**63**	3	0.0811
64	26	-0.0629	**64**	-	-	**64**	88	-0.0593	**64**	-	-	**64**	142	-0.0489	**64**	-	-	**64**	211	-0.0458	**64**	35	0.0833
66	38	0.0499	**66**	-	-	**66**	225	0.0496	**66**	-	-	**66**	254	0.0416	**66**	-	-	**66**	211	0.0400	**66**	3	-0.0610
67	31	-0.0420	**67**	176	**-0.1401**	**67**	310	**-0.0912**	**67**	187	-0.1461	**67**	363	**-0.1275**	**67**	112	-0.0977	**67**	357	**-0.1418**	**67**	97	-0.1485
69	19	-0.0407	**69**	76	0.0508	**69**	121	-0.0523	**69**	-	-	**69**	98	-0.0353	**69**	-	-	**69**	46	-0.0232	**69**	5	-0.0503
71	30	0.0396	**71**	6	-0.0682	**71**	192	0.0464	**71**	8	-0.0405	**71**	175	0.0328	**71**	3	-0.0374	**71**	100	0.0228	**71**	2	-0.0005
72	8	0.0326	**72**	11	0.0267	**72**	41	0.0265	**72**	-	-	**72**	76	0.0236	**72**	-	-	**72**	104	0.0219	**72**	10	0.0867
73	42	0.0594	**73**	50	0.0344	**73**	185	0.0747	**73**	5	0.0291	**73**	268	0.0709	**73**	5	0.0354	**73**	321	**0.0709**	**73**	2	0.0015
74	-	-	**74**	17	0.0248	**74**	122	0.0458	**74**	3	0.0242	**74**	276	**0.0611**	**74**	-	-	**74**	188	0.0446	**74**	-	-
75	15	0.0488	**75**	107	0.0456	**75**	254	0.0707	**75**	16	0.0436	**75**	301	0.0730	**75**	3	0.0451	**75**	332	**0.0844**	**75**	-	-
76	27	-0.0662	**76**	6	-0.0343	**76**	151	-0.0588	**76**	5	-0.0035	**76**	276	-0.0549	**76**	10	0.0779	**76**	271	-0.0491	**76**	27	0.0867
77	7	0.0400	**77**	14	0.0209	**77**	26	0.0373	**77**	5	0.0330	**77**	36	0.0075	**77**	24	0.0574	**77**	100	0.0391	**77**	5	0.0448

Table 3.11 – Parameter estimates $\boldsymbol{\beta}_0$, $\boldsymbol{\beta}_c$ for ZIP model before refit and $\boldsymbol{\beta}^{\mathrm{M}}_{\mathrm{L}}, \boldsymbol{\beta}^{\mathrm{M}}_{\mathrm{H}}$ for PM and $\boldsymbol{\beta}^{\mathrm{Z}}_0$, $\boldsymbol{\beta}^{\mathrm{Z}}_c$ for ZIP models after refitted to all data based on selected DVs and selection criteria $\boldsymbol{I}^{\mathrm{M}}_{\mathrm{L}}$, $\boldsymbol{I}^{\mathrm{M}}_{\mathrm{H}}$, $\boldsymbol{I}^{\mathrm{Z}}_0$, $\boldsymbol{I}^{\mathrm{Z}}_{\mathrm{c}}$ with $R = 100$ sub-samples of 70% data. For PM model, $J^{\mathrm{M}}_{\mathrm{L}}, J^{\mathrm{M}}_{\mathrm{H}}$ refer to the number of frequently selected DVs with $I^{\mathrm{M}}_{\mathrm{L}j} > 43$ (PML), 45 (PMA), and $I^{\mathrm{M}}_{\mathrm{H}j} > 19$ (PML), 17 (PMA); otherwise they are dropped as in grey highlight. For ZIP model, $J^{\mathrm{Z}}_0, J^{\mathrm{Z}}_{\mathrm{c}}$ refer to the number of frequently selected DVs with $I^{\mathrm{Z}}_{0j} > 62$ and $I^{\mathrm{Z}}_{\mathrm{c}j} > 62$; otherwise, β_{0j} and $\beta_{\mathrm{c}j}$ are excluded respectively as in grey highlight. Significant parameters $\beta^{\mathrm{M}}_{\mathrm{L}j}, \beta^{\mathrm{M}}_{\mathrm{H}j}, \beta^{\mathrm{Z}}_{0j}, \beta^{\mathrm{Z}}_{\mathrm{c}j}$ are boldfaced and yellow highlighted.

PML					PMA					ZIPA						
I^{M}_j	43		19		I^{M}_j	45		17		I^{Z}_j	62			62		
J^{M}	45		18		J^{M}	39		40		J^{Z}	4			45		
DVs	Low		High		DVs	Low		High		DVs	Zero			Count		
	$I^{\mathrm{M}}_{\mathrm{L}j}$	$\beta^{\mathrm{M}}_{\mathrm{L}j}$	$I^{\mathrm{M}}_{\mathrm{H}j}$	$\beta^{\mathrm{M}}_{\mathrm{H}j}$		$I^{\mathrm{M}}_{\mathrm{L}j}$	$\beta^{\mathrm{M}}_{\mathrm{L}j}$	$I^{\mathrm{M}}_{\mathrm{H}j}$	$\beta^{\mathrm{M}}_{\mathrm{H}j}$		I^{Z}_{0j}	β_{0j}	β^{Z}_{0j}	$I^{\mathrm{Z}}_{\mathrm{c}j}$	$\beta_{\mathrm{c}j}$	$\beta^{\mathrm{Z}}_{\mathrm{c}j}$
3	118	0.0182	47	0.0983	**3**	71	0.0380	126	0.1182	**3**	44	-0.0022	-	291	0.0404	**0.0514**
9	88	-0.0003	20	0.0275	**9**	27	-0.0226	16	0.0106	**9**	20	-0.0047	-	172	0.0170	0.0450
18	324	**0.0636**	27	**0.0477**	**18**	206	**0.0877**	149	0.1078	**18**	-	-	-	88	0.0044	**0.1352**
19	311	0.0491	3	0.1058	**19**	200	0.0877	82	0.0821	**19**	10	-0.0003	-	136	0.0050	-0.0263
20	78	-0.0197	37	0.0919	**20**	71	**-0.0443**	27	-0.0532	**20**	34	-0.0053	-	291	0.0552	**0.0717**
22	162	0.0098	54	0.0633	**22**	91	-0.0484	101	0.0839	**22**	47	-0.0104	-	217	0.0333	0.0441
23	335	-0.2076	24	-0.2634	**23**	219	-0.2801	159	-0.2732	**23**	-	-	-	84	-0.0025	-0.0247
26	250	0.0259	98	0.0370	**26**	212	0.0375	81	0.0119	**26**	3	1.91E-05	-	125	0.0069	0.0014
27	338	0.0450	3	0.0068	**27**	251	0.0755	60	0.0576	**27**	10	0.0004	-	339	-0.1900	-0.0495
29	324	**0.0355**	17	0.0633	**29**	172	0.0663	79	**0.0722**	**29**	24	-0.0005	-	267	0.0182	**0.0363**
31	294	0.0352	14	0.0390	**31**	165	0.0637	87	0.0496	**31**	20	-0.0024	-	234	0.0265	0.0515
33	138	-0.0150	-	-	**33**	50	-0.0516	13	-0.0265	**33**	55	-0.0123	-	213	0.0243	0.0426
34	287	0.0369	7	0.1254	**34**	127	0.0586	57	0.0533	**34**	3	-0.0001	-	88	-0.0022	-0.0234
35	331	0.0578	7	0.0130	**35**	299	0.1089	43	0.0735	**35**	20	-0.0029	-	132	0.0099	0.0232
36	335	0.0501	31	0.0450	**36**	214	0.0690	120	**0.0642**	**36**	30	-0.0103	-	281	0.0464	**0.0752**
37	304	**0.0284**	13	0.0522	**37**	183	0.0416	51	0.0567	**37**	10	-0.0009	-	298	0.0429	**0.0524**
38	230	-0.0516	7	0.0002	**38**	121	-0.1553	40	0.0017	**38**	98	-0.0265	-39.0618	121	0.0076	-0.0065
43	88	-0.0267	3	-0.0238	**43**	36	-0.0296	44	-0.0703	**43**	17	0.0008	-	173	-0.0397	-0.0471
44	57	0.0078	-	-	**44**	54	0.0575	37	0.0932	**44**	-	-	-	155	-0.0099	-0.0287
45	274	0.0541	24	0.0395	**45**	249	**0.1177**	79	0.0982	**45**	10	-0.0005	-	153	0.0090	0.0188
46	338	-0.0957	14	-0.0442	**46**	259	-0.1665	78	-0.1081	**46**	30	-0.0042	-	264	0.0569	0.0362
47	338	-0.1644	20	-0.0469	**47**	292	-0.2914	77	-0.1493	**47**	3	0.0000	-	322	-0.0843	**-0.0940**
49	249	0.0241	10	0.0235	**49**	91	0.0430	23	0.0378	**49**	165	0.0366	-0.0193	251	-0.1070	-0.0597
50	331	-0.0898	24	-0.0447	**50**	197	-0.1691	126	-0.1419	**50**	-	-	-	122	0.0077	0.0145
51	335	-0.0981	14	-0.0381	**51**	225	-0.1602	119	-0.1378	**51**	17	0.0004	-	301	-0.0865	-0.0868
52	314	0.0265	37	0.0265	**52**	93	0.0447	81	0.0538	**52**	10	0.0023	-	311	-0.0738	-0.0660
54	84	0.0134	-	-	**54**	40	0.0111	7	0.0496	**54**	17	-0.0013	-	213	0.0197	0.0182
55	112	0.0183	7	0.0196	**55**	33	0.0027	14	-0.0053	**55**	7	0.0001	-	88	0.0016	0.0109
56	142	0.0193	44	0.0636	**56**	33	0.0478	114	0.0780	**56**	3	-0.0003	-	75	0.0012	0.0002
58	240	0.0298	17	0.0632	**58**	154	**0.0718**	62	0.0692	**58**	3	-0.0001	-	128	0.0087	0.0157
59	294	-0.0433	17	-0.0817	**59**	115	-0.1431	73	**-0.1543**	**59**	24	-0.0035	-	200	0.0189	0.0409
60	318	0.0574	24	0.0934	**60**	183	0.1142	162	0.0945	**60**	13	0.0024	-	231	-0.0259	-0.0053
61	223	0.0297	3	0.0193	**61**	167	**0.0773**	37	0.0514	**61**	20	-0.0032	-	244	0.0403	0.0560
63	189	-0.0652	20	-0.0031	**63**	162	-0.1524	125	-0.0583	**63**	37	-0.0057	-	98	0.0088	0.0362
64	162	0.0132	7	0.0346	**64**	71	0.0231	7	0.0641	**64**	-	-	-	139	-0.0220	-0.0409
66	338	-0.1689	7	-0.0538	**66**	297	-0.3054	64	-0.1943	**66**	7	-0.0001	-	81	0.0050	0.0175
67	88	**-0.0263**	-	-	**67**	37	-0.0730	40	**0.0416**	**67**	20	0.0044	-	339	-0.1312	**-0.1510**
68	210	-0.0265	7	-0.0102	**68**	90	-0.0940	52	-0.1397	**68**	7	0.0001	-	67	-0.0013	-0.0041
69	199	0.0210	17	0.0611	**69**	95	0.0537	26	0.0420	**69**	7	0.0005	-	180	-0.0190	-0.0286
71	270	0.0388	14	0.0725	**71**	141	0.0869	76	0.0686	**71**	34	-0.0040	-	145	0.0098	0.0114
72	318	0.0473	183	0.1172	**72**	217	0.0774	194	0.0950	**72**	30	-0.0040	-	268	0.0338	0.0391
73	335	0.0631	88	**0.1193**	**73**	242	0.1072	109	**0.1064**	**73**	75	-0.0129	-0.1323	288	0.0391	**0.0617**
75	321	-0.0545	27	**-0.0622**	**75**	142	**-0.1177**	136	-0.1645	**75**	20	-0.0022	-	301	0.0511	**0.0634**
76	257	0.0361	10	0.0327	**76**	161	0.0722	83	0.0827	**76**	7	0.0009	-	244	-0.0380	-0.0656
77	284	0.0324	17	0.0370	**77**	170	0.0740	74	0.0615	**77**	98	-0.0189	-34.9726	149	0.0132	-0.0313

Chapter 4

Claim prediction using Poisson mixture neural network models

This chapter is a follow-up on Chapter 3, further researching auto insurance claim predictive models using telematics data. While Chapter 3 considers lasso regression to extract important DVs from telematics data, this chapter considers an alternative approach using neural networks (NNs) and deep learning NNs (DLNNs) with a multi-layer perceptron (MLP) or feed forward structure. MLP is the NN connecting multiple layers in a directed graph such that the signal path goes one way, also called feed-forward, whereas DLNN has three or more hidden layers between the input and output layers. The advantage of NNs stems from the ability to describe complicated non-linear feature patterns of DVs on claim counts. The idea of shrinking some DVs to reduce overfitting in lasso regularization can also be added to NNs. To classify drivers into safe and risky groups, Poisson mixture models are also applied. This chapter is organized as follows. Section 4.1 reviews the literature focusing on neural networks. Section 4.2 introduces the structures, loss functions, architecture search, etc, for DLNNs. Section 4.3 describes the empirical studies with technical details of imposing constraints to differentiate mixture components, estimating the non-network prior probability parameter, and implementing the PM DLNN using `Keras` and `Tensorflow` in `Python`. Section 4.4 presents experimental results of fitting

DLNNs to telematics data. Section 4.5 concludes the chapter with summaries of main contributions.

4.1 Introduction

Developing claim predictive models for auto insurance faces challenges attributed to data complications, policy and regulatory scrutiny, and privacy concerns. The exploration of methodologies for analyzing telematics data initially relied on traditional statistical models, but in recent years, there has been a notable shift towards machine learning (ML) techniques (Hanafy and Ming, 2021). Section 3.1 provides a complete review of ML techniques on loss models, including clustering, decision trees, random forest, gradient boosting, and NNs. These techniques offer distinct advantages over state-of-the-art GLMs. Its strength lies in capturing complicated non-linear relationships within the data, thereby enhancing the accuracy of predictive models for auto insurance.

Among the many studies on loss models employing distinct methodologies, this chapter focuses on NNs. NN emerges as an optimal method for modeling claim data due to its remarkable capacity for linear and non-linear mapping, surpassing the constraints of rigid model structures in traditional statistical models. NNs excel GLMs and lasso regression in learning empirically the intricate logic between input features and output targets throughout the training process. They adeptly construct linear and non-linear relationships, leveraging high-order interactions across layers to discern correlations between input and output variables. NNs proved highly proficient in recognizing complex patterns and relationships within extensive data sets. Fallah et al. (2009) compared the Poisson regressions model with Poisson NNs featuring six hidden layers for claim counts. Their evaluation across three simulated datasets, including one linear and two non-linear cases, revealed that Poisson NNs excel in the Poisson regression model in capturing nonlinear effects according to MSE.

For UBI claim prediction, the ability of NNs to incorporate customer-specific features, such as age, location, and driving behavior, enables efficient identification of concealed

driving patterns underlying auto insurance claims (Dong et al., 2016). Paefgen et al. (2013) conducted a study revealing that NNs exhibit superior performance compared to logistic regression and decision tree classifiers when utilizing 15 predictor variables to assess claim events. Kim et al. (2022) demonstrated that DLNNs outperform various models, including gradient boosting machines and PCA-based regression models, in terms of prediction accuracy and computation time. Additionally, Weerasinghe and Wijegunasekara (2016) performed a comparative analysis of NNs, decision trees, and multinomial logistic regression models and concluded that NNs demonstrated the most efficient predictive performance. Hence, there are many prominent applications of NNs on loss models. Smith et al. (2000) utilized decision trees and NNs to analyze the likelihood of policyholders submitting claims, discussing the consequential impact of prediction accuracy on insurance companies. Yunos et al. (2016) introduced DLNNs for two distinct claims components: claim count and claim severity (claim cost per claim). Employing nine different network structures, they underscored the significant impact of network architecture on prediction accuracy.

In recent years, the basic feedforward or MLP NNs have evolved into advanced architectures that enhance flexibility and adaptability. Notable extensions include DLNNs with multiple hidden layers between the input and output layers (Kim et al., 2022). Recurrent NNs (RNNs) are designed for sequential trip data by incorporating feedback connections, convolutional NNs (CNNs) are tailored for grid-like data such as images, and long short-term memory (LSTM) NNs focus on capturing and learning long-term dependencies in sequential trip data. Dong et al. (2016) employed 1-dimensional CNNs and RNNs to analyze a large GPS trip dataset, showcasing the superior performance of deep learning architectures over traditional machine learning methods. Simoncini et al. (2018) investigated RNNs for vehicle classification using GPS trip data. Additionally, DLNNs emerge as favorable choices for predictive claim models, offering a well-suited architecture for handling the nuanced relationships between DVs and claims/exposure data, outperforming other architectures like CNN and RNN (Zhang et al., 1998). The inherent complexity and dynamic nature of driving behavior data make DLNNs particularly applicable to capturing and learning

intricate patterns effectively, ensuring accurate and efficient claim prediction for the insurance domain. DLNNs, explored by Guo et al. (2018), prove effective in capturing hidden driving patterns across diverse drivers using telematics data.

However, the shortcomings of the common mean square error (MSE) loss function in NNs become apparent when dealing with skewed claim data exhibiting zero inflation. Essentially, NNs with MSE loss functions assume a normal distribution for symmetric data. This assumption fails to capture the right skewness of the claim data even though non-linear activation functions such as Sigmoid or equivalently logistic link function can be adopted. More importantly, MSE loss function does not include data distribution information and hence, it fails to provide uncertainties of the target variable estimates. Uncertainty quantification has received growing interest in NNs (Amini et al., 2020). Fallah et al. (2009) proposed Poisson NNs using the negative loglikelihood (NLL) loss function for the Poisson model and compared the Poisson NNs to Poisson regressions in GLMs for claim counts. Notably, Wüthrich and Merz (2019) innovatively blended the simple GLM with the NN architecture, termed it the Combined Actuarial Neural Net (CANN) approach. This NN is a hybrid GLM and NN model that can be used to test model improvement from either model. Sakthivel and Rajitha (2017) compared zero-inflated hurdle models with NNs in claim count modeling. Building upon the Poisson mixture (PM) lasso regression models in Chapter 3, this chapter extends the Poisson NNs to two-group PM DLNNs to classify safe and risky drivers. However, including the prior weight parameter π for the mixture components in (3.3) poses substantial challenges to model implementation. Similar to lasso regression models in Chapter 3, this weight parameter π is a non-regression parameter, meaning that it is kept constant across drivers. Hence, its estimation will be separate from the NN parameters using, for example, stochastic gradient descent (SGD).

While NNs can be very flexible in capturing the complex non-linear effects of DVs on claim counts, they are also subject to the challenge of overfitting, a problem prevalent in complex NN architectures that perform well on training data but may falter on other data sets (Goodfellow et al. 2016; Ying 2019). Regularization in NN serves

as a crucial set of techniques aimed at mitigating overfitting by creating a robust NN architecture with good generalization when dealing with intricate data (Mishra et al. 2019; Li et al. 2020). Several strategies, including batch normalization (Ioffe and Szegedy 2015; Garbin et al. 2020), dropout (Srivastava et al. 2014; Garbin et al. 2020), and early stopping, are employed to address overfitting concerns. However, it is essential to note that both batch normalization and dropout necessitate the tuning of hyperparameters to effectively align with the network architecture and train models with varying combinations of hyperparameters, impacting their behavior and increasing training time (Srivastava et al. 2014; Garbin et al. 2020). Batch normalization, in particular, significantly reduces training time by normalizing the input of neurons in each layer of the NN. It proves effective with high learning rates and reduces the number of training steps required for convergence (Ioffe and Szegedy 2015; Garbin et al. 2020). These regularization techniques play a pivotal role in enhancing the generalization capacity of NNs, ensuring their adaptability and resilience when handling complex data sets.

Once the regularized PM DLNNs are set up, the insights derived from the hidden patterns uncovered by the networks present the potential to create more precise risk profiles, offering a valuable tool for insurers to develop personalized pricing and underwriting strategies in alignment with the UBI experience-rating premium pricing method proposed in Chapter 3. With their inherent capabilities, DLNNs are pivotal in improving the accuracy, efficiency, and automation of claim prediction. By leveraging the power of NNs, insurers can make more informed decisions, effectively minimize risks, detect and mitigate instances of fraud, and ultimately enhance the overall customer experience within the insurance industry.

This chapter contributes on multiple fronts as outlined below:

1. **Pioneering PM DLNNs**: This chapter proposes the innovative PM DLNNs to learn hidden driving patterns for claim prediction and driver classification. Due to the complexity of PM DLNNs, architecture search is crucial and is guided by practical considerations, with a nuanced interplay of factors or fine-tuning parameters, including epochs, batch size, learning rates, and optimizers.

DLNNs with five hidden layers are meticulously chosen with each layer playing a pivotal role in discerning intricate patterns within DVs. Apart from architecture search, other techniques such as batch normalization, early stop, and dropout are also adopted to mitigate underfitting and overfitting as well as enable speedy convergence of parameters.

2. **Adopting NLL loss function with constraints**: A notable challenge to implement the designated PM DLNNs by defining the NLL loss function is the estimation of non-regression prior weight parameter π. Moreover, incorporating constraints in defining the mean parameters of the two Poisson distributions in the NLL loss function is important to distinguish and separate two distinct mixture groups of safe and risky drivers.

3. **Iteratively estimating the prior weight parameter**: This chapter proposes a methodology that involves iterative estimation of the DLNN parameters denoted by $\boldsymbol{\theta}$ condition on π, followed by re-estimation of π condition on the DLNN parameters $\boldsymbol{\theta}$ until convergence or early stop criteria are attained. The whole PM DLNNs are implemented via `Keras` and `TensorFlow` in the `Python` environment. Model performance is visualized by the scatter plot of predicted annual claims against the observed as well as numerical measures such as NLL and MSE that measure model fit and prediction accuracy respectively.

4.2 Deep learning neural networks

This section reviews DLNNs with a focus on PM models.

4.2.1 Architecture of DLNNs

The architecture of DLNNs is inspired by neuroplasticity models of the human brain. In the simple form of MLP (also called feed-forward) without loop connections, NNs form a finite acyclic graph, which is a finite directed graph with no directed cycles

(Goodfellow et al., 2016). Sze et al. (2017) believed that the brain is learned through changes to the weights associated with the synapses, and different weights respond to different inputs. In a close-up, the structure of the NN resembles a layered cobweb with a series of neurons that take one or more inputs on the left that come from raw data, and the values are fed forward from the input layer to the hidden and output layers. Each layer contains one or more neurons, also called perceptrons. The operation of a neuron involves two steps. In each layer, the input is first linearly combined with layer weights. Then, a non-linear activation function is applied to the result. These hierarchical abstract layers of neurons as latent variables perform pattern matching and forecasting.

DLNNs (LeCun et al., 2015) have multiple processing layers to hierarchically represent the data through the use of interconnected layers of artificial neurons. Through this high-level abstraction of data, they automatically learn to extract salient features of the input data. The common DL models are MLP, recurrent neural network (RNN), long short-term memory (LSTM), convolutional neural networks (CNN), restricted Boltzmann machines (RBM), and autoencoder (AE). Among these models, RNN and LSTM models are particularly suitable for financial time series returns and price volatility forecasting; CNN and AE models are widely used in data compression, data denoising, and image recognition for supervised and unsupervised learning, respectively; and RBM models are applied to natural language processing. As the same cross-section drivers by DVs data described in Section 3.2 is applied in this Chapter 4, DLNNs are the most suitable models.

4.2.2 Basic structure

To predict claims and classify drivers into safe and risky groups, DLNNs are derived within the PM model framework based on the model structure in Section 3.3.1 and the telematics data in Section 3.2.

Let y_i denote the observed claim count of driver i, $y_i, i = 1 \ldots, N$, and n_i be the policy exposure for driver i. There are $N = 14157$ drivers. The input features

or covariates are $J_0' = 45$ informative DVs, and they are presented as $\boldsymbol{X}_{(1:N)\times(1:J_0')} = (\boldsymbol{x}_{1\cdot}, \ldots, \boldsymbol{x}_{N\cdot})^\top \in \mathbb{R}^N \times \mathbb{R}^{J_0'}$. The target variable is the claim count $\boldsymbol{y}_{1:N}$ with exposure $\boldsymbol{n}_{1:N}$ and they are used to train the DLNN to predict claim counts $\boldsymbol{A}_{1:N,1:m_f} = (\boldsymbol{a}_{1\cdot}, \ldots, \boldsymbol{a}_{N\cdot})^\top \in \mathbb{R}^N \times \mathbb{R}^G$ as output, where $\boldsymbol{a}_i = (a_{i1}, \ldots, a_{iG})^\top$ represent the annual claims in G driver groups.

For the DLNN structure, assume that there are $\mathfrak{L}$ hidden layers and each hidden layer contains m_l neurons where $\boldsymbol{m} = (m_1, \ldots, m_l)$ and m_0, m_f are the size of input features (layer 0) and output targets (layer $\mathfrak{L}+1$) respectively. To find the predicted annual claim vector $\boldsymbol{A}$, DLNNs use forward propagation by departing from the left input layer to the hidden layers and arriving at the output layer using the MLP approach. The algorithm needs good weights linking the layers to get accurate predictions. For layer $l, l = 1, \ldots, \mathfrak{L}$, let the weights be $\mathrm{w}_{jk}^{(l)}, j = 1, \ldots, m_{l-1}, k = 1, \ldots, m_l$, the biases be $b_k^{(l)}$ and the k-th output neuron in layer l be $a_{ik}^{(l)}$. These outputs model the annual claims $a_i = y_i/n_i$ for driver i.

In each layer l, the output $a_{ik}^{(l)}$ for driver i and neuron k is obtained by first weighting $\boldsymbol{a}_i^{(l-1)} = (a_{i1}^{(l-1)}, \ldots, a_{im_{l-1}}^{(l-1)})^\top \in \mathbb{R}^{m_{l-1}}$ with the weight vector $\mathbf{w}_k^{(l)} = (\mathrm{w}_{1k}^{(l)}, \ldots, \mathrm{w}_{m_{l-1}k}^{(l)})^\top \in \mathbb{R}^{m_{l-1}}$. Then summing and bias-adjusting is applied as follows

$$z_{ik}^{(l)} = \sum_{j=1}^{m_{l-1}} \mathrm{w}_{jk}^{(l)} a_{ij}^{(l-1)} + b_k^{(l)} = {\mathbf{w}_k^{(l)}}^\top \boldsymbol{a}_i^{(l-1)} + b_k^{(l)} = \mathsf{f}(\boldsymbol{a}_i^{(l-1)} | \mathbf{w}_k^{(l)}, b_k^{(l)}),\ k = 1, \ldots, m_l \quad (4.1)$$

where $\mathsf{f}(\cdot|\mathbf{w}_k^{(l)}, b_k^{(l)})$ is the linear function and $a_{ik}^{(0)} = x_{ik}$ for the input layer. Lastly, $z_{ik}^{(l)}$ is non-linearly transformed as below:

$$f_l(z_{ik}^{(l)}) = a_{ik}^{(l)}$$

where $f_l(\cdot)$ is a non-linear function called *activation function* to provide a reasonable output range, for example, non-negative output. Choices of $f_l(\cdot)$ are reviewed in Section 4.2.3. In summary,

$$a_{ik}^{(l)} = f_l\left({\mathbf{w}_k^{(l)}}^\top \boldsymbol{a}_i^{(l-1)} + b_k^{(l)}\right) = f_l\left(\sum_{j=1}^{m_{l-1}} \mathrm{w}_{jk}^{(l)} a_{ij}^{(l-1)} + b_k^{(l)}\right) = f_l \circ \mathsf{f}(\boldsymbol{a}_i^{(l-1)} | \mathbf{w}_k^{(l)}, b_k^{(l)}) \quad (4.2)$$

where $f_l \circ \mathsf{f}$ is the composite mapping in each layer l and

$$\boldsymbol{a}_i^{(l)} = f_l(\boldsymbol{W}^{(l)\top}\boldsymbol{a}_i^{(l-1)} + \boldsymbol{b}^{(l)}) = f_l \circ \mathsf{f}(\boldsymbol{a}^{(l-1)}|\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}) := f_l^{\boldsymbol{W}^{(l)},\boldsymbol{b}^{(l)}}(\boldsymbol{a}^{(l-1)}) \tag{4.3}$$

where $\boldsymbol{a}_i^{(l)} = (a_{i1}^{(l)}, \ldots, a_{im_l}^{(l)})^\top$ is a $m_l \times 1$ dimensional neuron vector. Finally, the output $\boldsymbol{a}_i^{(l)}$ is taken as the input for the next layer $l+1$ and so on.

In the output layer $\mathfrak{L}+1$ with m_f neurons, the output $\boldsymbol{a}_i^{(\mathfrak{L}+1)} = (a_{i1}^{(\mathfrak{L}+1)} \ldots, a_{im_f}^{(\mathfrak{L}+1)})$ predicts the annual claim for driver i in group g, $g = 1, \ldots, m_f$. Writing $\mu_{a,ig} = a_{ig}^{(\mathfrak{L}+1)}$, the predict claims $\hat{y}_{ig} = \mu_{ig} = n_i\mu_{ig}$ for driver i in group g.

Then the vector of all parameters is $\boldsymbol{\theta} = (\boldsymbol{W}^{(1)}, \boldsymbol{b}^{(1)}, \ldots, \boldsymbol{W}^{(\mathfrak{L}+1)}, \boldsymbol{b}^{(\mathfrak{L}+1)})$ where $\boldsymbol{b}^{(l)} = (b_1^{(l)}, \ldots, b_{m_l}^{(l)})^\top$ is the $m_l \times 1$ dimensional bias vector, and $\mathbf{W}^{(l)} = (\mathrm{w}_{jk}^{(l)})$ is the $m_{l-1} \times m_l$ dimensional weight matrices. They are estimated through backward propagation in Section 4.2.5 by optimizing the loss function as described in Sections 4.2.4 and 4.2.6. The number M of all parameters in $\boldsymbol{\theta}$ is calculated as:

$$M = \sum_{l=1}^{\mathfrak{L}+1}[m_l \times (m_{l-1} + 1)]. \tag{4.4}$$

Similar to regression models, DLNN defines a function f_{DLNN} as a mapping from input data features to the desired annual claims by group, $\mathbf{X} \to \mathbf{A}$ via all hidden layers with activation functions $f_l^{\boldsymbol{W}^{(l)},\boldsymbol{b}^{(l)}}$ in (4.3). The deep predictor becomes a composite mapping :

$$\boldsymbol{a}_i^{(\mathfrak{L}+1)} = f(\boldsymbol{x}_i|\boldsymbol{\theta}) = f_{\mathfrak{L}+1}^{\boldsymbol{W}^{(\mathfrak{L}+1)},\boldsymbol{b}^{(\mathfrak{L}+1)}} \circ \cdots \circ f_1^{\boldsymbol{W}^{(1)},\boldsymbol{b}^{(1)}}(\boldsymbol{a}^{(0)}) \in \mathbb{R}^{+,m_f} \tag{4.5}$$

where $\boldsymbol{a}^{(0)} = \boldsymbol{x}_i$. Here, the mapping $f(\boldsymbol{X})$ can be modeled by a superposition.

In the two-group PM DLNN, $m_f = 2$ signifies the target of the predicted annual claim for each mixture component $g = 1, 2$. The predicted claims in each group are $\mu_{ig}, = n_i a_{ig}^{(\mathfrak{L}+1)}$, and $\hat{y}_i$ is given by (3.4). Section 4.2.8 provides details of hyperparameter tuning, including the number of hidden layers $\mathfrak{L}$ and the number of neurons $\boldsymbol{m}$ in each layer. Results find $\mathfrak{L} = 5$ hidden layers and $\boldsymbol{m} = (45, 30, 15, 10, 5)$ neurons

in the hidden layers with $m_0 = 45$ and $m_f = 2$ neurons in the input and output layers, respectively provide the good architecture for predicting annual claims of each mixture component. The `Keras`'s `Sequential()` model in `Python` is used to build the models. Figure 4.1 presents a DLNN for the PM model, and Table 4.1 illustrates the calculation of the number of parameters M using (4.4). The choice of architecture $\mathfrak{L} = 5$ and $\boldsymbol{m}$ is explained in Sections 4.2.8 and 4.2.9. The dropout layer for regularization and batch layer are explained in Section 4.2.7. The dropout layer does not involve new parameters.

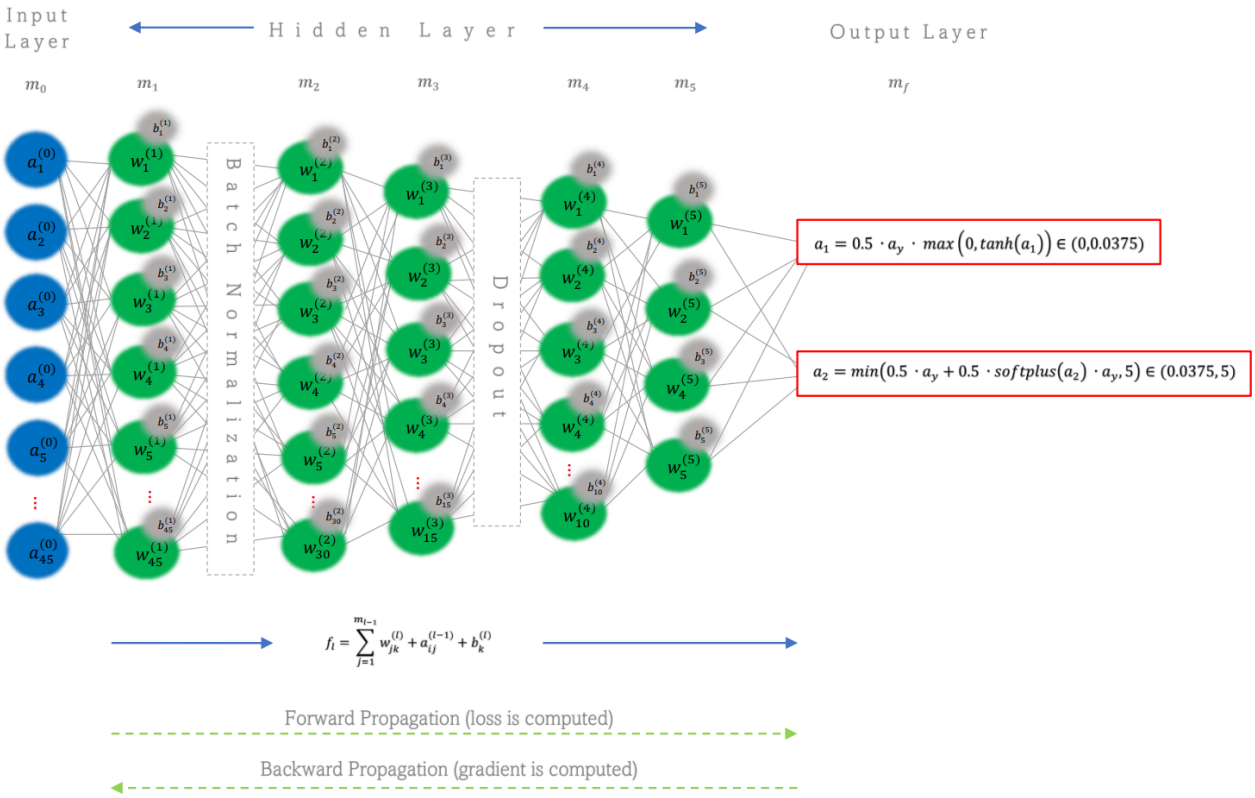
Table 4.1 – Computation of the number of parameters M for PM-DLNN with $m_0 = 45$ features, $\mathfrak{L} = 5$ hidden layers, $\boldsymbol{m} = (45, 30, 15, 10, 5)$ neurons in the hidden layers, and $m_f = 2$ neuron in the output layer.

Layer l	**No. of neuron** m_l	**No. of parameters**	**Total**
Input	$m_0 = 45$	0	0
1	$m_1 = m_0 = 45$	$45 \times 45 + 45$	2070
`BatchN`	45	4×45	180
2	$m_2 = 2m_1/3 = 30$	$45 \times 30 + 30$	1380
3	$m_3 = m_2/2 = 15$	$30 \times 15 + 15$	465
`DropOut` (0.20)	0	0	0
4	$m_4 = 2m_3/3 = 10$	$15 \times 10 + 10$	160
5	$m_5 = m_4/2 = 5$	$10 \times 5 + 5$	55
Output	$m_f = 2$	$5 \times 2 + 2$	12
Overall total M			4322

4.2.3 Activation function

The activation function can be specified as linear or non-linear in both hidden and output layers, depending on the type of target variables. It captures nonlinearity in the hidden layers and transforms the output to a desirable range such as positive for the predicted annual claims in the output layer. When choosing this function, it prefers to be non-linear to model non-linear patterns, differentiable for gradient calculation in backward propagation, and smooth to reduce the oscillations and noise during the training.

Figure 4.1 – Poisson mixture deep learning neural network (PM-DLNN) with $m_0 = 45$ features, $\mathfrak{L} = 5$ hidden layers, $\boldsymbol{m} = (45, 30, 15, 10, 5)$ neurons in the hidden layers, and $m_f = 2$ neuron in the output layer.



The Universal Approximation Theorem stipulates that any activation function $f(z)$ (suppressing the subscript l for layer l) has the following properties (Liew et al., 2016):

1. The output $f(z)$ is non-constant for all ranges of inputs z;
2. The output $f(z)$ is bounded within a range;
3. The function f is continuous over all points c in its domain, that is, $\lim_{z \to c} f(z) = f(c)$;
4. The function f is monotonically increasing, that is $f(z_2) \geq f(z_1)$ if $z_2 > z_1$;
5. The function f is differentiable everywhere, where $\lim_{c \to 0} \dfrac{f(z+c) - f(z)}{c} = f'(z)$ exists for all z.

The fifth property is required for backward propagation learning algorithms in Section 4.2.5. The choices of activation functions applied in this chapter include:

1. `Linear` is the default in any network and is defined as

$$f_{\texttt{linear}}(z) = z \in (-\infty, \infty).$$

2. Rectified linear unit (`ReLU`) is defined as

$$f_{\texttt{ReLU}}(z) = \max(0, z) \in (0, \infty).$$

   The `ReLU` function is faster to train because it has a fixed derivative (slope). However, when $z < 0$, $f_{\texttt{ReLU}}(z) = 0$ always, making the network unable to update the weights, a phenomenon called "neuron death" (Sharma et al., 2017). Moreover, the `ReLU` output is biased.

3. Exponential linear unit (`ELU`) solves the problems in dead neurons and is defined as

$$f_{\texttt{ELU}}(z) = \begin{cases} z, & \text{if } z > 0 \\ \xi(e^z - 1), & \text{if } z \leq 0 \end{cases}$$

   where ξ is a hyperparameter. When the input is negative, the `ELU` function is characterized by a negative output range $(-1, 0)$. This effectively avoids the problem of neuronal death and is employed in the last hidden layer.

4. Leaky rectified linear unit (`LeakyReLU`) is defined as

$$f_{\texttt{LeakyReLU}}(z) = \begin{cases} z, & \text{if } z \geq 0 \\ \xi z, & \text{if } z < 0. \end{cases}$$

   The hyper-parameter ξ is set as 0.01 (Sharma et al., 2017). Thus, it also solves the "dying `ReLU`" problem. Compared with `ReLU`, this allows a small amount of information to flow when $z < 0$.

5. Hyperbolic tangent (`tanh`) is defined as

$$f_{\texttt{tanh}}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = 2f_{\texttt{sigmoid}}(2z) - 1 \in (-1, 1) \text{ where } f_{\texttt{sigmoid}}(z) = \frac{1}{1 + e^{-z}}. \tag{4.6}$$

   Sharma et al. (2017) commented that `tanh` is preferred over the `sigmoid` function because the gradients are not restricted to vary in a certain direction and are zero-centered.

6. `Softplus` is a smoothed version of `ReLU` which avoids the zero gradients when $x \leq 0$ and is defined as

$$f_{\texttt{softplus}}(z) = \log(1 + e^z). \tag{4.7}$$

   The output of `Softplus` grows continuously, but the rate of increase is much slower when z is more negative.

4.2.4 Loss function

Parameters $\boldsymbol{\theta}$ are estimated to optimize some objective functions $\mathcal{J}(\boldsymbol{\theta})$, that is,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\theta} \mathcal{J}(\boldsymbol{\theta})$$

for training data. The MSE is a common metric to evaluate the performance of a model. When applying to telematics data, it measures the average of the squared errors

$$\mathcal{J}_{\text{MSE}}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \Big(a_i - \hat{a}_i(\boldsymbol{\theta})\Big)^2, \tag{4.8}$$

which are the differences between the observed $a_i = y_i/n_i$ and predicted annual claims $\hat{a}_i(\boldsymbol{\theta})$ in the output layer, and N is the data size. For the PM model, MSE as a prediction accuracy metric can be defined as

$$\mathcal{J}_{\text{PMMSE1}}(\boldsymbol{\theta}, \pi) = \frac{1}{N} \sum_{i=1}^{N} \Big(a_i - \hat{a}_i(\boldsymbol{\theta})\Big)^2, \text{ where } \hat{a}_i(\boldsymbol{\theta}) = \mathfrak{z}_{i1}\hat{a}_{i1}(\boldsymbol{\theta}) + (1 - \mathfrak{z}_{i1})\hat{a}_{i2}(\boldsymbol{\theta}) \tag{4.9}$$

where the marginal posterior predicted annual claims $\hat{a}_i(\boldsymbol{\theta})$ is a weighted average of the two predicted annual claims from each group using posterior probability, that is,

$$\hat{a}_i(\boldsymbol{\theta}) = \mathfrak{z}_{i1}\hat{a}_{i1}(\boldsymbol{\theta}) + (1 - \mathfrak{z}_{i1})\hat{a}_{i2}(\boldsymbol{\theta}), \tag{4.10}$$

$\hat{a}_{ig}(\boldsymbol{\theta}) = a_{ig}^{(\mathfrak{L}+1)}$ are the two output neurons for the predicted annual claims of each groups, and the posterior probability $\mathfrak{z}_{i1}$ is defined in (3.6). Instead of posterior probability $\mathfrak{z}_{i1}$, if the prior probability π defined in (3.3) is used, MSE can be defined as

$$\mathcal{J}_{\text{PMMSE2}}(\boldsymbol{\theta}, \pi) = \frac{1}{N}\sum_{i=1}^{N}\Big(a_i - \hat{a}_i(\boldsymbol{\theta})\Big)^2, \text{ where } \hat{y}_i(\boldsymbol{\theta}) = \pi\hat{a}_{i1}(\boldsymbol{\theta}) + (1-\pi)\hat{a}_{i2}(\boldsymbol{\theta}). \tag{4.11}$$

If the comparison is made to the observed and predicted claims, MSE can be defined as

$$\mathcal{J}_{\text{PMMSE3}}(\boldsymbol{\theta}, \pi) = \frac{1}{N}\sum_{i=1}^{N}\Big(y_i - \hat{y}_i(\boldsymbol{\theta})\Big)^2, \text{ where } \hat{y}_i(\boldsymbol{\theta}) = n_i[\pi\hat{a}_{i1}(\boldsymbol{\theta}) + (1-\pi)\hat{a}_{i2}(\boldsymbol{\theta})]. \tag{4.12}$$

Lastly, an alternative MSE can be defined as

$$\mathcal{J}_{\text{PMMSE4}}(\boldsymbol{\theta}, \pi) = \frac{1}{N}\sum_{i=1}^{N}\Big[\pi\Big(y_i - \hat{y}_{i1}(\boldsymbol{\theta})\Big)^2 + (1-\pi)\Big(y_i - \hat{y}_{i2}(\boldsymbol{\theta})\Big)^2\Big]$$

by weighting the squared errors, but this definition is less popular. When using MSE as a performance metric, PM MSE using $\mathcal{J}_{\text{PMMSE}}(\boldsymbol{\theta}, \pi)$ is used.

However, when MSE is used as a loss function, it essentially assumes a normal distribution for symmetric data since the negative loglikelihood (NLL) function $-\ell$ for normal distribution includes MSE term apart from the terms of normalizing constant and σ^2 which is also assumed constant. Hence, the shortcomings of the MSE loss function are obvious when dealing with the skewed annual claim data exhibiting zero inflation in the right plot of Figure 3.1. More importantly, the use of MSE loss function, as compared to NLL loss function, does not provide information regarding the shape of data distribution and hence the variability of the target estimates using MSE

loss function. Lastly, it should be noted that non-linear activation functions do not deal with data distribution.

To capture the asymmetry of the claim counts, Poisson and PM are proposed in Section 3.3.1. Similar to the MSE case representing the normal distribution NLL, the Poisson and PM NLL can be adopted for the loss function. The NLL loss function for Poisson and PM models are defined, respectively, as:

$$\mathcal{J}_{\text{PNLL}}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(n_i\hat{a}_i(\boldsymbol{\theta})\right) - n_i\hat{a}_i(\boldsymbol{\theta}) - \log(y_i!)\right]$$

and

$$\mathcal{J}_{\text{PMNLL}}(\boldsymbol{\theta},\pi) = -\frac{1}{N}\sum_{i=1}^{N}\log f_{\text{PM}}(y_i;\pi,\mu_{i1},\mu_{i2}) = -\frac{1}{N}\sum_{i=1}^{N}\log\left[\pi\frac{\mu_{i1}^{y_i}\mathrm{e}^{-\mu_{i1}}}{y_i!} + (1-\pi)\frac{\mu_{i2}^{y_i}\mathrm{e}^{-\mu_{i2}}}{y_i!}\right] \tag{4.13}$$

where $f_{\text{PM}}(y_i;\pi,\mu_{i1},\mu_{i2})$ is given in (3.3),

$$\mu_{ig} = n_i\hat{a}_{ig}(\boldsymbol{\theta}) = n_i a_{ig}^{\mathfrak{L}+1} \tag{4.14}$$

is the i-th entry in output vector for the g-th group. Again, the marginal predicted claims $\hat{y}_i$ is given in (3.4) where $\mathfrak{z}_{i1}$ by (3.6) in Section 3.3.1.

Lastly, instead of updating the model's weights based $\mathcal{J}(\boldsymbol{\theta})$ computed from the entire training dataset, training is often performed on smaller subsets of the training data, known as *batches* or *mini-batches*. The optimization algorithm using gradient descent and applying to batches is called batch gradient descent.

The advantages of using mini-batches include computational efficiency, taking advantage of parallelization, especially when working with large datasets, regularization effects by introducing randomness of mini-batch to prevent overfitting, and memory efficiency to reduce memory load. The process of training the DLNN based on $\mathcal{J}(\boldsymbol{\theta})$ using mini-batches with batch size $\mathcal{B}$ is commonly referred to as *mini-batch training*. Hence, the data size N may denote training data size $N_{\text{T}} = \mathsf{p}N$ where p is the proportion of data assigned to training, validation data size $N_{\text{V}} = (1-\mathsf{p})N$ or the

mini-batch size $\mathcal{B}$. The `Python` codes to calculate NLL and MSE losses are given in code lists 4.1 in Appendix 4.6.1 and 4.2 in Appendix 4.6.2.

4.2.5 Forward and backward propagation

Given the set of weights and bias, passing forward through the NN means finding all the neuron outputs in (4.2) and (4.3). Subsequently, traversing backward through the NN is to compute the gradients of a general loss function $\mathcal{J}(\boldsymbol{\theta})$ with respect to the network's parameters. These gradients are then used to update the parameters using various gradient descent methods. Given by the chain rule, the derivative of $\mathcal{J}(\boldsymbol{\theta})$ in terms of the input $\boldsymbol{x}$ evaluated at the value of the network at each neuron is:

$$\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \boldsymbol{a}^{\mathfrak{L}}} \cdot \frac{\partial \boldsymbol{a}^{\mathfrak{L}}}{\partial \boldsymbol{z}^{\mathfrak{L}}} \cdot \frac{\partial \boldsymbol{z}^{L}}{\partial \boldsymbol{a}^{\mathfrak{L}-1}} \cdot \frac{\partial \boldsymbol{a}^{\mathfrak{L}-1}}{\partial \boldsymbol{z}^{\mathfrak{L}-1}} \cdot \frac{\partial \boldsymbol{z}^{\mathfrak{L}-1}}{\partial \boldsymbol{a}^{L-2}} \cdot \ldots \cdot \frac{\partial \boldsymbol{a}^{1}}{\partial \boldsymbol{z}^{1}} \cdot \frac{\partial \boldsymbol{z}^{1}}{\partial \boldsymbol{x}},$$

where $\dfrac{\partial \boldsymbol{a}^{\mathfrak{L}}}{\partial \boldsymbol{z}^{\mathfrak{L}}}$ is a diagonal matrix and each term is a total derivative. These terms are the derivative of the loss function, the derivatives of the activation functions, and the weight matrices. Following from (4.5), it can be written as

$$(\boldsymbol{W}^{(1)})^{\top} \cdot (f_1)' \circ \ldots \circ (\boldsymbol{W}^{(\mathfrak{L}-1)})^{\top} \cdot (f_{\mathfrak{L}-1})' \circ (\boldsymbol{W}^{(\mathfrak{L})})^{\top} \cdot (f_{\mathfrak{L}})' \circ \frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \boldsymbol{a}^{(\mathfrak{L})}}$$

since the gradient is the transpose of the derivative of the output in terms of the input, so the matrices are transposed, and the order of multiplication is reversed. In layer l, the gradients are

$$\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \mathrm{w}_{jk}^{(l-1)}} = \frac{\partial z_k^{(l)}}{\partial \mathrm{w}_{jk}^{(l)}} \frac{\partial a_k^{(l)}}{\partial z_k^{(l)}} \frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial a_k^{(l)}}$$

where $z_k^{(l)}$ and $a_k^{(l)}$ are defined in (4.1) and (4.2) suppressing the subscript i for driver i and

$$\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial a_k^{(l-1)}} = \sum_{j=1}^{m_l} \frac{\partial z_j^{(l)}}{\partial a_k^{(l-1)}} \frac{\partial a_k^{(l)}}{\partial z_k^{(l)}} \frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial a_k^{(l)}}.$$

Then, a new version of weights is iteratively updated using the first-order gradient descent method that minimizes $\mathcal{J}(\boldsymbol{\theta})$. One *epoch* is an iterative process for an entire data set. It consists of $\mathfrak{I} \simeq N_{\mathrm{T}}/\mathcal{B}$ (called iteration) mini-batches of size $\mathcal{B}$ which depicts the number of samples in training data propagating through the NN before updating the model parameters in one iteration, and N_{T} is the training data size. This epoch is repeated until the model performance is optimized.

4.2.6 Model optimization

Some commonly used optimizers for $\mathcal{J}(\boldsymbol{\theta})$, including stochastic gradient descent, adaptive gradient algorithm, adaptive learning rate method, and adaptive moment estimation, are introduced. All these methods use mini-batch training.

First order stochastic gradient descent (`SGD`)

`SGD` is also known as the first-order Newton-Raphson method by replacing the second-order derivatives of the loss function with a constant learning rate (Drori, 2022). The algorithm is formulated as follows:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \zeta \left. \frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} \tag{4.15}$$

where $\boldsymbol{\theta}_t$ are the model parameters in the t-th iteration, $\mathcal{J}(\boldsymbol{\theta})$ is the loss function, ζ is the learning rate, and $\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$ is the gradient of the loss function $\mathcal{J}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}_t$. With mini-batch training, this method is called mini-batch `SGD` (MSGD).

Adaptive gradient algorithm (`AdaGrad`)

`AdaGrad` is an improved version of the `SGD` algorithm that increases ζ for sparser parameters and decreases ζ for less sparse ones. This strategy often improves convergence performance where data is sparse and sparse parameters are more informative.

This optimizer adjusts ζ for any parameter $\theta_{jt} \in \boldsymbol{\theta}_t = (\mathrm{w}_{jk,t}^{(l)}, b_{k,t}^{(l)})$ at iteration t depending on the squared sum of past partial derivatives. The update rule is given by:

$$\theta_{j,t+1} = \theta_{i,t} - \left(\frac{\zeta}{\sqrt{v_{jt}} + \epsilon} \right) \frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \theta_j} \bigg|_{\theta_j = \theta_{jt}} \tag{4.16}$$

where

$$v_{j,t+1} = v_{j,t} - \left(\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \theta_j} \bigg|_{\theta_j = \theta_{jt}} \right)^2, \tag{4.17}$$

$\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \theta_j}|_{\theta_j = \theta_{jt}}$ is the derivative of $\mathcal{J}(\boldsymbol{\theta})$ with respect to θ_j evaluated at $\theta_j = \theta_{jt}$ at iteration t and ϵ is a smoothing term to avoid zero value. However, the main flaw of `AdaGrad` is its accumulation of the squared gradients in the denominator, which causes the learning rate to shrink so that local estimates using recent gradients eventually become infinitesimally small (Zeiler, 2012).

Adaptive learning rate method (`AdaDelta`)

To resolve the above issues, `AdaDelta` reduces its aggressive, monotonically decreasing learning rate by using a decay rate:

$$v_{t+1} = \varphi v_t - (1 - \varphi) \left(\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \theta_j} \bigg|_{\theta_j = \theta_{jt}} \right)^2, \tag{4.18}$$

where φ is a decay rate, and the updating formula is given by (4.16). This way, it restricts the window of accumulated past gradients to a fixed size. It avoids storing previously squared gradients by recursively defining the sum of gradients as a decaying average of all past squared gradients so that it depends only on the previous standard and current gradient. See Qu et al. (2019) for an application of `AdaDelta`.

Adaptive moment estimation (`Adam`)

The `Adam` algorithm (Zhang, 2018) is another advanced algorithm defined as:

$$\theta_{j,t+1} = \theta_{jt} - \zeta \frac{\hat{\varsigma}_t}{\sqrt{\hat{v}_{jt}} + \epsilon} \tag{4.19}$$

where the first-order ς_t and second-order v_t moment estimates of the gradient $\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \theta_j}$ are given by

$$\varsigma_{jt} = \varphi_1 \varsigma_{j,t-1} + (1-\varphi_1) \frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \theta_j}\bigg|_{\theta_j=\theta_{jt}} \text{ and } v_{jt} = \varphi_2 v_{j,t-1} + (1-\varphi_2) \left(\frac{\partial \mathcal{J}(\boldsymbol{\theta})}{\partial \theta_j}\right)^2\bigg|_{\theta_j=\theta_{jt}} \tag{4.20}$$

respectively, the average value of the first moment and second moment (variance) of the gradient are

$$\hat{\varsigma}_t = \frac{\varsigma_t}{1-\varkappa_1^t}, \quad \text{and} \quad \hat{v}_t = \frac{v_t}{1-\varkappa_2^t}, \tag{4.21}$$

and φ_1 and φ_2 are the decay rates and $\varkappa_1$ and $\varkappa_2$ are constants proposed to take values 0.9 and 0.999 respectively.

4.2.7 Regularization

Regularization is important in NNs because it helps prevent overfitting, a phenomenon where a model learns the training data too well, capturing noise, specific patterns, and overly complex representations that do not generalize well to new, unseen data. Normalization, batch normalization (`BatchN`), dropout (`DropOut`), L1 & L2 regularization, and callback functions or early stopping are introduced as powerful regularization techniques to mitigate the challenges of both under and over-fitting in NNs and to ensure stability and efficiency of the learning process.

Normalization

Normalization can be applied to the entire data set, batches, etc. When applied to the whole data set, all feature vectors $\boldsymbol{x}$ are transformed into the same scale. There are two types:

$$\begin{aligned} \text{Min-max normalization: } \boldsymbol{x}_{\text{scaled}} &= \frac{\boldsymbol{x} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})} \in (0,1) \\ \text{Standardization: } \boldsymbol{x}_{\text{stand}} &= \frac{\boldsymbol{x} - \text{mean}(\boldsymbol{x})}{\text{sd}(\boldsymbol{x})},\ \text{E}(\boldsymbol{x}_{\text{stand}}) = 0,\ \text{Var}(\boldsymbol{x}_{\text{stand}}) = 1 \end{aligned}$$

In this chapter, standardization is applied to normalize the 45 features before fitting to DNLLs so that each feature will have $\mu = 0$ and $\sigma = 1$. `Python` code `StandardScaler()` is used to perform the standardization.

Batch normalization

`BatchN` is an important technique in mini-batch training. By adding noise in forming random batches during training, it allows the model to train more consistently and converge faster for each input variable across layers over a mini-batch (Santurkar et al., 2018; Ioffe and Szegedy, 2015). It also helps to address the vanishing and exploding gradient problems, making it easier for gradients to flow through the network, reducing sensitivity to initialization, and enabling higher learning rates.

`BatchN` is suggested to apply to layer 1 so that for any batch with size $\mathcal{B}$, it transforms each input $a_{ik}^{(1)}, k = 1, \ldots, m_1, i = 1, \ldots, \mathcal{B}$ for the observations within the batch by standardization and transformation using four parameters, namely the shifting parameters $\mathfrak{a}_k$, the scaling parameter $\mathfrak{b}_k$, the mean moving average (MA) parameter μ_k and the standard deviation MA parameter σ_k given by (Ioffe and Szegedy, 2015):

$$f_{\texttt{BatchN}}(a_{ik}^{(1)}) = \mathfrak{b}_k \left(\frac{a_{ik}^{(1)} - \mu_k}{\sigma_k} \right) + \mathfrak{a}_k,$$

where $f_{\texttt{BatchN}}(\cdot)$ is the `BatchN` transformation function and the estimates of μ_k and σ_k condition on the current batch are

$$\hat{\mu}_k = \frac{1}{\mathcal{B}} \sum_{i=1}^{\mathcal{B}} a_{ik}^{(1)} \quad \text{and} \quad \hat{\sigma}_k = \sqrt{\frac{1}{\mathcal{B}-1} \sum_{i=1}^{\mathcal{B}} (a_{ik}^{(1)} - \hat{\mu}_k)^2}.$$

Parameters μ_k, σ_k are non-trainable parameters whereas a_k, $\mathfrak{b}_k$ are trainable, similar to the weight and bias parameters $\boldsymbol{W}^{(l)}$, $\boldsymbol{b}^{(l)}$. In Table 4.1, $m_1 = 45$; hence the number of parameters for `BatchN` is $4 \times 45 = 180$.

Dropout

`DropOut` is a technique where randomly selected neurons in a hidden layer are dropped out during training at a probability $\mathfrak{p}$ such as 5%, 10%, 20%, ..., etc. Their contribution to the activation of downstream neurons is temporally removed on the forward pass, and any weight updates are not applied to the neuron on the backpropagation.

L1 and L2

L1 and L2 regularizations introduce penalty terms to the loss function as follows:

$$\mathcal{J}_{\text{reg}}(\boldsymbol{\theta}) = \mathcal{J}(\boldsymbol{\theta}) + \sum_{j=1}^{M} \mathfrak{w}_j ||\theta_j|| \tag{4.22}$$

based on the weights $\mathfrak{w}_j$ similar to lasso regression in (3.11) with $\lambda = 1$ and $||\theta_j||$ denote the L1 or L2 norm to θ_j. This helps prevent overly large weights, promoting sparsity in the model.

Early stopping

Early stopping refers to some auto-stop criteria such as the number of iterations or epochs $\mathcal{E}$ that has reached, or the pre-determined lower threshold value that the loss function $\mathcal{J}(\cdot)$ has attained (You et al., 2019). As the epoch increases, the loss function

$\mathcal{J}(\cdot)$ often decreases continuously, resulting in a near-perfect fit for the training data. One should track $\mathcal{J}(\cdot)$ for both training and validation/test data across epochs to detect early stopping. An additional parameter *patience* $\mathcal{P}$ to detect early stopping refers to the number of consecutive epochs with no improvement in the performance metric of the validation set before the training process is halted. The performance metric can be the loss function or accuracy measures such as MSE. *Determining the appropriate patience value can be challenging and crucial for achieving the best results.*

By default, early stopping begins from the first epoch. However, in detecting early stopping, one can exclude the initial epochs, which might exhibit transient behavior, and focus on stabilizing phases of the training process. This strategy provides flexibility in managing the early stopping, allows more informed decisions regarding model convergence, and prevents premature stopping during the warm-up period. Python codes are given in Appendix section 4.6.4.

4.2.8 Hyper-parameter tunning

Hyper-parameters of DLNNs including the number of hidden layers $\mathfrak{L}$, the number of neurons m_l in each layer, the activation function $f_l(\cdot)$, the mini-batch size $\mathcal{B}$, the number of epoch $\mathcal{E}$, the learning rate ζ, the dropout rate $\mathfrak{p}$ and the patient $\mathcal{P}$, etc, need to be determined.

Firstly, $\mathfrak{L}$ is an important parameter that describes the architecture of a NN. DLNNs need multiple layers to learn more detailed and abstract relationships within the data. However, more complicated networks require more data and regularization techniques, such as early stopping and dropout, etc., to not overfit the data. If the data have large dimensions or features, $\mathfrak{L}$ is suggested to be 3 to 5. Moreover, a better loss function can increase the network power and potentially reduce $\mathfrak{L}$.

Secondly, the number of hidden neurons $\boldsymbol{m} = (m_1, \ldots, m_L)$ in each layer should be determined. Some researchers suggest it should be between the size of the input layer m_0 and the output layer m_{L+1}. In the first hidden layer (m_1), the size should be

less than twice the size of the input layer (M), that is, $m_1 < 2M$ (Wong, 1991) or $m_1 < 2M + 1$ (Hecht-Nielsen, 1987) or $1 < m_1 < 66$ (Tang and Fishwick, 1993) or $m_1 = (2/3)M$ (Karsoliya and Azad, 2012) or $m_1 = M/2$ (Kang, 1991) or $0.7M < m_1 < 0.9M$ (Buyrukoglu et al., 2021). Buyrukoglu et al. (2021) suggested that m_l should be 70% to 90% of m_0, the size of the input layer, and less than $2m_0$ (Boger and Guterman 1997; Berry and Linoff 2004). On the other hand, Liu et al. (2007) proposed a new criterion based on the estimated signal-to-noise-ratio figure (SNRF) to optimize the number of hidden neurons $\boldsymbol{m}$ to avoid overfitting. Instead of using a separate validation set to detect overfitting, SNRF can quantitatively measure the useful information left unlearned so that overfitting can be automatically detected from the training error. However, the calculation of SNRF is more complicated.

The third choice is the learning rate ζ, which is fixed for the `SGD` algorithm according to (4.15). A large ζ allows the model to learn faster, at the cost of skipping the optimal weights and arriving at a sub-optimal set of weights (Ioffe and Szegedy, 2015). On the other hand, reducing ζ improves the likelihood of convergence but requires more iterations to find the optimal solution. Learning rates ζ can also be searched. Goodfellow et al. (2016) suggests searching ζ from the set {0.1, 0.01, 0.001, 0.0001, 0.00001}. Bengio (2012) suggests that a default ζ value of 0.01 typically works for standard DLNNs. Alternatively, the `Adam` activation function, etc., defined in (4.20) to (4.21), updates the learning rate for each network weight individually.

The fourth choice is the batch size $\mathcal{B}$. Masters and Luschi (2018) stated that training of DLNNs is typically based on mini-batches to reduce memory traces and allow parallel running to shorten the training times. The batch size $\mathcal{B}$ often depends on data, model architecture, and the trade-off of training time, memory usage, regularization, and accuracy. Common batch sizes $\mathcal{B}$ are powers of 2 (e.g., 32, 64, 128) for efficiency reasons and generally, $\mathcal{B} = 32$ is a good initial choice. Larger $\mathcal{B}$ leads to shorter training time, higher memory usage, and possibly lower accuracy. Another issue is the vanishing or exploding gradients when estimating the weight parameters. These problems can be addressed using techniques such as batch normalization, gradient clipping (re-scaling the gradient to keep it small), or other optimizer techniques,

discussed in Section 4.2.6.

Lastly, setting the patience $\mathcal{P}$ in early stopping is a trade-off. A smaller $\mathcal{P}$ may stop training early and save computational resources but might result in premature stopping if the model has not converged. On the other hand, a larger $\mathcal{P}$ allows more epochs for potential improvement but may waste resources and risk overfitting. The appropriate value for $\mathcal{P}$ depends on various factors, such as model complexity, epoch, batch size, learning rates, convergence speed, etc. Moreover, the exclusion of certain initial epochs allows for a better assessment of models' convergences and prevents premature stopping at very early epochs.

In conclusion, searching for these hyperparameters $\boldsymbol{\vartheta}$ is very important to optimize some objective functions. These optimal hyperparameters provide some good DLNN architectures for optimizing loss function $\mathcal{J}(\boldsymbol{\theta})$ for the network parameters $\boldsymbol{\theta}$ in (4.13).

4.2.9 Hyperparameter optimization

The tuning parameters include the number of layers $\mathfrak{L}$, the number of neurons $\boldsymbol{m}$, the number of epochs $\mathcal{E}$, batch size $\mathcal{B}$, learning rates ζ, dropout rate $\mathfrak{p}$ and optimizers, etc. To find the optimal set of hyperparameters, there are four methods to apply. Manual search picks hyperparameter sets based on experience and finds an optimal set. Grid search refers to a search over a grid of hyperparameter space for a set of parameters over all combinations that achieves the optimal of all objective function values, such as optimal validation loss or accuracy. Random search is similar to grid search but considers only a prespecified number of hyperparameter sets randomly drawn from the hyperparameter space. Lastly, Bayesian optimization (BO) uses advanced optimization techniques to improve the search speed using past performances.

This chapter employs two search methods:

Manual search

In this method, manual (MAN) search is combined with the guidelines in Section 4.2.8 to set $\mathfrak{L} = 5$ and $\boldsymbol{m} = (45, 30, 15, 10, 5)$. Activation functions are set as $f_{1:4} = f_{\texttt{Linear}}, f_5 = f_{\texttt{ELU}}$ to ensure the output neurons are non-negative. Then numerous hyperparameter combinations are experimented with early stopping. Experiments are conducted to identify the optimal patience value $\mathcal{P} = 1, 2, 3$. Besides, the early stop is activated only after the 50th epoch, providing the model with an opportunity to stabilize and avoid interference with the initial exploratory phase of training. Each architectural variation may demand different parameter configurations to achieve optimal performance. The process starts with fixed $\mathcal{E} = 400$ and optimizer `AdaDelta` due to its efficiency in handling complex optimization problems. Then it experiments different values of $\mathcal{B}$, ζ, and $\mathfrak{p}$ and the suitable values are $\mathcal{B} = 100, 110$, $\zeta = 0.0097, 0.01$, and $\mathfrak{p} = 0.05, 0.10, 0.20$ where π is the probability of mixture components for the PM model in (3.3). The choices of $\mathcal{B}$ are not exactly the power of 2 but are close to 128. In summary, the choices are

$$\begin{aligned}\text{Manual: } & \mathfrak{L} = 5, \boldsymbol{m} = (45, 30, 15, 10, 5), \mathfrak{p} = 0.05, 0.10, 0.20, f_{1:4} = f_{\texttt{Linear}}, f_5 = f_{\texttt{ELU}}, \mathcal{E} = 400, \\ & \mathcal{B} = 100, 110, \zeta = 0.0097, 0.01, \mathcal{P} = 1, 2, 3, \texttt{optimizer} = \texttt{AdaDelta}. \end{aligned} \tag{4.23}$$

The choice of $\mathfrak{p}$ also depends on the convergence behavior of the models, and $\boldsymbol{m}$ (Table 4.1) is to ensure that the total number of model parameters M should be less than the training data size $N_{\text{T}} = \mathfrak{p}N$. This rule prevents overfitting and enhances the network's capability to capture meaningful patterns.

Bayesian optimization

BO search (Shahriari et al. 2015; Klein et al. 2017; Kandasamy et al. 2018; Masum et al. 2021) uses a probability function based on the hyperparameters. It is a technique for tuning of hyper-parameter set $\boldsymbol{\vartheta}$ consisting of $\mathfrak{L}$, $\boldsymbol{m}$, $\mathcal{E}$, $\mathcal{B}$, ζ and optimizer (Section 4.2.6) to optimize the loss $\mathcal{J}(\boldsymbol{\theta}|\boldsymbol{\vartheta})$ which is the loss or objective function $\mathcal{J}(\boldsymbol{\theta})$ at a

given architecture defined by $\boldsymbol{\vartheta}$. This loss function can be chosen to be NLL or MSE. The strategy is to treat $\mathcal{J}(\boldsymbol{\theta})$ as a random function and place a prior over it. It uses surrogate models like Gaussian processes (GP) to define a prior distribution. Starting with $\mathfrak{n}$ sets of hyperparameters $\boldsymbol{\vartheta}_{1:\mathfrak{n}}$ and function evaluations $\mathcal{J}(\boldsymbol{\theta}|\boldsymbol{\vartheta}_{1:\mathfrak{n}})$, which are treated as data, the GP prior of $\mathcal{J}(\boldsymbol{\theta})$ is updated to form a GP of the posterior distribution. The posterior distribution, in turn, is used to construct an acquisition function (or selection function) such as Expected Improvement (EI)

$$\mathrm{EI}_{\mathfrak{n}}(\boldsymbol{\vartheta}) = \int_{-\infty}^{\infty} \max(\mathcal{J}(\boldsymbol{\theta}|\boldsymbol{\vartheta}_{\mathfrak{n}}^{*}) - \mathcal{J}(\boldsymbol{\theta}|\boldsymbol{\vartheta}), 0) f_{\mathrm{GP}}\Big(\mathcal{J}(\boldsymbol{\theta}|\boldsymbol{\vartheta}_{1:\mathfrak{n}})\Big) d\mathcal{J}(\boldsymbol{\theta}|\boldsymbol{\vartheta})$$

where $\boldsymbol{\vartheta}_{\mathfrak{n}}^{*} := \underset{i \in 1:\mathfrak{n}}{\operatorname{argmax}} \mathrm{EI}(\boldsymbol{\vartheta}_i)$ and $f_{\mathrm{GP}}\Big(\mathcal{J}(\boldsymbol{\vartheta}|\boldsymbol{\vartheta}_{1:\mathfrak{n}})\Big)$ is the density of the posterior GP. Then the next $\boldsymbol{\vartheta}$ to select is $\boldsymbol{\vartheta}_{\mathfrak{n}+1} = \operatorname{argmax} \mathrm{EI}_{\mathfrak{n}}(\boldsymbol{\vartheta})$ based on updated posterior. The search strategy iteratively chooses the next set of hyperparameters $\boldsymbol{\vartheta}_{\mathfrak{n}+1}$ according to acquisition function $\mathrm{EI}(\boldsymbol{\vartheta})$ acquisition function that balances exploration (sampling areas where the surrogate model is uncertain) and exploitation (sampling areas where the surrogate model predicts high performance). The process is repeated for a predefined number of iterations or until a stopping criterion is met. The advantage of BO is that it efficiently explores the hyperparameter space and converges to the optimal $\boldsymbol{\vartheta}$ with a relatively small number of evaluations compared to traditional grid search or random search.

Guided by the manual exploration, the BO search is conducted for the hyperparameter spaces:

$$\begin{aligned} \text{BO:} \quad & \mathcal{E} \in [250, 450], \mathcal{B} \in [75, 120], \zeta \in [0.0097, 0.01], \mathfrak{p} = 0.05, 0.10, 0.20, \\ & \texttt{optimizer} \in \{\texttt{SGD}, \texttt{Adam}, \texttt{AdaDelta}, \texttt{AdaGrad}\}. \end{aligned} \tag{4.24}$$

where $[a, b]$ indicates the inclusive range of values from a to b. This BO search is executed in each of the five folds cross-validation (CV) each repeated 3 times so that 15 distinct combinations are extracted, and the best set of hyperparameters is selected with a minimum score of custom PM MSE from the initial random sampling list of 15 with iterations 10 (see Table 4.2 for BO). The `Python` for implementing BO can

be found in code list 4.3 in Appendix 4.6.3.

4.3 Empirical studies

This section applies the telematics data in Section 3.2 to train the DLNN with the PM model. The PM model was introduced in Section 3.3.1. Section 4.2.9 details the chosen architecture which boasts five hidden layers equipped with neuron size $m_{1:5}$. The network can be described by the following flow chart with distinct activation functions $f_{1:5}$ in bracket:

$$45 \to 45(\texttt{Linear}) \to \texttt{BatchN} \to 30(\texttt{Linear}) \to 15(\texttt{Linear}) \to \texttt{DropOut} \to 10(\texttt{Linear}) \to 5(\texttt{ELU}) \to 2$$

Figure 4.1 and Table 4.1 also present this architecture.

4.3.1 Constraints for safe and risky groups

To implement the PM model, the NLL function is defined in (4.13) which involves annual claims $\hat{a}_{i,1:2}(\boldsymbol{\theta})$ from both safe and risky groups. To ensure that the two groups are well-defined, the predicted annual claims from the safe group should be restricted to be less than or at least mostly less than the those claims from the risky group, that is,

$$\hat{a}_{i1}(\boldsymbol{\theta}) \leq \hat{a}_{i2}(\boldsymbol{\theta}),$$

mostly. To secure such constraint, transformation functions $f_{cg}(\cdot)$ should be applied to the two output neurons, that is,

$$\hat{a}_{ig}(\boldsymbol{\theta}) = f_{cg}(a_{ig}^{(\mathfrak{L}+1)}), \ g = 1, 2 \tag{4.25}$$

instead of using the output neurons $\hat{a}_{ig}(\boldsymbol{\theta}) = a_{ig}^{(\mathfrak{L}+1)}$ directly for the predicted claims as in (4.14). This section proposes the two transformation functions given by

$$\begin{aligned} \hat{a}_{i1} &= f_{c1}(a_{i1}) = 0.5 \cdot \bar{a} \cdot \max\left(0, f_{\texttt{tanh}}(a_{i1})\right) \in (0, 0.0375), \\ \hat{a}_{i2} &= f_{c2}(a_{i2}) = \min\left(0.5 \cdot \bar{a} + 0.5 \cdot \bar{a} \cdot f_{\texttt{softplus}}(a_{i2}), \ 5\right) \in (0.0375, 5) \end{aligned} \tag{4.26}$$

where the output neurons $a_{i1} := a_{i1}(\boldsymbol{\theta}) = a_{i2}^{(\mathfrak{L}+1)}$, $f_{\texttt{tanh}}(\cdot)$ is defined in (4.6), $f_{\texttt{softplus}}(\cdot)$ is defined in (4.7) and $\bar{a} = 0.075$ as given in Section 3.2 is the average of the observed annual claims. The ideas behind these specifications are given below.

For the first constraint, the function $\max(0, f_{\texttt{tanh}}(\cdot))$, similar to the sigmoid function $f_{\texttt{sigmoid}}(\cdot)$, ensures the transformed annual claim stays within the range (0, 1) and the scaling factor $0.5\bar{a}$ maps the outputs to the desired range (0,0.0375). The choice of taking $0.5\bar{a}$ as the threshold between safe and risky groups is arbitrary, but it does not have a great impact on the results.

For the second constraint, the `softplus` function $f_{\texttt{softplus}}$ with a range $(0, \infty)$ ensures positive predicted claims. Then, scaling $0.5\bar{a}$ is applied, and lastly, shifting is conducted using $0.4\bar{a}$ or $0.5\bar{a}$ to achieve the desired output range (0.03, ∞) or (0.0375, ∞) respectively. The first range for the risky group overlaps with (0,0.0375) for the safe group whereas the second range does not overlap with (0,0.0375). These two choices are accessed with MSE and (0.0375, ∞) with $0.5\bar{a}$ shifting is chosen for $\hat{a}_{i2}$ to achieve equilibrium between $\hat{a}_{i,1:2}$ and maintain balance between conservatism and flexibility. Lastly, the range for the risky group is capped by 5 using `min(·,5)` to avoid excessively high predicted annual claims.

The PM MSE function as a performance metric in (4.9) can also be calculated again using (4.25) and (4.26).

4.3.2 Estimation of prior probability π

However, as the prior probability π is fixed in the DLNN, it is not part of the network parameters $\boldsymbol{\theta}$; hence, its optimal value needs to be searched over. There are three ways to estimate the optimal π.

Grid search

The PM DLNN is run for a set of fixed

$$\pi \in \{0.88, 0.89, 0.90, 0.91, 0.92\}$$

and optimal π is chosen according to objective functions such as NLL and MSE. The choice of π is based on the observed 0.92 proportion of zero claims for the telematics data and the belief that it is more likely to have risky zero-claim drivers than safe non-zero-claim drivers.

To ensure robust and reliable learning and balance the uncertainty due to fixing π, the proposed model undergoes refinement through ten repeat runs and each repeat comprises 2-3 trials to ensure that a stable model is obtained for each repeat. Each trial of each repeat uses a different seed. Instead of averaging the models across repeats, this strategy selects a model that optimizes an objective function such as MSE in (4.9). The repeats facilitate a thorough exploration of the networks' intricacies, contribute to identifying optimal settings and configurations, and provide valuable insights to understand networks' capabilities, so that they ultimately lead to fine-tuning for optimal outcomes. The chosen best sub-models should showcase convergence path in the epoch history plot with a consistent and progressively descending trajectory to reflect a robust and reliable learning process. Moreover, it should have a low MSE so that the predicted annual claims in the scatter plot against observed align along the 45° diagonal line. The `Python` code list 4.4 in Appendix 4.6.4 is provided to guide the implementation of the DLNN with the PM model. Conditional on $\pi = 0.88$ say, the models using architectures from Manual and BO search are called MAN-0.88 and BO-0.88, respectively.

Iterative conditional optimization

To conduct a more scientifically rigorous estimation of π, the estimation process consists of two iterative steps: one for updating the DLNN parameter $\boldsymbol{\theta}$ given π

and another for updating π given the DLNN parameter $\boldsymbol{\theta}$ using the PMNLL loss function in (4.13) which involves both parameters $\boldsymbol{\theta}$ and π. This iterative conditional optimization (ICO) process is repeated until the algorithm converges. In the training of π, it is conditional on the current DLNN and is adjusted based on its impact on the overall loss, considering the interplay between model predictions and the custom loss function. Another set of gradients, apart from those for the trainable network parameters $\boldsymbol{\theta}$, is computed for π based on the same loss function in (4.13). These gradients suggest adjusting an update value of π to minimize the overall loss. The whole procedures require inter epoch of size 50 and outer epoch of size 100. As the estimation of π changes along outer epoch allowing some flexibility, no repeats or reruns are conducted. The choice of architectures is based on Manual search for the grid search model.

Dynamic prior probability

As mentioned above, the assumption of taking the prior probability π in (3.3) as fixed presents some challenges in estimation. One consideration is to allow dynamic prior probabilities (DP) by setting π_i to vary across drivers i in the network and assigning the *third neuron* in the output layer to estimate π_i. To ensure $\pi_i \in (0.88, 0.93)$, the third constraint for π_i is

$$0.88 + 0.05 \cdot f_{\texttt{sigmoid}}(\pi_i)$$

where $f_{\texttt{sigmoid}}(\cdot) \in (0, 1)$. This model is different from the models assuming a fixed π, but as π_i is restricted to a small range of (0.88,0.93), one would expect the results to be similar to those with π fixed. Note also that this dynamic prior probability π_i is still different from the posterior probability $\mathfrak{z}_i$ in (3.6) and $\mathfrak{z}_i$ can be calculated using (3.6) by replacing π with π_i. The models are based on architectures from MAN search

$$\begin{aligned}\text{DP-MAN: } & \mathfrak{L} = 5, \boldsymbol{m} = (45, 30, 15, 10, 5), \mathfrak{p} = 0.20, f_{1:4} = f_{\texttt{Linear}}, f_5 = f_{\texttt{ELU}}, \mathcal{E} = 300, \\ & \mathcal{B} = 120, \zeta = 0.0097, \mathcal{P} = 1, \texttt{optimizer} = \texttt{AdaDelta}. \end{aligned} \tag{4.27}$$

and BO search with hyperparameter space

$$\begin{aligned} \text{DP-BO:} \quad & \mathcal{E} \in [200, 350], \mathcal{B} \in [90, 150], \zeta \in [0.0095, 0.01], \mathfrak{p} = 0.05, 0.10, 0.20, \\ & \texttt{optimizer} \in \{\texttt{SGD}, \texttt{Adam}, \texttt{AdaDelta}, \texttt{AdaGrad}\} \end{aligned} \tag{4.28}$$

and they are called DP-MAN and DP-BO, respectively. Since the dynamic π_i provides some flexibility in the network, no repeats or reruns is taken.

4.4 Experimental results

These DNLLs are applied to fit the telematics data with a train-test random split. The training data contains $\mathsf{p} = 80\%$ of $N = 14,157$ observations. The architectures from manual search are provided in (4.23) whereas the architectures from BO search are obtained by searching from the hyperparameter space (4.24).

Grid search

Results of the optimal submodel from ten repeats of each π using MAN and BO search are reported in Table 4.2. The best models for each architecture search method, namely MAN-0.88 and BO-0.89, are selected according to prediction accuracy using the PM MSE in (4.9), and they are highlighted in yellow. This PM MSE compares the network accuracy in terms of annual claims of the observed and predicted using posterior aggregated output neurons. Table 4.2 also reports the more common MSE in (4.12) comparing model accuracy in terms of claim counts of the observed and predicted claims using PM model. The MSE, as well as NLL, is cross-classified by training and validation sets. The convergence and stability of the two best models are checked from the epoch history of NLL showing slow downward trends consistent for both training and validation data.

To access classification performance, the posterior probabilities $\mathfrak{z}_{i1}$ in (3.6) and $1 - \mathfrak{z}_{i1}$ are plotted in Figure 4.2 for the three best models. The distributions of $\mathfrak{z}_{i1}$ and $1 - \mathfrak{z}_{i1}$

indicate clear classification to the safe group for the majority of drivers as most of the $\mathfrak{z}_{i1} \simeq 1$ as expected since 92% of drivers have zero claims. However, a small portion of drivers with $\mathfrak{z}_{i1} \simeq 0.5$ shows some uncertainties in classifying safe drivers. Driver classification can be obtained by calculating the posterior group memberships based on the if-else conditions on the posterior group probabilities $\mathfrak{z}_{i1}$, that is, driver i is classified to safe group $\mathcal{G}_1$ if $\mathfrak{z}_{i1} > 0.5$; otherwise, he/she is classified to risky group $\mathcal{G}_2$. Results show that there is a small portion of drivers with $\mathfrak{z}_{i1} > 0.5$ (3.40% for model MAN-0.88 and 2.67% for model BO-0.89) and they are classified as risky drivers.

Lastly, the prediction accuracy overall and by driver groups can be visualized in the scatter plots of predicted marginal posterior annual claims in (4.10) against observed annual claims graphed in Figure 4.3. Both selected best models exhibit desirable agreement of predicted annual claims as the majority of blue points with one claim align closely to the diagonal line without any discernible patterns. Moreover, the small red dot indicates that the majority of zero-claim drivers are predicted to have close to zero annual claims. These drivers are classified mostly to the safe group. Moreover, the right plots indicate that all drivers with at least two claims are risky drivers, as expected, and they are predicted to have higher than observed annual claims (above the diagonal line). In summary, the amount of deviation of the points from the diagonal line is reflected in the PM MSE measure.

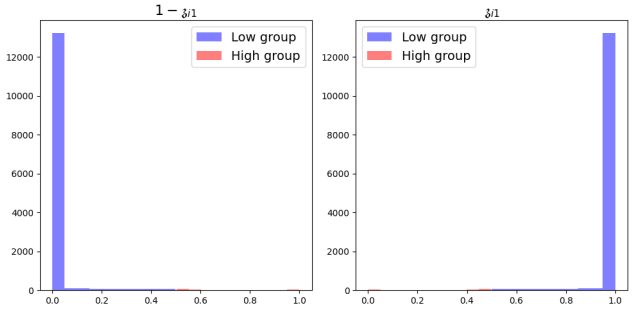
Iterative conditional optimization

Table 4.2 and Figures 4.2 and 4.3 report results using the DP method, similar to the grid search method. The estimate for π is 0.92 at epoch 80 which shows minimal NLL for the validation set. Moreover, about 0.9% of drivers are classified as risky drivers. Although PM NLL values defined in (4.13) are lower using this model for both training and validation data, the PM MSE values defined in (4.9) are larger showing more discrepanies between observed and predicted annual claims. These results can be seen from Figure 4.3 in which the points are further away from the diagonal line. As a result, this ICO model is not selected.

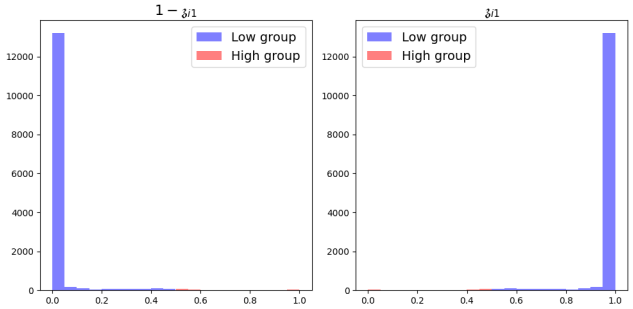
Table 4.2 – Model performance for 15 DLNNs with MAN search to obtain the hyperparameter set in (4.23) and BO search from the hyperparameter space in (4.24). Each grid search model is repeated 10 times, and the best model is selected if the epoch history shows convergence and the PM MSE is minimum, highlighted in yellow. The averaged π value for both DP-MAN and DP-BO is 0.93.

π	$\mathcal{B}$	$\mathcal{E}$	ζ	optimizer	p	mean NLL		mean MSE metrics		Measures
						Training	Validation	Training	Validation	PM MSE
Grid search method using architecture from Manual search										
0.88	110	400	0.0100	AdaDelta	0.20	0.3883	0.4124	0.1144	0.1379	**0.0129**
0.89	110	400	0.0097	AdaDelta	0.20	0.4001	0.3828	0.1212	0.1087	0.0135
0.90	110	400	0.0097	AdaDelta	0.20	0.4091	0.3914	0.1222	0.1097	0.0279
0.91	110	400	0.0097	AdaDelta	0.20	0.4114	0.4066	0.1191	0.1203	0.0154
0.92	110	400	0.0097	AdaDelta	0.20	0.4189	0.4139	0.1194	0.1207	0.0223
0.93	110	400	0.0100	AdaDelta	0.20	0.4256	0.4206	0.1195	0.1208	0.0173
Grid search method using architecture from Bayesian optimization (BO) search										
0.88	93	267	0.0097	SGD	0.10	0.3915	0.3999	0.1190	0.1183	0.0144
0.89	93	267	0.0097	SGD	0.20	0.3972	0.4057	0.1193	0.1186	**0.0123**
0.90	107	337	0.0098	SGD	0.10	0.3979	0.4214	0.1141	0.1352	0.0158
0.91	120	396	0.0099	SGD	0.20	0.4067	0.4315	0.1153	0.1382	0.0161
0.92	99	291	0.0098	SGD	0.20	0.4132	0.4386	0.1154	0.1368	0.0228
0.93	107	337	0.0098	SGD	0.20	0.4209	0.4465	0.1158	0.1371	0.0233
Iterative conditional optimization (ICO) $\hat{\pi} = 0.92$										
MAN	110	80	0.0097	AdaDelta	0.10	0.1302	0.1357	0.0652	0.0767	0.0675
Dynamic prior probability (DP) $\bar{\pi} = 0.93$										
MAN	120	300	0.0097	AdaDelta	0.20	0.4070	0.4156	0.1186	0.1174	**0.0140**
BO	97	232	0.0096	SGD	0.20	0.4206	0.3798	0.1192	0.1242	0.0141

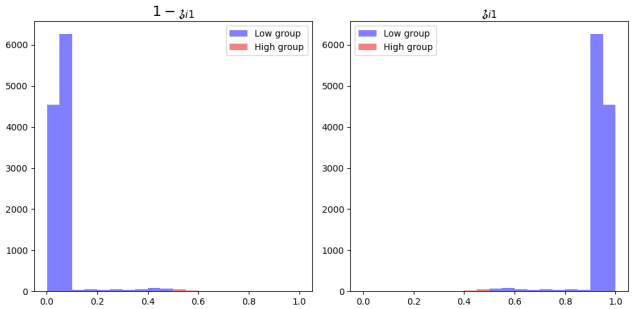
Figure 4.2 – Distribution of posterior probabilities $\mathfrak{z}_{i1}$ and $1-\mathfrak{z}_{i1}$ for model MAN-0.88 (row 1), model BO-0.89 (row 2), model ICO (row 3) and model DP-MAN (row 4).



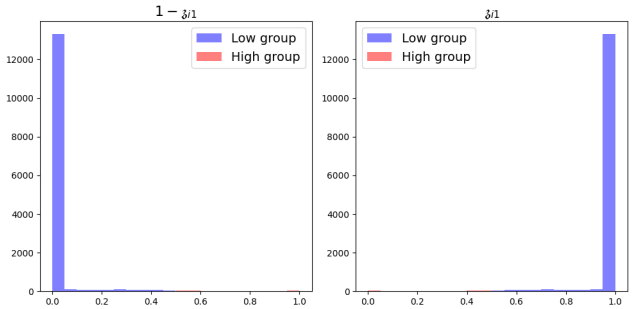
Model MAN-0.88 $\mathcal{E} = 400$; $\mathcal{B} = 110$; $\zeta = 0.0100$



Model BO-0.89 $\mathcal{E} = 267$; $\mathcal{B} = 93$; $\zeta = 0.0097$

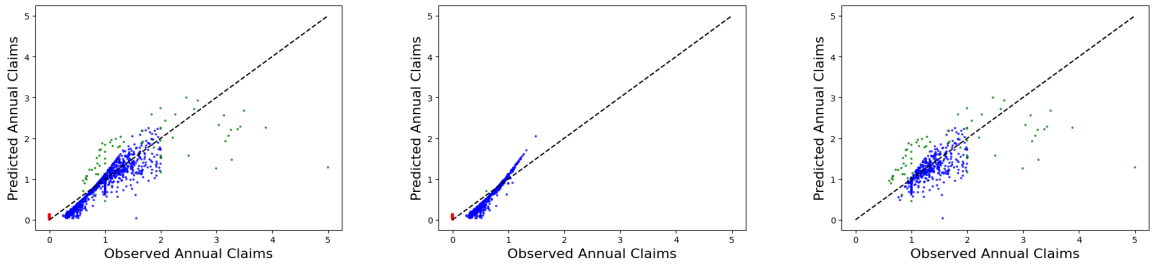


Model ICO $\mathcal{E} = 80$; $\mathcal{B} = 110$; $\zeta = 0.0097$



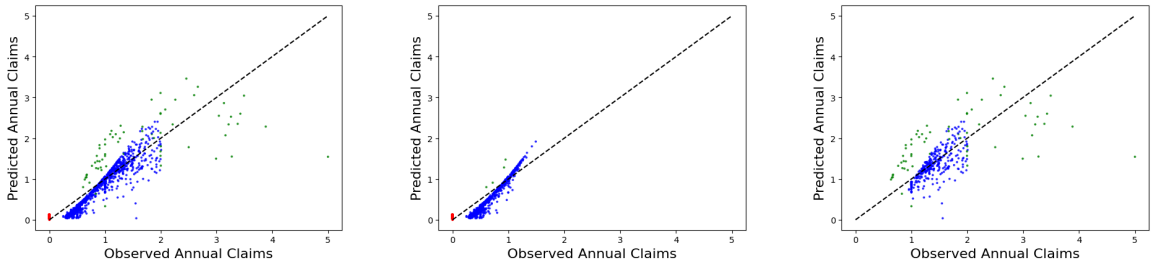
Model DP-MAN $\mathcal{E} = 300$; $\mathcal{B} = 120$; $\zeta = 0.0097$

Figure 4.3 – Predicted marginal posterior annual claims against observed annual claims for the best models, MAN-0.88 (row 1), BO-0.89 (row 2), ICO (row 3) and DP-MAN (row 4) for the all (left), low (middle), and high (right) claim groups. Red, blue, and green dots denote 0, 1, and ≥2 observed claims, respectively.



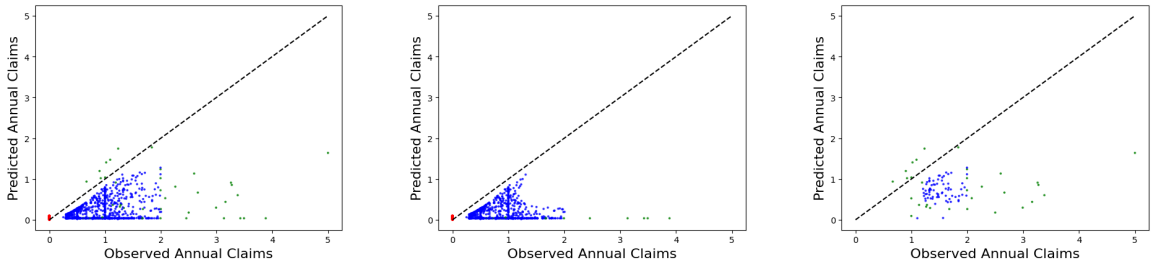
(a) Overall **(b)** Low group **(c)** High group

Model MAN-0.88 $\mathcal{E} = 400$; $\mathcal{B} = 110$; $\zeta = 0.01$; PM MSE = 0.0129



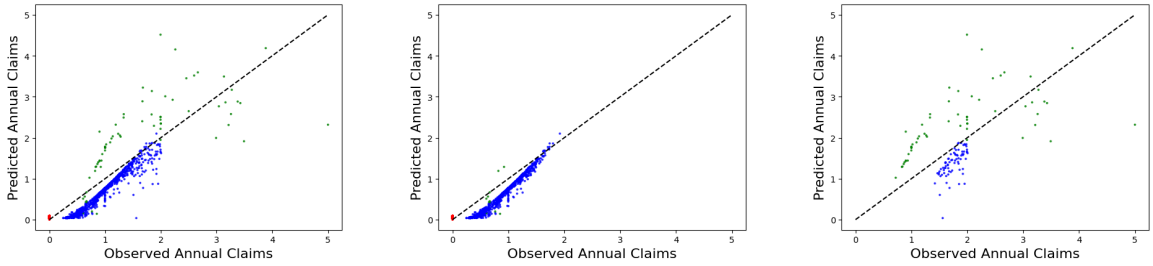
(d) Overall **(e)** Low group **(f)** High group

Model BO-0.89 $\mathcal{E} = 267$; $\mathcal{B} = 93$; $\zeta = 0.0097$; PM MSE = 0.0123



(g) Overall **(h)** Low group **(i)** High group

Model ICO $\mathcal{E} = 80$; $\mathcal{B} = 110$; $\zeta = 0.0097$; PM MSE = 0.0675



(j) Overall **(k)** Low group **(l)** High group

Model DP-MAN $\mathcal{E} = 300$; $\mathcal{B} = 120$; $\zeta = 0.0097$; PM MSE = 0.0140

Dynamic prior probability

Similarly, Table 4.2 and Figures 4.2 and 4.3 also report results using the DP method. The proportion of drivers classified into risky groups is 0.93% for the DP-MAN model and 1.08% for the DP-BO model. Between using MAN and BO architecture search, the DP-MAN model using MAN architecture search is chosen to be the best model according to PM MSE. For this chosen model, the average of π_i overall drivers is $\bar{\pi} = 0.93$, and it has slightly higher PM MSE than MAN-0.88 and BO-0.89 models. Figure 4.3 shows that DP-MAN slightly underestimates annual claims for the safe group and slightly overestimates annual claims for the risky group.

4.5 Conclusion

The complexity of the telematics data, characterized by excessive zero claims counts and 45 noisy features, necessitates a nuanced understanding. This chapter underscores the practical applicability of PM DLNN as potent tools for analyzing the telematics data and predicting driver groups and claims. The comprehensive discussion traverses the intricacies of the model's development, compilation, and training, specifically emphasizing the innovative PM DLNN using NLL loss function, PM MSE prediction accuracy metric, and the strategic utilization of custom constraints on output neurons to distinctly identify the two driver groups. The contribution of the PM DLNN research is multifaceted, addressing fundamental challenges and pushing the boundaries of modeling precision.

Firstly, the research begins with a careful search of architectures based on experience, guidelines from literature, and efficient BO search techniques. Architecture search involves fine-tuning parameters such as epochs, batch size, learning rates, and optimizers, including techniques like batch normalization, early stop, and dropout to enhance convergence and balance the trade-off between under and over-fittings. The deliberate and strategic use of manual and BO searches further enhances the model effectiveness and balance model efficiency and computational resources. The man-

ual search is an initial exploration, identifying optimal hyperparameter combinations and providing insights into the data's intricacies and model complexity. The manual search also aids in deciphering these nuances, enabling researchers to establish appropriate ranges of hyperparameters for subsequent more advanced and efficient BO.

Secondly, introducing the designated PM NLL and PM MSE with two mean parameters addresses a notable challenge, including the estimation of the non-regression prior probability parameter π. Two methods, grid search, and iterative conditional optimization, are proposed to ensure that the networks align with the underlying dynamics of the data, emphasizing precision in reflecting real-world scenarios. Additionally, the incorporation of constraints for the ranges of predicted annual claims, $a_{i1} \in (0, 0.0375)$ and $a_{i2} \in (0.0375, 5)$ reflects a strategic effort to ensure that the two driver groups are well-defined.

Lastly, the simultaneous estimation of network parameters $\boldsymbol{\theta}$ and non-network parameter π, which is the prior probability of the mixture components of the PM model, presents challenges to model implementation. The common grid search strategy is first employed using six π values, and a PM DLNN is run with ten repeats for each π. The ten repeats provide a refinement to the optimal PM DLNN and enhance robustness and adaptability. Two PM DLNNs are selected according to PM MSE for the architectures using manual and BO search, respectively. They provide insights into the iterative conditional optimization method, a novel proposed method to iteratively estimate $\boldsymbol{\theta}$ condition on π and then estimate π condition on $\boldsymbol{\theta}$ until convergence or early stop criteria are met. Lastly, the dynamic π method allows π to vary within (0.88,0.93) to be a network parameter assigned to the third output neuron. The implementation of the three methods utilizes `Keras` and `TensorFlow` in `Python`, and model performance is assessed through visualizations and numerical measures, such as NLL and MSE, to evaluate model fit and prediction accuracy respectively. These best-chosen models predict the margin posterior annual claims in (4.10) to high accuracy as they align well with the diagonal line in the scatter plot of predict annual claims against observed, reflecting a robust and accurate representation of the un-

derlying data dynamics using the best-chosen models. In conclusion, experimental results validate the effectiveness of the PM DLNN in classifying drivers and accurately predicting drivers' claims. Then, the UBI experience-rating method in Section 3.5 can be applied to calculate UBI premiums as feedback to driving behaviors more frequently and more closely.

4.6 Appendix

4.6.1 Define PM NLL function with fixed π

To implement the PM DLNNs, the NLL function in (4.13) with $\mu_{ig} = n_i\hat{a}_{ig}(\boldsymbol{\theta}) = n_i a_{ig}^{\mathfrak{L}+1}$ called `mixture_poisson_loss` is defined in `Python`. The breakdown of the methods in the code performs the following:

- The true values are `y_true` and the predicted values are `y_pred`.
- a_{i1} (`a1 = y_pred[:, 0]`) and a_{i2} (`a2 = y_pred[:, 0]`) extract the predicted outputs for two groups. These values are used in the calculation of NLL.
- `n = y_true[:, -1]` extracts the exposure column values (n_i) from the y array and `y = y_true [:, :-1]` extracts the claim (y_i) values by first removing the last columns (n_i) from the y array, leaving only the claim values for two groups.
- Constraints are imposed on two predicted outputs, where a_{i1} is designated as low claims and a_{i2} as high claims. The specifications are as follows:

  Constraint for a_{i1}: `a1 = 0.5*max(0.0, tanh(a1))* a_bar`

  Constraint for a_{i2}: `a2 = min(0.5*a_bar+0.5*softplus (a2)*a_bar, 5)`

Code Listing 4.1 – Customise PM NLL in `Python`

```
# Define the custom loss function
def mixture_poisson_loss_with_exposure(y_true, y_pred):
    a1 = y_pred[:, 0]  # Extracting the predicted output a1
```

```
    a2 = y_pred[:, 1]  # Extracting the predicted output a2

    pi = 0.92  # Define the pi values
    a_bar = 0.075  # annual claim 0.0751375
    n = y_true[:, -1]
    y = y_true[:, :-1]
    # Small epsilon value to avoid zero or negative values in
        logarithm
    epsilon = 1e-10

    # Apply the constraint a1 < a2, a1 (0,0.0375), a2 (0.0375, 5)
    a1 = 0.5 * tf.math.maximum(0.0, tf.math.tanh(a1)) * a_bar
    a2 = tf.minimum(0.5 * a_bar + 0.5 * tf.math.softplus(a2) * a_bar
        , 5)

    mu1 = tf.multiply(a1, n)
    mu2 = tf.multiply(a2, n)

    p1 = tf.divide(
        tf.multiply(tf.math.pow(mu1, y), tf.exp(-mu1)),
        tf.exp(tf.math.lgamma(y + 1))
    )
    p2 = tf.divide(
        tf.multiply(tf.math.pow(mu2, y), tf.exp(-mu2)),
        tf.exp(tf.math.lgamma(y + 1))
    )

    loss0 = -tf.math.log(tf.multiply(pi, p1) + tf.multiply((1 - pi),
        p2))
    loss = tf.reduce_mean(loss0)

    return loss
```

4.6.2 Define PM MSE function with fixed π

Code Listing 4.2 – Customise PM MSE metrics in Python

```python
# Define the custom MSE metrics function
def pmmse(y_true, y_pred):
    a1 = y_pred[:, 0]  # Extracting the predicted output a1
    a2 = y_pred[:, 1]  # Extracting the predicted output a2

    pi= 0.92  # Define the pi values
    a_bar = 0.075 # claim 0.0751375

    n = y_true[:, -1]
    y = y_true[:, :-1]

    epsilon = 1e-10  # Small epsilon value to avoid zero or negative
        values in logarithm

    # Apply the constraints, a1 (0,0.0375), a2 (0.0375,5)
    a1 = 0.5 * a_bar * tf.math.maximum(0.0, tf.math.tanh(a1))
    a2 = tf.minimum(0.5 * a_bar + 0.5 * tf.math.softplus(a2) * a_bar
       ,5)

    m1 = tf.multiply(a1, n)
    m2 = tf.multiply(a2, n)

    p1 = tf.divide(
        tf.multiply(tf.math.pow(m1, y), tf.exp(-m1)),
        tf.exp(tf.math.lgamma(y + 1))
    )
    p2 = tf.divide(
        tf.multiply(tf.math.pow(m2, y), tf.exp(-m2)),
        tf.exp(tf.math.lgamma(y + 1))
    )

    #posterior prob of group 1
    z1= tf.divide(
    (tf.multiply(pi, p1)),
```

```
    (tf.multiply(pi, p1) + tf.multiply((1 - pi), p2))
    )

    # ahat using posterior prob
    ahat=tf.multiply(z1, a1) + tf.multiply((1 - z1), a2)

    pmmse = tf.reduce_mean(tf.square((y/n) - ahat))

    return pmmse
```

4.6.3 Define Bayesian optimization

The package `:bayes_opt` in `Python` is implemented for BO search. The breakdown of each code is explained below:

- `init_points=15` specifies 15 initial random samples.
- `n_iter=10` specifies 10 iterations.
- `nn_cl_bo`) defines a DLNN model with hyperparameters and returns a PM MSE score.
- `optimizerL` contains the names of optimizers and `optimizerD` is dictionary maps of the optimizer names with the specified learning rate (`lr`).
- `nn_poi_fun` is an inner function that defines the structure of the DLNN with specified neurons and `DropOut` rate. The model is compiled with PM NLL loss and the specified optimizer with the learning rate.
- `KerasRegressor` is created with the defined model function `nn_poi_fun` with the specified number of epochs and batch size.
- `KFold` conducts K-fold cross-validation is used with five splits, shuffling the data, and a fixed random state for reproducibility.

- `cross_val_score` defines the score for cross-validation within the `KerasRegressor` model, and a customized PM MSE metric in (4.9) is used as the scoring metric.

Code Listing 4.3 – Bayesian optimization search in `Python`

```
# Create a custom scoring function using the customized PMMSE
def custom_scoring(y_true, y_pred):
    # Convert numpy arrays to TensorFlow tensors
    y_true = tf.constant(y_true, dtype=tf.float32)
    y_pred = tf.constant(y_pred, dtype=tf.float32)

    # Calculate the score using the pmmse
    score = pmmse(y_true, y_pred)

    # Convert TensorFlow tensor back to numpy array and return the
        negative value (for minimization)
    return score.numpy()

# Create function
def nn_cl_bo(optimizer, learning_rate, batch_size, epochs):
    optimizerL = ['SGD', 'Adam', 'Adadelta', 'Adagrad']
    optimizerD= {'Adam':Adam(lr=learning_rate), 'SGD':SGD(lr=
        learning_rate),
    'Adadelta':Adadelta(lr=learning_rate),
                 'Adagrad':Adagrad(lr=learning_rate)}
    optimizer = optimizerD[optimizerL[round(optimizer)]]
    batch_size = round(batch_size)
    epochs = round(epochs)
    def nn_cl_fun():
        nn = Sequential()
        nn.add(Dense(45, input_dim=45, activation='linear'))
        nn.add(BatchNormalization())
        nn.add(Dense(30, activation='linear'))
        nn.add(Dense(15, activation='linear'))
        nn.add(Dropout(0.05))
        nn.add(Dense(10, activation='linear'))
        nn.add(Dense(5, activation='elu'))
        nn.add(Dense(2))
```

```
32          nn.compile(loss=mixture_poisson_loss_with_exposure,
                optimizer=optimizer)
33          return nn
34      es = EarlyStopping(monitor="val_loss",verbose=0, patience=1)
35      nn = KerasRegressor(build_fn=nn_cl_fun, epochs=epochs,
            batch_size=batch_size,
36      verbose=0)
37      kfold = KFold(n_splits=5, shuffle=True, random_state=123)
38      score = cross_val_score(nn, x_train, y_train, scoring=
            make_scorer(custom_scoring,
39      greater_is_better=False), cv=kfold, fit_params={'callbacks':[es
            ]}).mean()
40      return score
41
42 params_nn ={
43      'optimizer':(0,3),
44      'learning_rate':(0.0097, 0.01),
45      'batch_size':(75,120),
46      'epochs':(250, 450),
47 }
48
49 # Run Bayesian Optimization
50 nn_bo = BayesianOptimization(nn_cl_bo, params_nn,random_state=121)
51 nn_bo.maximize(init_points=15, n_iter=10)
```

4.6.4 Define PM DLNN architecture using grid search for π

- The command `StandardScaler()` is used to normalize the 45 features so that each feature will have $\mu = 0$ and $\sigma = 1$.
- The `start_from_epoch` parameter in `keras.callbacks.EarlyStopping` allows to specify from which epoch the callback should start monitoring for early stopping.

Code Listing 4.4 – PM DLNN architecture in Python

```python
# Define the input shape
input_shape = (45,)

# Define the neural network model
inputs1 = layers.Input(shape=input_shape)
x1 = layers.Dense(45)(inputs1)
x1 = layers.BatchNormalization()(x1)
x2 = layers.Dense(30)(x1)
x3 = layers.Dense(15)(x2)
x3 = layers.Dropout(0.05)(x3)
x4 = layers.Dense(10)(x3)
x5 = layers.Dense(5, activation="elu")(x4)

# Define the output layers
a_bar = 0.075
apred01 = layers.Dense(1, activation="tanh")(x5)
apred1 = 0.5 * a_bar * tf.math.maximum(0.0, apred01) #(0,0.0375)

apred02 = layers.Dense(1, activation="softplus")(x5)
apred2 = tf.math.minimum(0.5 * a_bar + 0.5 * apred02 * a_bar,5)

output_params = layers.Concatenate(name="pvec", axis=1)([apred1,
    apred2])
model = Model(inputs=inputs1, outputs=output_params)

# Compile the model (optional, but required for some operations)
model.compile(optimizer=tf.keras.optimizers.Adadelta(learning_rate
    =0.01),
loss=mixture_poisson_loss_with_exposure, metrics=[pmmse])

# Lists to store training and validation loss history
train_loss_history = []
val_loss_history = []
train_pmmse_history = []
val_pmmse_history = []
ahat = []
```

```
ahat_results = []

# Number of repeats
num_repeats = 10

# Early stopping callback
early_stopping = EarlyStopping(monitor="val_pmmse", patience=1)

# Training loop
for epoch in range(num_repeats):
    print(f"Epoch {epoch + 1}/{num_epochs}")

    # Generate a new random seed for each loop iteration
    seed_value = np.random.randint(40, 50)
    tf.random.set_seed(seed_value)

    # Randomly split the data into a 20% test set
    X_train, X_test, YY_train, YY_test = train_test_split(XX, YY,
        test_size=0.2,
    random_state=seed_value)

    # Train the model on the training data
    history = model.fit(X_train, YY_train, epochs=400, batch_size
        =110, verbose=1,
    validation_data=(X_test, YY_test), callbacks=[early_stopping])

    # Store the loss history for this fold
    fold_train_loss = history.history['loss']
    fold_val_loss = history.history['val_loss']

    # Store the training and validation loss for this epoch
    train_loss_history.append(history.history['loss'])
    val_loss_history.append(history.history['val_loss'])

    # Store the loss history for this fold
    fold_mse_loss = history.history['pmmse']
    fold_mse_loss = history.history['val_pmmse']
```

```
70
71      # Store the training and validation loss for this epoch
72      train_mse_history.append(history.history['pmmse'])
73      val_pmmse_history.append(history.history['val_pmmse'])
74
75      # Predictions for train and test sets
76      ahat = model.predict(XX)
77      # Append the predictions for this epoch
78      ahat_results.append(ahat)
79
80 #Extract predicted values from best iteration in epoch history
81 ahat1=pd.DataFrame(ahat_results[0])
82
83 YY.reset_index(drop=True, inplace=True)
84 ahat1.reset_index(drop=True, inplace=True)
85
86 df_results = pd.concat([YY, ahat1], axis=1)
87 df_results.columns = ['y','n', 'a1', 'a2']
88
89 frames = [df_results]
90 result = pd.concat(frames)
91
92 # Extract the predicted output
93 a1 = result['a1']
94 a2 = result['a2']
95 n=result['n']
96 y=result['y']
97
98 # Calculate the mean functions
99 mu1=n*a1
100 mu2=n*a2
101 # Assuming mu1, mu2, a1, a2, and y are column vectors or Numpy
      arrays
102 mu1 = np.array(mu1)  # Replace with the actual values
103 mu2 = np.array(mu2)  # Replace with the actual values
104 a1 = np.array(a1)  # Replace with the actual values
105 a2 = np.array(a2)  # Replace with the actual values
```

```
y = np.array(y)  # Replace with the actual values
# Calculate the density p1 and p2
p1 = ((mu1 ** y) * tf.math.exp(-mu1)) / tf.math.exp(tf.math.lgamma(y
    + 1))
p2 = ((mu2 ** y) * tf.math.exp(-mu2)) / tf.math.exp(tf.math.lgamma(y
    + 1))
#Calculate posterior probabilities
Zi1 = pi*p1/(pi*p1+(1-pi)*p2)
Zi2 = 1-Zi1

result['mu1']=mu1
result['mu2']=mu2
result['p1']=p1
result['p2']=p2
result['Zi1']=Zi1
result['Zi2']=Zi2
result['a1*Zi1']=a1*Zi1
result['a2*Zi2']=a2*Zi2

# Define function agreement
def agreement(row):
    a1 = row['a1']
    a2 = row['a2']

    if a1 < a2:
        agreement = 0
    else:
        agreement = 1

    return agreement

# Define function group
def group(row):
    Zi1 = row['Zi1']
    Zi2 = row['Zi2']

    if Zi1 > 0.5:
```

```
141         group = 0
142     else:
143         group = 1
144 
145     return group
146 
147 # Apply the function to each row using axis=1
148 result['ahat'] = result['a1*Zi1']+result['a2*Zi2']
149 result['agreement'] = result.apply(agreement, axis=1)
150 result['group'] = result.apply(group, axis=1)
151 
152 #Performance measure PM MSE
153 from sklearn.metrics import mean_squared_error
154 PMMSE=mean_squared_error(result['y'].values/result['n'].values,
      result['ahat'].values)
```

Chapter 5

Conclusion and future research

By conducting meticulous data analysis and employing robust statistical and machine learning techniques, insurance companies can enhance the predictive capabilities of their loss reserves and loss claims tools. This advancement facilitates a more profound understanding, more accurate measurement, and effective management of the risks they encounter. The proposed techniques outlined in this chapter, focusing on loss reserving and telematics insurance, are poised to contribute significantly to advancing actuarial practices. To wrap up this thesis, this concluding chapter summarizes the significant findings from the preceding chapters. Additionally, it highlights avenues for future research and explores potential applications in industrial and practical contexts.

5.1 Loss reserve models for run-off triangle data

Chapter 2 presents four critical contributions to the field of loss reserving. Firstly, it introduces a novel approach by incorporating the persistence of the development year effect using the CARR model, enriching the conventional ANOVA and ANCOVA models for accident year, development year, and calendar year effects. The observed losses suggest linear trends for accident and calendar year effects and quadratic trends for the development year effect, which are integrated into the ANCOVA models.

Secondly, the chapter extends the data distribution to GB2 and compares it with the GG, Weibull, and EE distributions. It provides details of model implementation, including commands for implementing the four distributions and performing posterior simulations.

Thirdly, the chapter demonstrates how to forecast partial and calendar year reserves using run-off triangle data. This involves forecasting losses in the lower run-off triangle and the diagonals of the trapezoidal data below the run-off triangle. Additional assumptions are introduced to enable forecasts using ANOVA models and the trapezoidal CL method. Model selection involves 41 mean functions, both with and without the development year persistence effect, applied to the four distributions, with the GB2-$\beta\delta_l\gamma_1\gamma_2$ model identified as the best choice.

Finally, a series of validation experiments assess the reliability of the `Stan` package, the robustness of the CARR model, and the accuracy of forecasts. These experiments include simulation-based validation tests and scenarios, confirming the model's robustness and forecasting accuracy.

In conclusion, the study affirms that the GB2-$\beta\delta_l\gamma_1\gamma_2$ model stands out as the most reliable choice for calendar year reserve forecasting, marking a promising methodological advancement in loss reserve prediction and forecasting.

5.2 Claim count models for telematics data

Chapter 3 presents a comprehensive analysis of claim data from 14,157 drivers, revealing equi-dispersion and a substantial 92% prevalence of zero claims. The proposed two-stage TP, PM, and ZIP regressions and various regularization techniques offer a robust framework for predicting annual claims, selecting predictive driver variables, and classifying drivers into low or high-claim groups. The empirical findings underscore the effectiveness of adaptive lasso with elastic net regularization, particularly in TP models. Furthermore, Chapter 3 introduces a novel UBI experience-rating method derived using claims and driver groups predicted from the best PMA model.

This method revolutionizes premium pricing by rewarding drivers for recent good driving rather than relying solely on historical claim records. Moreover, instead of revising premiums annually in traditional auto insurance policies, the proposed UBI premium pricing method allows for more frequent updates based on recent driving behavior, fostering safer driving habits, reducing cross-subsidization of risky drivers, and enhancing loss reserving for auto insurance companies.

Moreover, exploring DLNNs in analyzing telematics data offers a compelling direction, as these networks could capture more complex dynamics and interactions among DVs without explicit variable selection. These extensions are further investigated in Chapter 4.

Chapter 4 affirms that DLNNs are pragmatic learning tools to discern intricate patterns within the data and hence, better predict drivers' claims and groups. The extensive exploration covers crucial aspects of models' development, compilation, and training. It focuses on customizing the PM NLL loss function and PM MSE prediction accuracy metric, adopting techniques like batch normalization, early stop, and dropout to enhance convergence, robust generalization to different data and prevention of overfitting, searching for optimal architectures/tuning parameters using manual and BO methods, implementing custom output constraints to empower sensible ranges of predicted annual claims to distinguish safe and risky driver groups, estimating the non-regression prior probability parameter π in PM model using grid search and iterative conditional optimization methods, and lastly, repeating each model in grid search ten times to refine the model and balance the uncertainty due to fixing π. The best-performing models, MAN-0.88, BO-0.89, and DP-MAN, provide good prediction accuracy as the predicted posterior marginal annual claims $\hat{a}_i$ in (4.10) align with the diagonal line in the scatter plot against observed annual claims $a_i = y_i/n_i$. Results confirm that the three best PM DLNNs can robustly and accurately represent the underlying data dynamics.

5.3 Future research

For loss reserving of run-off triangle, future research can be directed to incorporate differential development year effect in later development years $j \geq t_\beta$ for some $t_\beta < t$. For example, one can set

$$\beta_j = \beta_0 \text{ if } j \geq t_\beta$$

to reduce the number of parameters in the ANOVA model as the development year effect often stabilizes in a later development year. For the ANCOVA model, one can set

$$\beta_j = \begin{cases} \beta_1 j + \beta_2 j^2 & \text{if } j < t_\beta \\ \beta_0 j & \text{if } j \geq t_\beta \end{cases}$$

as the trend approaches linear and stabilizes in later development years. These models are especially suitable for larger run-off triangles. To enrich the model capacity and forecasting accuracy, one can advocate to include a bilinear persistent term for the development year which is an interaction term of the weak and strong persistent effects. Tan et al. (2021) applied the bilinear CARR model to model the financial time series of volatilities. Taking the exponentially transformed order (1,1) mean function in (2.15) as an example, the mean function can be extended to

$$\begin{aligned} &\mu_{i,j}|\boldsymbol{y}_{i,1:(j-1)}, \boldsymbol{\mu}_{i,1:(j-1)} \\ &= \exp\Big[\mu_0 + \alpha_i + \beta_j + \delta_{i+j-1} + \gamma_{11} \log(y_{i,j-l}) + \gamma_{21} \log(\mu_{i,j-l}) + \gamma_{31} \log(y_{i,j-l}) \times \log(\mu_{i,j-l})\Big]. \end{aligned} \quad (5.1)$$

For the claim prediction model with telematics data, an intriguing avenue for exploration involves extending the number of driver groups G for the PM model to beyond two, capturing other distinct driving behaviors. The model defined in terms of log-likelihood function can be found in (3.2). Previous studies in clinical and criminology (Chan et al., 1998; Brame et al., 2006) demonstrated the applicability and superior performance of mixture models with three or more components. This extension to telematics data could closely examine different driving styles using both lasso regressions and DLNNs and, hence, enhance their predictive capabilities. For the case with $G = 3$, the model can potentially identify a group with excessive or structural zero claims, another group of low claims, and lastly, a group of high claims, combining the

ZIP with PM models to capture the heterogeneity in the data and enhance prediction accuracy. Hence, the extended multi-component mixture models remain a promising avenue, although the identification and interpretation of these groups pose additional challenges. These challenges stem from setting up proper constraints to differentiate driver groups in order to avoid label switching (Stephens, 2000; Yao, 2012) in some iterative estimating procedures such as MCMC and SGD. The lack of prior exploration in this direction makes it promising to derive unprecedented insights into the driving behavior dynamics from the telematics data.

Future research should address the inherent challenges and intricacies of optimizing network architecture, particularly when dealing with complex count models such as the multi-component PM model. The difficulties involve striking a delicate balance between model efficiency and computational resources, considering time constraints. To overcome these challenges, exploring more advanced optimization strategies is imperative. Researchers can investigate innovative approaches to efficiently tailor DLNN architectures to address the unique characteristics of complex count data, focusing on methods that enhance computational efficiency while maintaining or improving the model's capacity to generalize and discern intricate patterns.

Collaborating with telematics data providers is a common practice in the insurance sector. These providers process raw telematics data, augmenting it with external sources like road maps, and ultimately deliver structured aggregated telematics information. To enrich the current analysis using PM DLNNs, researchers can integrate additional information in a trip, including specific driving contexts such as road type and traffic condition (Winlaw et al., 2019; Meng et al., 2022) as well as weather (Winlaw et al., 2019). It is advisable to thoroughly examine this information in the networks as it offers a deeper understanding of driving behavior and crash risk, especially as risky drivers may demonstrate more aggressive driving in adverse weather conditions (Winlaw et al., 2019). Incorporating trip and weather information means extending the feature input format from an aggregated cross-sectional vector to a matrix with time steps for each driver (Verbelen, 2017). The LSTM and one-dimensional CNN (1D-CNN) have emerged as widely adopted methods for dealing with time series

features that enable deeper analyses of driver behavior classifications, such as safe, aggressive, distracted, drowsy, and drunk driving (Shahverdy et al., 2020). Although CNNs are conventionally associated with image data, the application of 1D-CNN has proven to yield superior results in both classification and trip behavior data, as evidenced by studies conducted by Erramaline et al. (2022) and Escottá et al. (2022). Furthermore, considerations for future directions, as suggested by Meng et al. (2022), include exploring scenarios where claim counts lead to overdispersion in the context of using dense neural networks. In such instances, zero-modified Poisson models, such as the zero-truncated Poisson model or the zero-inflated Poisson model, and the negative binomial model present a promising avenue for exploration.

Lastly, uncertainty quantification has been a growing research area in NNs. Basic NNs provide estimates of target variables, but unlike statistical models, uncertainties of these estimates are unknown as they are not described in the MSE loss function. Adopting the NLL loss function provides a data distribution assumption to enable uncertainty quantification. Although such uncertainty is not utilized in premium calculation using the UBI experience-rating method in (3.5), it is vital for loss reserving as insurance companies reserve losses at higher quantile levels, such as the 75 percentile. To provide such a quantile estimate for driver i, say, one can simulate the distribution of y_i from the fitted model in (3.3) and take the 75 percentile of the simulated set of claim counts as the potential number of claims.

Bibliography

AbuJarad, M. H. and Khan, A. A. (2018). Exponential model: a Bayesian study with Stan. *Int J Recent Sci Res*, 9(8):28495–28506.

Adam, F. F. (2018). Claim reserving estimation by using the chain ladder method. *KnE Social Sciences*, pages 1192–1204.

Aiuppa, T. A. (1988). Evaluation of Pearson curves as an approximation of the maximum probable annual aggregate loss. *The Journal of Risk and Insurance*, 55(3):425–441.

Amini, A., Schwarting, W., Soleimany, A., and Rus, D. (2020). Deep evidential regression. *Advances in Neural Information Processing Systems*, 33:14927–14937.

Anderson, D. R. (1971). Effects of under and over-evaluations in loss reserves. *Journal of Risk and Insurance*, pages 585–600.

Annis, J., Miller, B. J., and Palmeri, T. J. (2017). Bayesian inference with Stan: A tutorial on adding custom distributions. *Behavior research methods*, 49(3):863–886.

Antonio, K. and Plat, R. (2014). Micro-level stochastic loss reserving for general insurance. *Scandinavian Actuarial Journal*, 2014(7):649–669.

Arumugam, S. and Bhargavi, R. (2019). A survey on driving behavior analysis in usage based insurance using big data. *Journal of Big Data*, 6:1–21.

Atherino, R., Pizzinga, A., and Fernandes, C. (2010). A row-wise stacking of the run-off triangle: State space alternatives for IBNR reserve prediction. *ASTIN Bulletin: The Journal of the IAA*, 40(2):917–946.

Bakar, S. A., Hamzah, N. A., Maghsoudi, M., and Nadarajah, S. (2015). Modeling loss data using composite models. *Insurance: Mathematics and Economics*, 61:146–154.

Banerjee, P., Garai, B., Mallick, H., Chowdhury, S., and Chatterjee, S. (2018). A note on the adaptive lasso for zero-inflated Poisson regression. *Journal of Probability and Statistics*, 2018.

Barry, L. and Charpentier, A. (2020). Personalization as a promise: Can big data change the practice of insurance? *Big Data & Society*, 7(1):2053951720935143.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade: Second Edition*, pages 437–478.

Berger, J. (2006). The case for objective Bayesian analysis. *Bayesian analysis*, 1(3):385–402.

Berry, M. J. and Linoff, G. S. (2004). *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons.

Bhattacharya, S. and McNicholas, P. D. (2014). An adaptive lasso-penalized BIC. *arXiv preprint arXiv:1406.1332*.

Björkwall, S., Hössjer, O., Ohlsson, E., and Verrall, R. (2011). A generalized linear model with smoothing effects for claims reserving. *Insurance: Mathematics and Economics*, 49(1):27–37.

Boger, Z. and Guterman, H. (1997). Knowledge extraction from artificial neural network models. In *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 4, pages 3030–3035 vol.4.

Bolderdijk, J. W., Knockaert, J., Steg, E., and Verhoef, E. T. (2011). Effects of pay-as-you-drive vehicle insurance on young drivers' speed choice: Results of a dutch field experiment. *Accident Analysis & Prevention*, 43(3):1181–1186.

Brame, R., Nagin, D. S., and Wasserman, L. (2006). Exploring some analytical characteristics of finite mixture models. *Journal of Quantitative Criminology*, 22:31–59.

Buyrukoglu, G., Buyrukoglu, S., and Topalcengiz, Z. (2021). Comparing regression models with count data to artificial neural network and ensemble models for prediction of generic escherichia coli population in agricultural ponds based on weather station measurements. *Microbial Risk Analysis*, 19.

Cameron, A. and Trivedi, P. K. (1990). Regression-based tests for overdispersion in the Poisson model. *Journal of Econometrics*, 46(3):347–364.

Chan, J. (2015). Predicting loss reserves using quantile regression. *Journal of Data Science*, 13(1):127–155.

Chan, J., Choy, S., and Makov, U. (2008). Robust Bayesian analysis of loss reserves data using the generalized-t distribution. *ASTIN Bulletin: The Journal of the IAA*, 38(1):207–230.

Chan, J., Choy, S., Makov, U., and Landsman, Z. (2018). Modeling insurance losses using contaminated generalized beta type-II distribution. *ASTIN Bulletin: The Journal of the IAA*, 48(2):871–904.

Chan, J. S. K., Kuk, A. Y. C., Bell, J., and Gilchrist, C. M. (1998). The analysis of methadone clinic data using marginal and conditional logistic models with mixture or random effects. *Australian & New Zealand Journal of Statistics*, 40.

Chang, I. and Kim, S. W. (2012). Modelling for identifying accident-prone spots: Bayesian approach with a poisson mixture model. *KSCE Journal of Civil Engineering*, 16:441–449.

Chassagnon, A., Chiappori, P.-A., et al. (1997). Insurance under moral hazard and adverse selection: the case of pure competition. *Delta-CREST Document*.

Chou, R. Y., Chou, H., and Liu, N. (2015). Range volatility: A review of models and empirical studies. *Handbook of financial econometrics and statistics*, pages 2029–2050.

Costa, L. and Pizzinga, A. (2020). State-space models for predicting IBNR reserve in row-wise ordered run-off triangles: Calendar year IBNR reserves & tail effects. *Journal of Forecasting*, 39(3):438–448.

Costa, L., Pizzinga, A., and Atherino, R. (2016). Modeling and predicting IBNR reserve: extended chain ladder and heteroscedastic regression analysis. *Journal of Applied Statistics*, 43(5):847–870.

Courtois, É., Tubert-Bitter, P., and Ahmed, I. (2021). New adaptive lasso approaches for variable selection in automated pharmacovigilance signal detection. *BMC medical research methodology*, 21(1):1–17.

Cummins, D., Lewis, C., and Phillips, R. (1999). Pricing excess-of-loss reinsurance contracts against cat as trophic loss. In *The financing of catastrophe risk*, pages 93–148. University of Chicago Press.

Cummins, J., Mcdonald, J., and Merrill, C. (2007). Risky loss distributions and modeling the loss reserve pay-out tail. *Review of Applied Economics*, 3:1–23.

Cummins, J. D., Dionne, G., McDonald, J. B., and Pritchett, B. M. (1990). Applications of the GB2 family of distributions in modeling insurance loss processes. *Insurance: Mathematics and Economics*, 9(4):257–272.

De Jong, P. (2006). Forecasting runoff triangles. *North American Actuarial Journal*, 10(2):28–38.

De Jong, P., Heller, G. Z., et al. (2008). Generalized linear models for insurance data. *Cambridge Books*.

De Jong, P. and Zehnwirth, B. (1983). Claims reserving, state-space models and the Kalman filter. *Journal of the Institute of Actuaries (1886-1994)*, 110(1):157–181.

De Myttenaere, A., Golden, B., Le Grand, B., and Rossi, F. (2016). Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48.

Dean, C. G. (1997). An introduction to credibility. In *Casualty actuary forum: Casualty actuarial society*.

Dong, A. X. and Chan, J. (2013). Bayesian analysis of loss reserving using dynamic models with generalized beta distribution. *Insurance: Mathematics and Economics*, 53(2):355–365.

Dong, W., Li, J., Yao, R., Li, C., Yuan, T., and Wang, L. (2016). Characterizing driving styles with deep learning. *arXiv preprint arXiv:1607.03611*.

Drori, I. (2022). *The Science of Deep Learning*. Cambridge University Press.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222.

Eling, M. and Kraft, M. (2020). The impact of telematics on the insurability of risks. *The Journal of Risk Finance*, 21(2):77–109.

Ellison, A. B., Bliemer, M. C., and Greaves, S. P. (2015). Evaluating changes in driver behavior: A risk profiling approach. *Accident Analysis & Prevention*, 75:298–309.

Elvik, R. (2014). Rewarding safe and environmentally sustainable driving: systematic review of trials. *Transportation Research Record*, 2465(1):1–7.

Embrechts, P., Klüppelberg, C., and Mikosch, T. (2013). *Modeling extremal events: for insurance and finance*, volume 33. Springer Science & Business Media.

England, P. D. and Verrall, R. J. (2001). A flexible framework for stochastic claims reserving. In *Proceedings of the Casualty Actuarial Society*, volume 88, pages 1–38.

Erramaline, A., Badard, T., Côté, M.-P., Duchesne, T., and Mercier, O. (2022). Identification of road network intersection types from vehicle telemetry data using a convolutional neural network. *ISPRS International Journal of Geo-Information*, 11(9):475.

Escottá, Á. T., Beccaro, W., and Ramírez, M. A. (2022). Evaluation of 1D and 2D deep convolutional neural networks for driving event recognition. *Sensors*, 22(11):4226.

Fallah, N., Gu, H., Mohammad, K., Seyyedsalehi, S., Nourijelyani, K., and Eshraghian, M. (2009). Nonlinear Poisson regression using neural networks: A simulation study. *Neural Computing and Applications*, 18:939–943.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874.

Fernandes, M. and Grammig, J. (2006). A family of autoregressive conditional duration models. *Journal of Econometrics*, 130(1):1–23.

Gao, G., Meng, S., and Wüthrich, M. V. (2019a). Claims frequency modeling using telematics car driving data. *Scandinavian Actuarial Journal*, 2019(2):143–162.

Gao, G. and Wüthrich, M. V. (2018). Feature extraction from telematics car driving heatmaps. *European Actuarial Journal*, 8(2):383–406.

Gao, G., Wüthrich, M. V., and Yang, H. (2019b). Evaluation of driving risk at different speeds. *Insurance: Mathematics and Economics*, 88:108–119.

Garbin, C., Zhu, X., and Marques, O. (2020). Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79:12777–12815.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Guillen, M., Nielsen, J. P., and Pérez-Marín, A. M. (2021). Near-miss telematics in motor insurance. *Journal of Risk and Insurance*.

Guo, J., Liu, Y., Zhang, L., and Wang, Y. (2018). Driving behavior style study with a hybrid deep learning framework based on GPS data. *Sustainability*, 10(7):2351.

Hanafy, M. and Ming, R. (2021). Machine learning approaches for auto insurance big data. *Risks*, 9(2):42.

Hecht-Nielsen, R. (1987). Kolmogorov"s mapping neural network existence theorem.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

Huang, Y. and Meng, S. (2019). Automobile insurance classification ratemaking based on telematics driving data. *Decision Support Systems*, 127:113156.

Hurley, R., Evans, P., and Menon, A. (2015). Insurance disrupted: General insurance in a connected world. *London: The Creative Studio, Deloitte*.

Husnjak, S., Peraković, D., Forenbacher, I., and Mumdziev, M. (2015). Telematics system in usage based motor insurance. *Procedia Engineering*, 100:816–825.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr.

Jeong, H. and Valdez, E. A. (2018). Ratemaking application of Bayesian lasso with conjugate hyperprior. *Available at SSRN 3251623*.

Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., and Xing, E. P. (2018). Neural architecture search with Bayesian optimization and optimal transport. *Advances in neural information processing systems*, 31.

Kang, S. Y. (1991). *An investigation of the use of feedforward neural networks for forecasting*. PhD thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2022-11-23.

Kantor, S. and Stárek, T. (2014). Design of algorithms for payment telematics systems evaluating driver's driving style. *Transactions on Transport Sciences*, 7(1):9.

Karsoliya, S. and Azad, M. A. K. (2012). Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture.

Kim, J.-M., Kim, J.-M., and Ha, I. (2022). Application of deep learning and neural network to speeding ticket and insurance claim count data. *Axioms*, 11.

Klein, A., Falkner, S., Bartels, S., Hennig, P., and Hutter, F. (2017). Fast Bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial intelligence and statistics*, pages 528–536. PMLR.

Kremer, E. (1984). A class of autoregressive models for predicting the final claims amount. *Insurance: Mathematics and Economics*, 3(2):111–119.

Kuang, D. and Nielsen, B. (2020). Generalized log-normal chain-ladder. *Scandinavian Actuarial Journal*, 2020(6):553–576.

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

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lee, M. D. and Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge university press.

Leisch, F. (2004). Flexmix: A general framework for finite mixture models and latent class regression in R. *Journal of Statistical Software*, 11(8):1–18.

Li, M., Soltanolkotabi, M., and Oymak, S. (2020). Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pages 4313–4324. PMLR.

Liew, S. S., Khalil-Hani, M., and Bakhteri, R. (2016). Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing*, 216:718–734.

Litman, T. (2007). Distance-based vehicle insurance feasibility, costs and benefits. *Victoria*, 11.

Litman, T. (2011). Distance-based vehicle insurance feasibility, costs, and benefits. victoria transport policy institute. *Victoria, BC, V8V 3R7, Canada*.

Liu, Y., Starzyk, J. A., and Zhu, Z. (2007). Optimizing number of hidden neurons in neural networks. *EeC*, 1(1):6.

Mack, T. (1993). Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin: The Journal of the IAA*, 23(2):213–225.

Mack, T. and Venter, G. (2000). A comparison of stochastic models that reproduce chain ladder reserve estimates. *Insurance: mathematics and economics*, 26(1):101–107.

Makov, U. and Weiss, J. (2016). Predictive modeling for usage-based auto insurance. *Predictive Modeling Applications in Actuarial Science*, 2:290.

Masters, D. and Luschi, C. (2018). Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*.

Masum, M., Shahriar, H., Haddad, H., Faruk, M. J. H., Valero, M., Khan, M. A., Rahman, M. A., Adnan, M. I., Cuzzocrea, A., and Wu, F. (2021). Bayesian hyperparameter optimization for deep neural network-based network intrusion detection. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 5413–5419. IEEE.

McKenzie, J. (2011). Mean absolute percentage error and bias in economic forecasting. *Economics Letters*, 113(3):259–262.

Meinshausen, N. and Bühlmann, P. (2006). Variable selection and high-dimensional graphs with the lasso. *Ann Stat*, 34:1436–1462.

Meng, S., Wang, H., Shi, Y., and Gao, G. (2022). Improving automobile insurance claims frequency prediction with telematics car driving data. *ASTIN Bulletin: The Journal of the IAA*, 52(2):363–391.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Meyers, G. (2015). Stochastic loss reserving with Bayesian MCMC models. *Geophysical Monograph Series*, 1:1.

Mishra, S., Yamasaki, T., and Imaizumi, H. (2019). Improving image classifiers for small datasets by learning rate adaptations. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE.

Morteza, K. and Ahmadabadi, A. (2010). Some properties of generalized gamma distribution. *Mathematical Sciences Quarterly Journal*, 4:9–28.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nadarajah, S. and Haghighi, F. (2011). An extension of the exponential distribution. *Statistics*, 45:543–558.

Neal, R. M. (1994). An improved acceptance procedure for the hybrid Monte Carlo algorithm. *Journal of Computational Physics*, 111(1):194–203.

Ntzoufras, I. and Dellaportas, P. (2002). Bayesian modeling of outstanding liabilities incorporating claim count uncertainty. *North American Actuarial Journal*, 6(1):113–125.

Osafune, T., Takahashi, T., Kiyama, N., Sobue, T., Yamaguchi, H., and Higashino, T. (2017). Analysis of accident risks from driving behaviors. *International journal of intelligent transportation systems research*, 15(3):192–202.

Paefgen, J., Staake, T., and Fleisch, E. (2014). Multivariate exposure modeling of accident risk: Insights from pay-as-you-drive insurance data. *Transportation Research Part A: Policy and Practice*, 61:27–40.

Paefgen, J., Staake, T., and Thiesse, F. (2013). Evaluation and aggregation of pay-as-you-drive insurance rate factors: A classification analysis approach. *Decision Support Systems*, 56:192–201.

Park, B.-J. and Lord, D. (2009). Application of finite mixture models for vehicle crash data analysis. *Accident Analysis & Prevention*, 41(4):683–691.

Park, T. and Casella, G. (2008). The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.

Paulson, A. S. and Faris, N. (1985). A practical approach to measuring the distribution of total annual claims. In *Strategic Planning and Modeling in Property-Liability Insurance*, pages 205–223. Springer.

Peters, G. W., Wüthrich, M. V., and Shevchenko, P. V. (2010). Chain ladder method: Bayesian bootstrap versus classical bootstrap. *Insurance: Mathematics and Economics*, 47(1):36–51.

Pigeon, M., Antonio, K., and Denuit, M. (2014). Individual loss reserving using paid–incurred data. *Insurance: Mathematics and Economics*, 58:121–131.

Pinals, L., Kerin, A., Craig Van Alsten, F., Sharp, R., and Madden, S. (2022). Motivating safer driving with telematics.

Qu, Z., Yuan, S., Chi, R., Chang, L., and Zhao, L. (2019). Genetic optimization method of pantograph and catenary comprehensive monitor status prediction model based on adadelta deep neural network. *IEEE Access*, PP:1–1.

Ramlau-Hansen, H. (1988). A solvency study in non-life insurance. *Scandinavian Actuarial Journal*, 1988(1-3):35–59.

Reagan, I., Cicchino, J., and Teoh, E. (2023). The utility of telematics data for estimating the prevalence of driver handheld cellphone use.

Renshaw, A. E. (1989). Chain ladder and interactive modeling (Claims reserving and GLIM). *Journal of the Institute of Actuaries (1886-1994)*, 116(3):559–587.

Renshaw, A. E. and Verrall, R. J. (1998). A stochastic model underlying the chain-ladder technique. *British Actuarial Journal*, 4(4):903–923.

Sagberg, F., Selpi, Bianchi Piccinini, G. F., and Engström, J. (2015). A review of research on driving styles and road safety. *Human factors*, 57(7):1248–1275.

Sakthivel, K. and Rajitha, C. (2017). A comparative study of zero-inflated, hurdle models with artificial neural network in claim count modeling. *International Journal of Statistics and Systems*, 12(2):265–276.

Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization? *Advances in neural information processing systems*, 31.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Shahverdy, M., Fathy, M., Berangi, R., and Sabokrou, M. (2020). Driver behavior detection and classification using deep convolutional neural networks. *Expert Systems with Applications*, 149:113240.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55.

Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316.

Simoncini, M., Taccari, L., Sambo, F., Bravi, L., Salti, S., and Lori, A. (2018). Vehicle classification from low-frequency GPS data with recurrent neural networks. *Transportation Research Part C: Emerging Technologies*, 91:176–191.

Smith, K. A., Willis, R. J., and Brooks, M. (2000). An analysis of customer retention and insurance claim patterns using data mining: A case study. *Journal of the operational research society*, 51(5):532–541.

Soleymanian, M., Weinberg, C. B., and Zhu, T. (2019). Sensor data and behavioral tracking: Does usage-based auto insurance benefit drivers? *Marketing Science*, 38(1):21–43.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Städler, N., Bühlmann, P., and Van De Geer, S. (2010). L1-penalization for mixture regression models. *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research*, 19:209–256.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809.

Sze, V., Chen, Y.-H., Yang, T.-J., and Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329.

Tan, S. K., Chan, J. S. K., and Ng, K. H. (2021). Modelling and forecasting stock volatility and return: A new approach based on quantile rogers–satchell volatility measure with asymmetric bilinear carr model. *Studies in Nonlinear Dynamics & Econometrics*, 26(3):437–474.

Tang, Y., Xiang, L., and Zhu, Z. (2014). Risk factor selection in rate making: EM adaptive LASSO for zero-inflated Poisson regression models. *Risk Analysis*, 34(6):1112–1127.

Tang, Z. and Fishwick, P. (1993). Feedforward neural nets as models for time series forecasting. *INFORMS Journal on Computing*, 5:374–385.

Taylor, G. (2012). *Loss reserving: an actuarial perspective*, volume 21. Springer Science & Business Media.

Taylor, G. and McGuire, G. (2016). Stochastic loss reserving using generalized linear models. *Arlington: Casualty Actuarial Society*.

Team, S. (2016). Rstan: the interface to stan, version 2.14. 1.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Toledo, T. and Lotan, T. (2006). In-vehicle data recorder for evaluation of driving behavior and safety. *Transportation research record*, 1953(1):112–119.

Tselentis, D. I., Yannis, G., and Vlahogianni, E. I. (2016). Innovative insurance schemes: pay as/how you drive. *Transportation Research Procedia*, 14:362–371.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: an improved R for assessing convergence of MCMC (with discussion). *Bayesian analysis*, 16(2):667–718.

Venter, G. (2007). Generalized linear models beyond the exponential family with loss reserve applications. *ASTIN Bulletin: The Journal of the IAA*, 37(2):345–364.

Venter, G. (2018). Loss reserving using estimation methods designed for error reduction. *Available at SSRN 3096178*.

Venter, G., Gutkovich, R., and Gao, Q. (2017). Parameter reduction in actuarial triangle models. *Variance, Forthcoming, UNSW Business School Research Paper Forthcoming*.

Venter, G. and Şahın, Ş. (2018). Parsimonious parameterization of age-period-cohort models by Bayesian shrinkage. *ASTIN Bulletin: The Journal of the IAA*, 48(1):89–110.

Verbelen, R. (2017). *Data analytics for insurance loss modeling, telematics pricing and claims reserving*. PhD thesis. KU Leuven.

Verbelen, R., Antonio, K., and Claeskens, G. (2018). Unravelling the predictive power of telematics data in car insurance pricing. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(5):1275–1304.

Verrall, R. (1996). Claims reserving and generalized additive models. *Insurance: Mathematics and Economics*, 19(1):31–43.

Verrall, R. (2000). An investigation into stochastic claims reserving models and the chain-ladder technique. *Insurance: Mathematics and Economics*, 26:91–99.

Weerasinghe, K. and Wijegunasekara, M. (2016). A comparative study of data mining algorithms in the prediction of auto insurance claims. *European International Journal of Science and Technology*, 5(1):47–54.

Winlaw, M., Steiner, S. H., MacKay, R. J., and Hilal, A. R. (2019). Using telematics data to find risky driver behavior. *Accident Analysis & Prevention*, 131:131–136.

Wong, F. (1991). Time series forecasting using backpropagation neural networks. *Neurocomputing*, 2(4):147–159.

Wouters, P. I. and Bos, J. M. (2000). Traffic accident reduction by monitoring driver behavior with in-car data recorders. *Accident Analysis & Prevention*, 32(5):643–650.

Wüthrich, M. V. (2017). Covariate selection from telematics car driving data. *European Actuarial Journal*, 7(1):89–108.

Wüthrich, M. V. and Merz, M. (2008). *Stochastic claims reserving methods in insurance*. John Wiley & Sons.

Wüthrich, M. V. and Merz, M. (2019). Yes, we CANN! *ASTIN Bulletin: The Journal of the IAA*, 49(1):1–3.

Yao, W. (2012). Model based labeling for mixture models. *Statistics and Computing*, 22:337–347.

Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing.

You, K., Long, M., Wang, J., and Jordan, M. I. (2019). How does learning rate decay help modern neural networks?

Yunos, Z., Ali, A., Shamsuddin, S. M., Noriszura, I., and Sallehuddin, R. (2016). Predictive modeling for motor insurance claims using artificial neural networks. *International Journal of Advances in Soft Computing and its Applications*, 8.

Zantema, J., van Amelsfort, D. H., Bliemer, M. C., and Bovy, P. H. (2008). Pay-as-you-drive strategies: case study of safety and accessibility effects. *Transportation Research Record*, 2078(1):8–16.

Zeileis, A., Kleiber, C., and Jackman, S. (2008). Regression models for count data in R. *Journal of statistical software*, 27(8):1–25.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, C. and Browne, M. J. (2013). Loss reserve errors, income smoothing and firm risk of property and casualty insurance companies. In *Annual Meeting of the American Risk and Insurance Association, Working Pa*, pages 1–55.

Zhang, P., Patuwo, E., and Hu, M. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14:35–62.

Zhang, Z. (2018). Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pages 1–2. IEEE.

Zhang, Z. J., Milovanovic, D. J., Tomita, M., and Zicarreli, D. J. (2018). Using generalized linear models to develop loss triangles in reserving. *Actuarial Review*, pages 37–40.

Ziakopoulos, A., Petraki, V., Kontaxi, A., and Yannis, G. (2022). The transformation of the insurance industry and road safety by driver safety behavior telematics. *Case Studies on Transport Policy*, 10(4):2271–2279.

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

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.

Zou, H. and Zhang, H. H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Annals of statistics*, 37(4):1733.