Machine Learning Techniques for Detecting Hierarchical Interactions in Insurance Claims Models

Sandra Maria Nawar

A Thesis

for The Department of

Mathematics and Statistics

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Science (Mathematics) at

Concordia University

Montreal, Quebec, Canada

July 2016

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Sandra Maria Nawar**

Entitled: **Machine Learning Techniques for Detecting Hierarchical Interactions in Insurance Claims Models**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Mathematics)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Examiner

Dr. Y. P. Chaubey

_____ Examiner

Dr. F. Godin

_____ Thesis Supervisor

Dr. J. Garrido

_____ Thesis Co-Supervisor

Dr. M. Mailhot

Approved by _____

Chair of Department or Graduate Program Director

_____

Dean of Faculty

Date _____

Abstract

Machine Learning Techniques for Detecting Hierarchical Interactions in Insurance Claims Models

by Sandra Maria Nawar

This thesis presents an intuitive way to do predictive modeling in actuarial science. Generalized Linear Models (GLMs) are the standard tool for predictive modeling in the actuarial literature and in actuarial practice, yet GLMs can be quite restrictive. The aim of this work is to model claims and to propose solutions to current actuarial problems such as high variability in large data-sets, variable selection, overfitting, dealing with highly correlated variables and detecting non-linear effects such as interactions.

Regularization techniques are crucial for modeling big data, which means dealing with high-dimensionality, sometimes noisy data that often contains many irrelevant predictors. Penalized regression is a set of regression techniques that impose a constraint/penalty on the regression coefficients and can be used as a powerful variable selection tool as well. They are a generalization of GLMs and include techniques such as Ridge regression, lasso, group-lasso and Elastic Net. The proposed approach is a hierarchical group-lasso-type model that can efficiently handle variable selection and interaction detection between variables while enforcing strong hierarchy. This is achieved by imposing a penalty on the coefficients at the individual and group level. By optimizing the penalized objective function the model performs variable selection and estimation. Additionally, the model automatically detects interactions which is another important factor to achieve a high predictive power. For those purposes the group-lasso method is investigated for the Poisson and gamma distributions to perform frequency-severity modeling.

Acknowledgments

I cannot be more grateful to my supervisor, Dr. José Garrido, for introducing me to predictive modeling and actuarial research. I am very thankful for providing me with the opportunity to work with him for the past two years, providing me with all the possible resources a student can ask for to succeed. His mentorship, knowledge, encouragement, patience and continuous help have been invaluable to complete my thesis and my experience as a Masters student.

I would also like to thank Dr. Melina Mailhot for Co-supervising me and guiding me during my studies even when she was away.

I would also like to thank Dr. Yogendra Chaubey and Dr. Frédéric Godin for their comments and for reviewing this work. Furthermore, I would like to mention all my colleagues at Concordia and at my internships for all the support.

I would like to thank the Department of Mathematics and Statistics at Concordia University and the MITACS organization for financial support.

The biggest thank you goes to my parents, my grandmother and siblings for their unconditional support and confidence in me. I could not have done it without them. Words cannot express my gratitude to my friends for their care and support. Lastly, I would like to thank my boyfriend who stood by my side during my education to motivate me and encourage me throughout.

To my parents: Nabil and Nermine.

Contents

List of Figures

List of Tables

Introduction

The Property and Casualty insurance industry has widely embraced predictive modeling as a tool for competitiveness in the market. These models are being deployed for pricing, risk segmentation as well as reserving work. With increasing availability of data, building interpretable predictive models with high prediction accuracy remains a challenge. Predictive models help actuaries and insurance underwriters better select, price risks and generate more accurate premium predictions, thus helping solve the adverse selection issue.

Generalized linear models (GLMs) have been the standard tool for predictive modeling in the industry. GLMs have a major limitation in terms of finding non-linear model structures. Here we combine machine learning tools and regularization to generalized linear models, which are commonly accepted building blocks of insurance pricing and scoring models. Regularization plays an important role in modeling data with a large number of covariates where automatic variable selection is required. Advances in technology have made it possible to capture and store large amounts of data and while the number of observations has increased the number of variables is increasing as well. Sometimes, it actually exceeds it, which is known as the "$p > n$" problem. A popular approaches in machine learning is to use regularization, such as adding ℓ_2 penalty of the form $\|\beta\|_2^2$ or a ℓ_1 penalty of the form $\|\beta\|_1$ to the model coefficients.

In particular, we focus on the class of group-lasso hierarchical interaction models for insurance data. To fit frequency-severity models in this framework a regularized Poisson and gamma model is proposed. The purpose is to find the most predictive subset of factors and the corresponding hierarchical interactions. We use the lasso technique which imposes

the ℓ_1 penalty for model selection and reduce the variability in estimated parameters. The estimation is carried out with gradient descent methods. A recurring concept that is relevant in actuarial and statistical modeling is parsimony. Models that over-fit training data are a concern and the basic underlying belief is that parsimonious models with fewer terms provide better predictions. For big data, feature selections can only be achieved automatically through regularized models, such as least adaptive shrinkage and selection operator (lasso) and ridge regression (Bishop, 2007). It turns out, that regularization methods form a fruitful area of research in statistical learning. The idea is simply to add a "penalty" term to the model to achieve a desirable behavior. The main result that we demonstrate in detail is the reduction in the variability of estimated model parameters and improved the predictive ability.

Property and Casualty insurance is a complex and dynamic business due to the short duration of the insurance coverage. The real price of the product is usually unknown at the time of sale. It is therefore practically impossible for actuarial models to be "comprehensive", in the sense that it should include all relevant variables that affect the number and size of claims. A solution to that is to use statistical shrinkage methods for feature selection in order to identify those with the most predictive power. The need for predictive models emerges from the fact that the expected loss is highly dependent on the characteristics of an individual policy, such as age and motor vehicle record points of the policyholder, population density in the policyholder's residential area, age and vehicle type of the driver.

In this thesis, a new model is proposed that combines the simplicity and interpretability of generalized linear models with the predictive power of machine learning algorithms. This would aid in ratemaking purposes for data sets with large number of covariates in a Property and Casualty insurance portfolio. Predictive modeling - and in particular claim predictions are essential for ratemaking and reserving purposes. For classification ratemaking and predictive modeling applications, actuaries add hierarchal structures to their generalized linear models in order to accommodate variations in model variables.

Linear models remain popular due to their simplicity and interpretability however they

suffer severe limitations. Significance and statistical estimation is easily trackable through the well-developed theories of Analysis of Variance (ANOVA), P-tests and goodness of fit metrics. However, limitations of linear models include generating extreme values for the predicted value of the response Y. It goes back to the fact that the assumption of linear relationship between the response and covariates does not hold always in reality. In addition, linear models include only main effects so interactions have to be added manually, which is not usually an efficient way. Without model selection, linear models are subject to overfitting when a large number of covariates are present. And they are not appropriate for case-finding and classification that requires partitioning and addressing subgroups.

Another important issue in predictive models is the the trade-off between prediction accuracy and model interpretability. For most models, predictive accuracy comes at the expense of interpretability. For example, boosting methods which are fully non-linear can lead to complicated estimates that make it difficult to understand how any individual predictor is associated with the response. However, predictive accuracy is mostly high.

The lasso technique relies on linear models but uses an alternative fitting procedure for estimating coefficients. This model is more restrictive in estimating the coefficients and it sets a number of them to exactly zero. Consequently, the lasso is a less flexible approach than linear regression, however more interpretable than it, because in the final model the response variable only depends on a smaller subset of predictors, those with nonzero coefficients. Depending on whether interpretability is a priority, there are advantages to using simple and relatively inflexible statistical learning methods. However, when prediction is a priority and the interpretability of the predictive model is simply not of interest it is suggested to use highly flexible models.

The goals of this thesis are twofold: the first is to review the concept of regularization and its applicability in actuarial modeling. The second goal, is to illustrate how to automatically select features that "learn" hierarchical interactions in the group-lasso framework for claims modeling. Chapters 1 and 2 review the background material for generalized linear models and regularization methods, explaining the main concepts of these subjects as they are build-

ing blocks for the model derivation. Chapter 3 explains the use of hierarchical modeling and more specifically hierarchical interactions while deriving all the necessary models. Chapter 4 contains the main idea presented in this thesis; it gives the derivation of the regularized models for frequency and severity modeling. The thesis concludes with the results obtained from the proposed model, comparing it to results obtained from a generalized linear model and gradient boosting model. In my future actuarial research and predictive modeling endeavor, I wish to explore other predictive models in the linear and non-linear frameworks and compare them to traditional methods in actuarial science to improve the status quo. This will be the subject of future work.

"In God we trust, all else bring data"

by Edward Deeming

Chapter 1

Generalized Linear Models

1.1 Introduction

Predictive modeling involves the use of historical data to forecast future events by capturing relationships between explanatory variables and response variables. The standard predictive modeling technique in Property and Casualty insurance is generalized linear models (GLMs). The family of GLMs is an extension of the linear regression model (McCullagh and Nelder, 1989) that transforms the mean response by a chosen link function. The response variable Y can then be linked to a linear function of predictor variables with a non-linear link function.

The log-link is the most popular for insurance data, where the linear predictor gets exponentiated to ensure that premiums are positive. It also preserves the multiplicative structure of the variable relativities. The relationship between the response and the covariates is no longer directly linear. Simply using a non-linear link function allows the linearity of the exponentiated term to be preserved.

The reason why linearity is so desirable lies in its simplicity to model and to interpret. On the other hand, by transforming the mean response the model can no longer be fitted using ordinary least square (OLS). Other methods such as maximum likelihood estimation (MLE) and gradient descent are used. Important examples include the logistic regression for binary responses, the Poisson regression for count, log-linear models for contingency tables

and the gamma regression for continuous skewed data. Another difference between GLMs and linear models is that the variance of the response variable is not required to be constant across observations but can be made a function of the Y's expected value. Conversely, the main drawback of GLMs is the assumption that covariate effects are linearly associated with the predictor, which in reality is quite a restrictive assumption in predictive modeling. This chapter provides a summary of GLMs, discusses model assumptions, parameter estimation and model validation. All this provides a foundation of the advanced statistical and machine-learning techniques developed on later chapters.

1.2 Distributional Assumptions

Consider a model with a multivariate response vector $Y = (y_1, \ldots, y_n)$ and p-dimensional covariates arranged in a n $\times$ p matrix (design matrix) $X = (x_1, \ldots, x_p)$. Responses $y_1, \ldots, y_n$ are assumed to be independent and linearly related to the predictor variables through a non-linear link function as follows:

$$g(\mathbb{E}[Y_i|X_i = x_i]) = \sum_{j=0}^{p-1} \beta_j x_{ij}, \tag{1.1}$$

where β is the vector of coefficients to be estimated and $g(.)$ is a known non-linear link function and the covariates x are either fixed or random. The link function $g(.)$ is restricted to be differentiable and strictly monotonic. For the linear predictor we use the notation,

$$\eta_i = \sum_{j=0}^{p-1} \beta_j x_{ij}. \tag{1.2}$$

The implicit assumption for the generalized linear model family (Frees et al, 2014) is that the mean of Y_i depends on the X_i only through the link function $g(.)$. The GLM framework also assumes that the response Y comes from the exponential dispersion distribution family specified through certain characteristics discussed in detail in the next subsection. A linear model is a special case of a generalized linear model with the identity link function $g(\mu) = \mu$. The true mean can always be retrieved by taking the inverse transformation as follows:

$$\mu_i = g^{-1}(\eta_i). \tag{1.3}$$

The other important GLM assumption is that random variables Y_i are members of the exponential dispersion family of distributions. A distribution from this family is chosen appropriately to fit a model on the given response. Consequently, the relationship between variance $\mathbb{V}(Y_i)$ and expected value $\mathbb{E}(Y_i)$ will depend on the chosen distribution. The advantage of choosing a particular distribution for the model is that maximum likelihood estimation can be used to obtain the coefficients, and there are algorithms for computing the coefficients that work for all distributions in the exponential family and their corresponding canonical link functions (Frees et al., 2014).

1.3 Exponential Dispersion Family

Consider (X_i, Y_i), where $X_i \in \mathbb{R}^p$ is a vector of p predictors and a response $Y_i \in \mathbb{R}$, for observation $i = 1, \ldots, n$ to follow any distribution in the exponential dispersion family (EDF) with mean $\mu_i = \mathbb{E}(Y_i)$ and variance $v_i = \mathbb{V}(Y_i)$. The domain of Y_i varies for each distribution and could be a subset of $\mathbb{R}$. For Generalized Linear Models (GLMs) the mean μ_i is equated to the linear predictor η_i through the link function $g(.)$ such as,

$$\eta_i = g(\mu_i) = X_i^T \beta. \tag{1.4}$$

Also, the density function of Y_i at a given value y_i in the linear exponential family is expressed as follows,

$$f(y_i; \theta_i; \phi) = \exp\left[\frac{y_i\theta - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right] \tag{1.5}$$

and the likelihood function becomes,

$$L(y_i; \theta_i; \phi) = \prod_{i=1}^{n} \exp\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right] \tag{1.6}$$

where $y = (y_1 \ldots, y_n)$ are the observations and $a(.)$, $b(.)$, and $c(.)$ are functions that vary according to the particular distributions that are member of the exponential dispersion family. Note that θ_i are functions of the parameter β_j, but ϕ does not depend on the β_j. The mean and variance of the distribution are simply $\mathbb{E}(Y_i) = b'(\theta_i)$ and $\mathbb{V}(Y_i) = a(\phi)b''(\theta_i)$, where

$b'(\theta_i)$ is the first derivative with respect to θ_i and $b''(\theta_i)$ is the second derivative. Assume that the dispersion parameter ϕ is known and we are interested in finding the maximum likelihood solution for the natural parameter $\theta(\beta)$, which is a function of the coefficients β. The log-likelihood function becomes,

$$l(\theta; \phi, y) = \sum_{i=1}^{n} \left[\frac{y_i \theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \right] \tag{1.7}$$

where l denotes the log-likelihood function. The GLM coefficients are then estimated by minimizing the negative log-likelihood function (Frees, 2009) which will be explained in the next section.

For GLMs there exists two types of link functions; the canonical link and other link functions. Following the definition of the exponential dispersion, for a given distribution and thus given functions $a(.)$, $b(.)$ and $c(.)$ a given link function g is said to be canonical if it satisfies $g^{-1}(\eta) = b'(\theta) = \mu$. Thus a canonical link function allows expressing θ as a function of the mean: $\theta = g(\mu)$. Thus the canonical parameter θ in the probability density function is related to the mean $\mathbb{E}(Y) = \mu$ by the equality $\mu = b'(\theta)$. Then by inverting the function it gives the canonical parameter θ as a function of $\mu : \theta = b'^{-1}(\mu)$. If the link function $g(.)$ is chosen such that $\theta = g(\mu)$ then

$$\theta = X\beta. \tag{1.8}$$

Thus the canonical parameter θ is equal to a linear function of the predictors. The chosen link function $g(.)$ is called a canonical link function. Canonical links generate linear equations for the unknown parameters θ. A canonical link may be a good choice for modeling a particular problem, but is not necessary. Canonical links lead to desirable statistical properties of the GLM and hence tend to be used by default. The overall fit of the model and other considerations such as intuitive appeal may be more important and thus motivate to use a non-canonical link. Table 1.1 gives a summary of commonly used distributions and their canonical link function.

For example, the Poisson distribution to model the number of claims. The Poisson

Distribution	Canonical Parameter	Canonical Link $\eta = g(\mu)$	Link Function
Normal	$\theta = \mu$	$\eta = \mu$	identity
Binomial	$\theta = \ln(\frac{\mu}{1-\mu})$	$\eta = \ln(\frac{\mu}{1-\mu})$	logit
Poisson	$\theta = \ln(\mu)$	$\eta = \ln(\mu)$	log
Gamma	$\theta = \frac{-1}{\mu}$	$\eta = \frac{-1}{\mu}$	reciprocal

Table 1.1: Common Canonical Link Functions for EDF Models

distribution is a member for exponential dispersion family with a probability mass function.

$$\begin{aligned} f(y_i;\mu_i) &= \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!} \\ &= \exp\left[\log\left(\frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!}\right)\right] \\ &= \exp\left[\frac{y_i \log \mu_i - \mu_i}{1} - \log y_i!\right]. \end{aligned} \tag{1.9}$$

Substituting $\ln \mu = \theta$ to match the exponential family parameterization. This produces,

$$f(y_i;\theta_i) = \exp\left[\frac{y_i\theta_i - e^{\theta_i}}{1} - \ln y_i!\right], \tag{1.10}$$

where $b(\theta) = e^\theta$ and $c(y,\phi) = -\ln y!$. For the Poisson distribution $a(\phi) = \phi = 1$, the mean and variance are equal and can be derived as follows,

$$\begin{aligned} \mathbb{E}(Y) &= b'(\theta) = e^\theta = \mu \\ \mathbb{V}(Y) &= a(\theta)b''(\theta) = e^\theta = \mu. \end{aligned} \tag{1.11}$$

The reason for using a Poisson GLM is to produce a multiplicative model to estimate the expected annual claims frequency. A multiplicative model is desirable so that the individual pure premium can be calculated as a product of the variables' relativities. This is basically the exponential of the GLM coefficient of a variable multiplied by the level of the variable. The log-link takes a range of linear predictor values η_i from $(-\infty, \infty)$ and maps it onto a range of $(0, \infty)$ for the claim frequency. The log-link for the Poisson distribution generates

a multiplicative model as follows,

$$\begin{aligned} \log(\mu_i) &= X_i^T \beta, \\ \mu_i &= e^{(X_i^T \beta)}. \\ \mu_i &= \left(e^{\beta_1}\right)^{X_{i0}} \left(e^{\beta_2}\right)^{X_{i1}} \dots \left(e^{\beta_p}\right)^{X_{i,p-1}}. \end{aligned} \tag{1.12}$$

where μ_i is the expected annual claims frequency for the i^{th} policy. The multiplying factor for each predictor is given by e^{β_j} for the p predictors.

When trying to estimate annual claims frequency, an offset term is put into the model to account for the varying number of policy years also known as exposure years. Let Y_i be the be a random variable representing the total number of claims and an exposure variable t_i in insurance application it would be the number of policy years. Then we get $\mu_i = \mathbb{E}[Y_i/t_i]$. Substituting this in Equation (1.12),

$$\log(\mathbb{E}[Y_i/t_i]) = X_i^T \beta, \tag{1.13}$$

we can separate the terms as follows,

$$\log(\mathbb{E}[Y_i]) = X\beta + \ln(t_i). \tag{1.14}$$

The offset term is a known effect and is one of the model inputs. It is given as a vector of length n and must be included in the model estimation because the number of claims depends on the number of years of observations.

1.4 Maximum Likelihood Estimation

To estimate the GLM regression parameters β_j, the log-likelihood function is derived and differentiated with respect to the parameter of interest. The negative log-likelihood equals

$$l(y_1 \dots, y_n; \theta, \phi) = -\sum_{i=1}^{n} \log f(y_i, \theta_i) \tag{1.15}$$

where $f(y_i;\theta_i)$ of the selected distribution is given for $Y_i = y_i$ and $\theta = (\theta_1, \ldots, \theta_n)$ is a function of the β_j's. By choosing a distribution to model random variables Y_i it allows one to apply Maximum Likelihood Estimation (MLE). The log-likelihood (1.7) can be maximized by calculating partial derivatives with respect to the β_j and setting them equal to zero. The system of partial derivatives, also known as gradients of the log-likelihood, is called score functions and is defined as:

$$s(y;\theta,\phi) = \frac{\partial}{\partial\beta} l(\theta;\phi,y) \tag{1.16}$$

The maximum likelihood estimate $\hat{\beta}$ is then found as the solution to the system of equations $s(y;\theta) = 0$. The partial derivative with respect to β_j is

$$\frac{\partial l(y;\beta,\phi)}{\partial\beta_j} = \sum_{i=1}^{n} \frac{\partial}{\partial\beta_j}\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right] \tag{1.17}$$

$$= \sum_{i=1}^{n} \frac{1}{a(\phi)}\left[y_i\frac{\partial\theta_i}{\partial\beta_j} - \frac{\partial b(\theta_i)}{\partial\beta_j}\right]. \tag{1.18}$$

From the chain rule of differentiation,

$$\frac{\partial}{\partial\beta_j} = \frac{\partial}{\partial\theta_i}\frac{\partial\theta_i}{\partial\mu_i}\frac{\partial\mu_i}{\partial\eta_i}\frac{\partial\eta_i}{\partial\beta_j}, \tag{1.19}$$

the result is

$$\frac{\partial l(y;\beta,\phi)}{\partial\beta_j} = \sum_{i=1}^{n} \frac{(y_i-\mu_i)x_{ij}}{a(\phi)b''(\theta_i)g'(\mu_i)}. \tag{1.20}$$

Given the definition of the exponential dispersion family, the variance of Y_i, $\mathbb{V}(Y_i) = a(\phi)b''(\theta_i)$. This can be rewritten using the variance function and by adding weights for $\mathbb{V}(Y_i) = (\phi/w_i)V(\mu_i)$, with $a_i(\phi) = \phi/w_i$. Therefore, the solution becomes,

$$\frac{\partial l(y;\beta;\phi)}{\partial\beta_j} = \sum_{i=1}^{n} \frac{w_i(y_i-\mu_i)x_{ij}}{\phi V(\mu_i)g'(\mu_i)} = 0. \tag{1.21}$$

Closed form solution for GLMs can not always be derived. Therefore, it is expected that computer software can be used to compute MLEs numerically, using efficient methods. Statistical packages have numerical methods to maximize the log-likelihood function given in Equation (1.20). Common techniques include iteratively reweighted least squares (IRLS), Fisher Scoring Algorithm, Newton Raphson and gradient descent methods. For example, the

R software can also be used to find the MLE with numerical iterative methods such as gradient descent or grid search MLE. In the lasso problem, that will be further discussed in the next chapter, a ℓ_1 penalty is added to the log-likelihood function. The problem can no longer be solved in closed form even in cases where the unpenalized likelihood can be maximized analytically. Therefore, we revert to gradient descent methods to solve the problem.

1.5 Goodness of Fit and Deviance Residuals

A statistical measure to check model performance called deviance residual is commonly used to evaluate and compare GLMs. Deviance residuals are based on the log-likelihoods of the distribution. Recall the log-likelihood function of exponential family distributions derived in Equation (1.7) is:

$$l(y;\theta,\phi)=\sum_{i=1}^{n}\left[\frac{y_i\theta_i-b(\theta_i)}{a(\phi)}+c(y_i,\phi)\right]. \tag{1.22}$$

Generally, a GLM is constructed in a way to maximize the coefficients of the log-likelihood function. This is achieved as explained in the previous section when we maximize the likelihood by taking partial derivatives with respect to θ_i. Consider the saturated model with $\hat{\mu}_i^S=y_i$, where the model replicates the observed values and the corresponding value of the log-likelihood given as $l(y;\theta^S,\phi)$. Then consider, any other model, with a log-likelihood $l(y;\theta^M,\phi)$, calculated from the maximized model. Taking the difference between the log-likelihoods of the saturated model S and model M we get,

$$l(y;\theta^S,\phi)-l(y;\theta^M,\phi)=\sum_{i=1}^{n}\Big[\frac{y_i(\hat{\theta_i^S}-\hat{\theta_i^M})-(b(\hat{\theta_i^S}))-(b(\hat{\theta_i^M}))}{a(\phi)}\Big]. \tag{1.23}$$

This quantity is positive since $l(y;\theta^S,\phi)\geq l(y;\theta^M,\phi)$. For $a_i(\phi)=\phi/\omega_i$, the deviance for model M becomes,

$$\begin{aligned} D^*(y;\hat{\theta}^M,\phi)&=2\big[l(y;\theta^S,\phi)-l(y;\theta^M,\phi)\big]\\ &=\frac{1}{\phi}\sum_{i=1}^{n}2\omega_i\big[y_i(\hat{\theta_i^S}-\hat{\theta_i^M})-(b(\hat{\theta_i^S}))-(b(\hat{\theta_i^M}))\big]. \end{aligned} \tag{1.24}$$

Note that the deviance residual for the saturated model will be $D^*(y;\hat{\theta}^S,\phi)=0$. Another example, is the deviance for the Gaussian distribution. With a Gaussian distribution the

identity link is the canonical link and the deviance for the M model becomes,

$$\begin{aligned} D^*(y;\hat{\theta}^M,\phi) &= \sum_{i=1}^{n} 2\big[y_i(y_i-\hat{\mu}_i^M) - (y_i^2/2 - (\hat{\mu}_i^M)^2/2)\big] \\ &= \sum_{i=1}^{n}(y_i-\hat{\mu}_i^M)^2. \end{aligned} \tag{1.25}$$

The deviance of the Gaussian distribution becomes the residual sum of squares, which is a common goodness of fit measure used in linear regression. For the Poisson distribution, the response variable is derived in Equation (1.10) with $\theta_i = \ln \mu_i$ and distribution means μ_i.

Recall that in the saturated model $\hat{\mu}_i^S = y_i$ and so $\hat{\theta}_i^S = \ln y_i$. The fitted parameters of the Poisson are $\hat{\theta}_i^P = \ln \mu_i^P$ and thus the deviance becomes,

$$D^*(y;\hat{\theta}^P,\phi) = \sum_{i=1}^{n} 2\big[y_i(\ln y_i - \ln \hat{\theta}_i^P) - (y_i - \hat{\theta}_i^P)\big]. \tag{1.26}$$

It is to be noted that the deviance will be approaching zero as fitted means $\hat{\theta}_i^P$ approach the observed valued y_i. The residual deviance is usually used to compare the performance of two nested models. Consider a model P with p variables and another model Q with q variables where $q > p$. To compare, the difference in the residual deviances of the models is taken. This is equivalent to a likelihood-ratio statistic:

$$D^*(y;\hat{\theta}^P) - D^*(y;\hat{\theta}^Q) = 2\big[l(y;\theta^Q) - l(y;\theta^P)\big] = 2\ln\frac{L(y;\hat{\theta}^Q)}{L(y;\hat{\theta}^P)} \tag{1.27}$$

This statistic has an asymptotic Chi-Square distribution with $q-p$ degrees of freedom. However, increasing the number of variables does not necessarily mean an improved fit and that is where a regularized model might be of advantage.

Another goodness of fit measure is the null deviance which is the residual deviance for a GLM with only a constant term, the intercept. The null deviance provides information about how much better a model M performs when adding variables to the intercept. It is used as an upper bound on residual deviance while the deviance of the saturated model, is the lower bound since its value is zero: $0 \leq D^*(\hat{\theta}^M;y) \leq D^*(\hat{\theta}^{Null};y)$.

1.6 Generalized Linear Models and Non-Life Insurance

Insurance companies accept premiums to indemnify a policyholder for the occurrence of an uncertain event. Accurate pricing of insurance premiums is an important goal for insurance providers. GLMs are the standard predictive modeling framework that non-life insurers use for portfolio segmentation and estimation of the pure premium for homogeneous classes of policyholders. They are also accepted in North America by regulatory entities as the standard tool for ratemaking. GLMs are used to predict and explain the heterogeneity among policyholders. This leads to successful adverse risk selection. Historical claim frequency and severity is the response variable of the model. It is therefore used to fit a model that can then make predictions for the pure premium. It is important not just to look at the expected future outcome of the model but to look at the variability of the predictions as well. The variability of the ultimate model outcome is critical to understanding the extent of the risks faced by the risk-bearing entity that either has adopted or is contemplating the adoption of a certain risk. One of the methods to control variability is the shrinkage method which will be discussed in further detail in Chapter 2.

The frequency and severity of claims are usually modeled independently. Consequently the pure premium is then the product of the two means, estimated from separate GLMs, whether fitted at the individual or class level. Aggregate losses can also be modeled directly as a GLM using a Tweedie distribution which is also a member of the exponential dispersion family. The Tweedie distribution then models aggregate claims as a compound Poisson-gamma sum.

Assume that aggregate losses Y_i, for a given class of policyholders, are represented by the sum of individual claim amounts,

$$Y_i = \sum_{k=1}^{N_i} Y_{ik} \tag{1.28}$$

with $Y_i = 0$ if $N_i = 0$ and where N_i is the number of claims incurred and Y_{ik} is the severity random variable for i $k = 1, \ldots, N$. Here, N_i and $Y_{i1}, \ldots, Y_{iN}$ are assumed to be independent and are modeled separately using a different GLM distribution and link function. For severity

the gamma distribution is widely accepted for insurance claims and used only for policies that did claim, that is $N_i > 0$. Likewise, for frequency modeling the Poisson is used for all claim counts. Assuming independence between frequency and severity we have,

$$\mathbb{E}(Y_i) = \mathbb{E}(N_i)\mathbb{E}(Y_{ik}), \qquad \mathbb{V}(Y_i) = \mathbb{E}(Y_{ik})^2\mathbb{V}(N_i) + \mathbb{V}(Y_{ik})\mathbb{E}(N_i) \tag{1.29}$$

For these, a vector $x_i = (x_{i0}, \ldots, x_{i,p-1})$ of p covariates is used to fit separate GLMs for N_i and Y_{ik}. These covariates incorporate information about the individual policyholders that help make predictions about their claim behavior in terms of frequency and severity.

The loss cost also known as the pure premium is the total claim size divided by the exposure, i.e. the average amount paid per unit time. The pure premium is calculated assuming independence as the product of the frequency and severity. The same paradigm can be used for the loss ratio, which is calculated as $\frac{losses}{premium}$. And the loss cost is split as follows,

$$\text{pure premium} = \text{frequency} \times \text{severity} = \left(\frac{\text{number of losses}}{\text{exposure}}\right) \times \left(\frac{\text{amount of losses}}{\text{number of losses}}\right) \tag{1.30}$$

The frequency and severity are calculated separately and then multiplied together to get the pure premium. Losses can also be modeled directly with a Tweedie GLM. However, frequency and severity may be affected by different predictors and thus models may be different. Modeling them separately may provide more insight into the loss process than modeling the losses directly.

Chapter 2

Regularization

2.1 Introduction

Regularization is a machine learning technique that refers to subset selection methods. It is mainly a technique to improve the generalization of a learned model and thus prevent overfitting. These methods can use different convergence criteria to fit a model with the selected predictors since no closed form solution is achievable. The model fits p predictors using a technique that regularizes the coefficient estimates, or shrinks the coefficients to zero depending on their predictive ability (Hastie et al., 2008). The imposed penalty term in the model forces many coefficients to be zero and removes them from the fitted model. The penalty term λ controls which variables are included in the model based on their correlation with the response Y. Intuitively, $\lambda = 0$ reduces the problem to an unregularized model and on the other hand, when $\lambda = \infty$ the coefficients are forced to zero. Along the grid of λ values, variables are added to the model based on their significance. The resulting order is a natural proxy for variable importance and a mechanism for model selection by mapping the size of the fitted model. This is called, the active set with a specific number of variables given the corresponding value of λ.

This is useful specially when dealing with large data sets with hundreds or thousands of predictors. In general shrinkage methods will help build parsimonious models that are easier

to interpret without loosing significant predictive ability. It is not intuitive to understand that shrinkage methods improve model fit, but with parameter tuning this can be achieved. It is shown that shrinking the coefficient estimates can reduce their variance significantly (James et al., 2014). The two most popular regularization methods are ridge regression and lasso (Tibshirani, 1996), which are discussed in further detail in the following sections. For additional details see Tibshirani (1996).

2.2 Lasso Regularization

The least absolute shrinkage and selection operator (lasso) is a regularization technique that performs feature selection and coefficient estimation to solve the high dimensionality problem in model construction (James et al., 2014). For example, in linear models the lasso minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant s.

This technique is aimed to be an efficient model selection method in high dimensionality contexts. The lasso includes an ℓ_1-penalty term that constraints the minimum size of the estimated model coefficients, forcing the model to have fewer parameters. This dimensionality reduction technique creates a subset by generating zero valued coefficients. Because of the nature of the constraint it tends to produce some coefficients that are exactly zero. This is in contrast to ridge regression which will shrink all of the coefficients towards zero, but will not set any of them exactly to zero, unless $\lambda = \infty$, where λ is the coefficient of the ℓ_2-penalty for the ridge. The lasso technique is general and can also be extended to GLM and tree-based models. The lasso coefficient estimates solve the following problem;

$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2, \qquad \text{subject to } \sum_{j=1}^{p} \mid \beta_j \mid \leq s, \tag{2.1}$$

where s has to be greater than zero. The lasso estimate is a non-linear and non-differentiable function of the response values even for fixed values of s. Thus it is difficult to obtain an accurate estimate of the standard error. Bootstrap methods can be used to estimate the standard error either for a fixed s or an optimized s.

2.2.1 Lasso Regularization for Linear Models

For linear regression the lasso penalty is added to the ordinary least squares. We penalize the squared error loss in linear regression by adding the ℓ_1-penalty and then solving the objective function. Using a Lagrange multiplier λ, we see that the lasso coefficients, $\hat{\beta}_\lambda^L$, minimize the following equivalent optimization problem to equation 2.1 in Lagrange form as follows:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_i j)^2 + \lambda \sum_{j=1}^{p} \mid \beta_j \mid . \tag{2.2}$$

The ℓ_1-norm of the coefficient vector $\beta = (\beta_0, \ldots, \beta_p)$ is given by $\|\beta\|_1 = \sum \mid \beta_j \mid$. The tuning parameter λ determines how many coefficients are set to zero. This is usually determined by cross-validation or a grid search. A parameter β has to be sufficiently large to be included in the model, however too large values of λ will force all coefficients to be zero anyhow. Hence, lasso performs variable selection making models easier to interpret like subset selection. These are called sparse models, i.e. models that only include a subset of variables with most predictive power.

2.2.2 Lasso Regularization for Generalized Linear Models

For GLMs, penalizing the negative log-likelihood with the ℓ_1 - penalty is called the lasso Regularization. In many examples this is conceptually similar to the case with squared error loss in linear regression, due to the convexity of the negative log-likelihood. To apply the lasso regularization method to GLMs we have to solve the following optimization problem:

$$\hat{\beta} = \arg\min_{\beta} \; l(Y, X; \beta) + \lambda \|\beta\|_1, \tag{2.3}$$

by minimizing the constrained likelihood function, i.e. the negative log-likelihood function given by $l(Y, X; \beta)$ plus a constraint on the size of the parameters. It is not necessary to use the log-likelihood function and we can use any convex loss function denoted as $\mathcal{L}$. For example, given the response is binary, the logistic loss is given as,

$$\mathcal{L}(Y, X; \beta) = -\big[Y^T(X\beta) - 1^T \log(1 + \exp(X\beta))\big]. \tag{2.4}$$

Note, that the ℓ_1-penalization is a special case of the group ℓ_1-penalty that is further explained in the next subsection of this chapter. The convexity of the log-likelihood (Buhlmann and Van de Geer, 2011) implies statistical properties that are attractive as well as a computational simplicity of algorithms.

Assumption 1. *Assume that the given loss function* $\mathcal{L}(X, Y; \beta) \geq C > -\infty$ *for all* β, X_i, Y_i *given* $(i = 1, \ldots, n)$*, is continuously differentiable with respect to* β*, and that the empirical risk* $\mathcal{L}(\beta)$ *is convex.*

For any convex loss function, other than the squared error loss, numerical optimization methods are needed for parameter updates. The algorithm takes a step in the direction of steepest decrease of the loss function. These small steps are taken in the opposite direction of the gradient of the negative log-likelihood function in Equation (1.15) to ensure convergence. An approximate minimization and a convergence to a local minimum will be sufficient for computational purposes.

The parameter estimates $\hat{\beta}(\lambda)$ are calculated for a grid of λ values. For example, the algorithm could start by $\lambda_{\max}$ where all parameters in all the groups will be equal to zero. Then moving from this starting point and proceeding iteratively to include more variables in the model, as the value of λ decreases, until λ is close or equal to zero. Cross-validation is then used to choose the optimal parameter $\hat{\lambda}$ among the candidate values from the grid.

Figure 2.1 shows how coefficients change as s changes for a lasso GLM. Labeled on the bottom are the value of s. Each curve represents a coefficient as a function of the scaled lasso parameter s. Note that the absolute value of the coefficients tends to zero as the value of s goes to zero. In the upper scale the number of variables captured is denoted out of 27. The optimal value of s and λ accordingly is chosen by cross validation.

2.3 Group Lasso

The Group-lasso (Simon et al., 2013) is an extension of the lasso that does variable selection on non-overlapping groups of variables and sets groups of coefficients to zero. This means

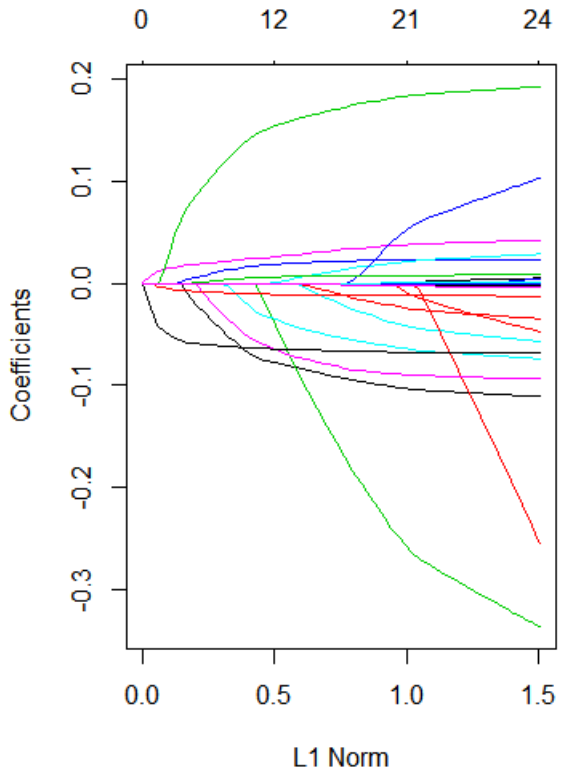

Figure 2.1: Shrinkage Coefficients for lasso GLM

Bottom: The value of constraint s. Top: Number of variables captured out of 27. Each curve represents a coefficient as a function of the scaled lasso parameter s.

groups of variables are given, potentially with no overlaps between the groups. For high dimensional parameter vectors a group structure can be found where the parameter space is partitioned into disjoint pieces. Sometimes, we are interested in finding important explanatory factors to predict the response variable, where each explanatory factor is represented by a group of input features. The group structure is the simplest for high-dimensional data. The goal is to model high-dimensional data in a linear or a generalized linear model and to have sparsity with respect to whole groups. The group-lasso (Yuan and Lin, 2006) achieves such group sparsity. The non-overlapping structure is not easily applicable in practice therefore the overlapping structure is considered. This simply means that variables are given with potential overlaps between the groups. As a consequence the resulting optimization is much more challenging to solve due to the group overlaps.

We will consider the group-lasso penalty for linear models with squared error loss and a non-squared error loss for generalized linear models. First when we estimate parameters with a group structure we encourage sparsity at the group level, i.e. either all group entries are zero or non-zero. This can be achieved using the group-lasso penalty for K groups of variables,

$$\lambda \sum_{k=1}^{K} \gamma_k \left\| \beta_k \right\|_2 , \tag{2.5}$$

where $\left\| \beta_j \right\|_2$ denotes the standard Euclidean norm and λ is the tuning parameter. Here, the γ_i serves as a balancing weight where groups are of different sizes. This is an extension of lasso for selecting groups of variables instead of individual variables. The group-lasso estimator can be defined by solving the following objective function for linear models and for generalized linear models,

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \mathcal{L}(Y, X; \beta) + \lambda \sum_{k=1}^{K} \gamma_k \left\| \beta_k \right\|_2 , \tag{2.6}$$

where $\mathcal{L}(Y, X; \beta)$ is the loss function for linear and generalized linear models. It represents the squared error or the negative log-likelihood function, respectively. For squared error loss,

$$\mathcal{L}(Y, X; \beta) = \left\| Y - \sum_{k=1}^{K} X\beta_k \right\|_2^2 , \qquad X \in \mathbb{R}^{n \times p}, Y \in \mathbb{R}. \tag{2.7}$$

For generalized linear models, the loss function $\mathcal{L}(Y, X; \beta)$ which is the negative log-likelihood, varies by distribution and chosen link function. As an example, the logit link is given as $\log \frac{p_i}{1-p_i} = f_\beta(X)$ for the logistic function and then the loss function becomes,

$$\mathcal{L}(Y, X; \beta) = -Y f_\beta(X) + \log \Big(1 + \exp \big(f_\beta(X) \big) \Big), \qquad X \in \mathbb{R}^{n \times p}, Y \in 0, 1, \tag{2.8}$$

for,

$$f_\beta(X_i) = \eta_i = X_i^T \beta, \tag{2.9}$$

which describes the linear predictor. The estimator $\hat{\beta}(\lambda)$ is solved my minimizing the convex objective function in β.

Another penalty blends the lasso ℓ_1-norm with the group-lasso ("two-norm"). The advantage of this penalty is the fact that it yields solutions that are sparse at both the group and

individual feature levels. Suppose that the p predictors are divided into K groups, with p_k the number in group k. We denote the matrix X_k to represent the predictors corresponding to the k-th group, with coefficient vector β_k. Yuan and Lin (2006) proposed the group-lasso which solves the convex optimization problem

$$\min_{\beta \in \mathbb{R}^p} \mathcal{L}(Y, X; \beta) + \lambda \sum_{k=1}^{K} \sqrt{p_k} \left\| \beta_k \right\|_2 \tag{2.10}$$

where the $\sqrt{p_k}$ term accounts for the varying group sizes and $|| \cdot ||_2$ is the Euclidean norm. This penalty acts like the lasso but at a group level and depending on λ an entire group of predictors can drop out of the model. If all the group sizes are one, it reduces to lasso. The group-lasso does not, however, yield sparsity within a group. This means that if a subset of parameters are non-zero within a group, the remaining parameters will all be non-zero. A general penalty that yields sparsity at both the group and individual feature levels is achieved by adding a γ penalty term, in order to select groups and predictors within a group (Friedman et al., 2010).

In the multi-variate case, the sparse group-lasso criterion solves the following optimization problem:

$$\hat{\beta} = \arg\min_{\beta} \mathcal{L}(Y, X; \beta) + \lambda_1 \sum_{k=1}^{K} \left\| \beta_k \right\|_2 + \lambda_2 \left\| \beta \right\|_1, \tag{2.11}$$

where $\mathcal{L}(Y, X; \beta)$ is the negative log-likelihood function, that can be derived for different distributions. This can be generalized for any loss function as long as it is a convex function. The γ_i in Equation (2.6) control how some groups are penalized more or less than others. This procedure acts like lasso at the group level and depending on the value of the parameter λ_1 it could lead to an entire group of predictors dropping out of the model. The value of λ controls how much the coefficients are penalized. This is usually optimally chosen by cross validation or grid search over a specified range of possible λ values. The γ_i is chosen so that if the signal were pure noise, then all groups would equally likely be nonzero. When $\lambda_2 = 0$, criterion (2.11) reduces to the lasso in (2.6).

An active set strategy is used for sparse problems with a large number of groups K but with only few of them being active. This is used to speed up the algorithm considerably. An

active set is here defined as the set of groups whose coefficient vector is non-zero. If we go through the groups we then restrict ourselves to the current active set and visit only "rarely" the remaining groups. So this means the active set is only updated after a certain number of iterations that can be specified. Thus, the number of iterations in the algorithm is reduced.

An accelerated gradient descent type of algorithm is used for the optimization by solving the smooth and convex dual problem (Yuan et al., 2013). This method is efficient and can allow convergence at a fast rate even for non-smooth convex problems. Other methods to solve the group-lasso optimization are the block coordinate descent (Buhlmann and Van de Geer, 2011). Other commonly used methods of fit are some form of gradient descent, and convergence can be confirmed by checking that the solutions satisfy the Karush-Kuhn-Tucker (KKT) conditions. These optimality conditions for the group-lasso are essentially equivalent to compute,

$$\begin{aligned} \left\| X_i^T (Y - \hat{Y}) \right\|_2 &< \gamma_i \lambda, \qquad \hat{\beta}_i = 0, \\ \left\| X_i^T (Y - \hat{Y}) \right\|_2 &< \gamma_i \lambda, \qquad \hat{\beta}_i \neq 0, \end{aligned} \tag{2.12}$$

where the fitted values of the group-lasso are $\hat{Y} = \sum_{k=1}^{K} X_k \hat{\beta}$. In the R package "glinternet2",the group-lasso Interaction Network, which is an extension of the "glinternet" that includes Poisson and gamma families, an adaptive version of Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) and cyclic group-wise coordinate descent is used. This package has been extended to be able to model claim frequency and severity.

2.4 Ridge Regression

The other shrinkage method that is similar to lasso is the ridge regression which imposes an ℓ_2 penalty instead. Ridge regression is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different objective function. Compared to lasso, the objective function in ridge regression is differentiable and a closed form solution

can be found. The ridge coefficient estimates solve the following problem:

$$\arg\min_{\beta} \sum_{i=1}^{n}(y_i - \sum_{j=0}^{p-1}\beta_j x_{ij})^2 \quad \text{subject to} \quad \sum_{j=0}^{p-1}\beta_j^2 \leq s. \tag{2.13}$$

The objective function to be minimized is,

$$\arg\min_{\beta} \sum_{i=1}^{n}(y_i - \sum_{j=0}^{p-1}\beta_j x_{ij})^2 + \lambda \sum_{j=0}^{p-1}\beta_j^2. \tag{2.14}$$

Same as least squares, ridge regression searches for coefficient estimates that fit the data well, by making the root of the sum of least squares small. The second term, $\lambda \sum_{j=0}^{p-1}\beta_j^2$, the shrinkage penalty, is small when the coefficients are close to zero, and so it has the effect of shrinking the coefficient estimates towards zero. The tuning parameter λ controls the relative impact of these two terms on the regression coefficient estimates. Intuitively, when $\lambda = 0$, the penalty term has no effect, and ridge regression reduces to least squares regression.

Adding a ridge penalty to the group-lasso results in the class of the elastic-net. The objective function for linear model case would be as follows,

$$\arg\min_{\beta} \frac{1}{2}\left\| Y - \sum_{k=1}^{K} X_k\beta_k \right\|_2^2 + \lambda \sum_{k=1}^{K}\gamma_k \left\|\beta_k\right\|_2 + \alpha \left\|\beta\right\|_2^2. \tag{2.15}$$

2.5 Comparing Lasso and Ridge Regression

lasso has a major advantage over ridge regression which is the sparse solution obtained when the problem is solved. Ridge regression, on the other hand, reduces the values of variable coefficients but does not necessarily forces them to zero. This means, that variable selection is not achieved through the ridge regression. Figure 2.2 shows a comparison of contours of the errors and constraint functions between the ℓ_1 and ℓ_2 penalty. The red ellipses in the two figures represent regions with given mean square error and the corresponding optimal value of $\hat{\beta}$, while the green circle and green square represent the space of solutions for ridge and lasso regression, respectively. The end results are represented by the intersection of the green space with the red circles. It can be seen that the lasso solution obviously forces one of the parameters, here β_1 to be zero while ridge regression only reduces its value. Geometrically,

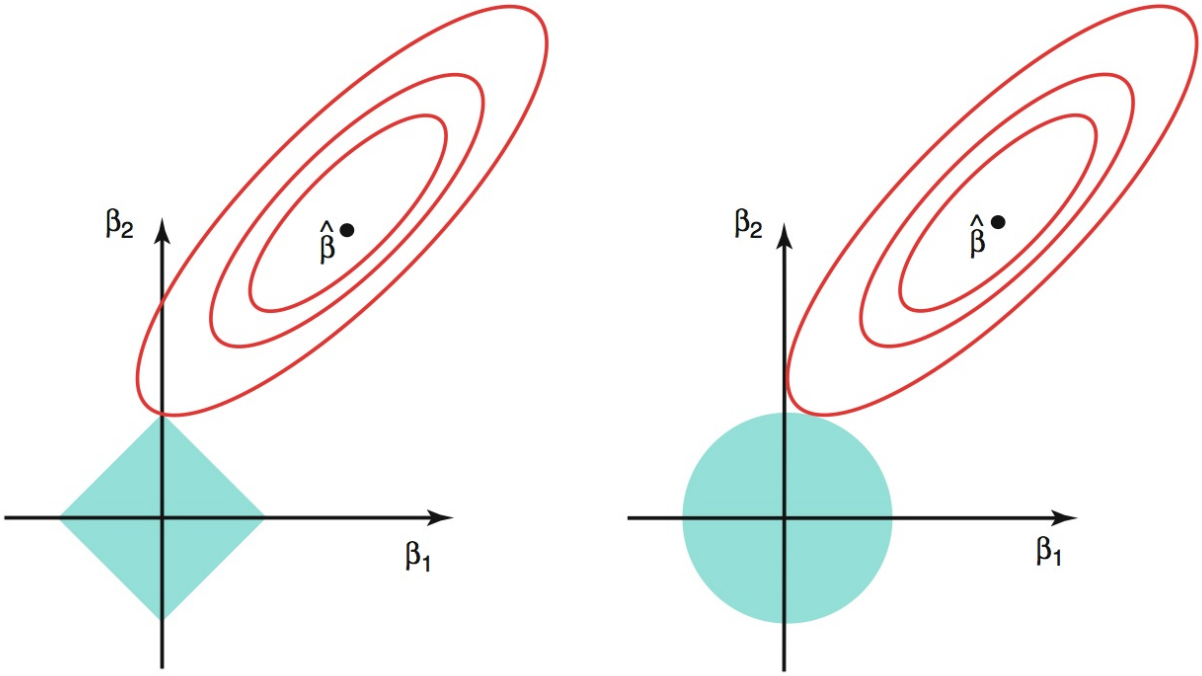

Figure 2.2: Comparison of Contours Between lasso and Ridge Regression

this can be explained by the analogy that the expense of traveling a straight line is less than that of a circle. The lasso is therefore more effective in terms of forcing parameters to be equal to zero. Overall, it can be shown how solutions change from the least squares estimate compared to lasso and ridge regression.

An obvious advantage of ridge regression over least squares is in terms of the bias-variance trade-off. As λ increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias. For the least squares coefficient estimates, which corresponds to ridge regression with $\lambda = 0$, the variance is high but there is no bias. Regularization methods do control variability in the parameter estimate and thus as λ increases, the shrinkage of the ridge coefficient estimates leads to a substantial reduction in the variance of the predictions. However, at the expense of a slight increase in bias. This works particularly well when the least squares estimates have high variance.

2.6 Subset Selection

The problem of selecting the best model from among the 2^p possible models is not easy. The aim is to select the best subset through an exhaustive search method. This can be done as shown in Table 2.1, step by step. This subsection is aimed to show what used to be done to select a subset before regularization methods gained popularity.

Best Subset Selection
1. Starting with M_0, the null model, which contains no predictors. This model gives the sample mean of each prediction. **2.** Then fit a ordinary least square on all $\binom{p}{k}$ for $k = 1, \ldots, p$ models that contain exactly k predictors. Among these k models pick the best, and call it M_k. Here best is defined as having the smallest root for the sum of least squares, or equivalently largest R^2. **3.** Among $M_0, ..., M_p$ select a single best model using cross-validated prediction error, AIC, BIC, or adjusted R^2.

Table 2.1: Steps to Select Best Subset

Following these steps to find the best subset, the problem is reduced from having 2^p possibilities to $p + 1$. To select a single best model, we must simply choose among these $p + 1$ options. This task must be performed with care, because the sum of least squares of these $p + 1$ models decreases monotonically, and the R^2 increases monotonically, as the number of features included in the model increases. Therefore, if we solely rely on these statistics to select the best model, then we will always end up with a model involving all of the variables. The problem is that a low sum of least squares or a high R^2 indicates a model with a low training error. However the model of interest to us and to perform predictive modeling is the one that has a low test error. This also applies to other types of models, such as logistic regression and other GLMs. Although this method is simple, it suffers from computational limitations. As the number of p predictors increases the number of models

to be considered rapidly increases and the method of best subset selection becomes quickly infeasible. Therefore, we tend to use a different regularization method to conduct best subset selection. For cross validation, the data set is divided into k subsets, and the holdout method

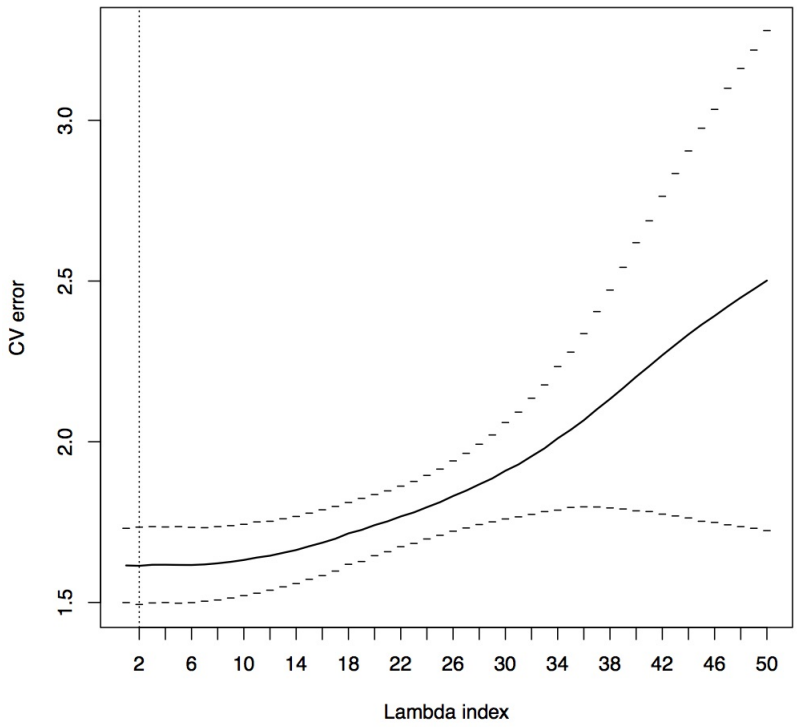

Figure 2.3: Ten Fold Cross Validation Error

is repeated k times. Each time, one of the k subsets is used as the test set and the other $k-1$ subsets are wrapped up together to form a training set. This is a way for testing how well the model performs on a subset of the data that was not used to fit the model. Here, for the Group-lasso Interaction Network "glinternet" procedure, we cross validate on different λ values. In Figure 2.3 the λ index keeps decreasing and thus the errors increase for low values of λ. This indicates, that a stronger penalty improves model fit and leads to a better model generalization on new data.

Chapter 3

Hierarchical Models

3.1 Introduction

Hierarchical modeling is an extension of GLMs that gives specific model parameters their own sub-model. This allows the building of models that can be grouped along a dimension containing multiple levels. Similar to linear and logistic regression, generalized linear models can be fit to multilevel and hierarchical structures by including coefficients for group indicators and then adding group-level models. These models are particularly used for longitudinal and repeated measures datasets that contain multiple levels for each of several subjects. Generally, hierarchical models are used only when data is hierarchically structured data and can be grouped, outperforming classical regression in predictive accuracy. Actuaries can consider hierarchical generalized linear models (HGLMs) as an alternative to "fixed effects" GLMs.

3.2 Theory and Assumptions

Multilevel models are extensions of regression where data is structured in groups and where coefficients can vary by group instead of only by variable. The advantage of using a hierarchical modeling framework is that it works well even if the data set contains a large number of

levels. It is used to model variations in the individual level regression coefficients. The multilevel model has the appeal of fitting two levels or more together. Classically, this is done by using indicator variables which can be cumbersome for large number of variables. Hierarchical modeling captures the variation of these coefficients across groups, make predictions for new groups, or account for group-level variation in the uncertainty for individual-level coefficients (Guszcza, 2008). If instead non-hierarchical models are deployed it would potentially require hundreds of indicator variables, specially for categorical variables. The hierarchical modeling framework provides an automatic mechanism to handle large categorical variables with a larger number of variables. Using hierarchical modeling can be useful to estimate model coefficients for particular groups since it incorporates group-level variations (Gelman and Hill, 2007).

3.3 Tree-Based Models

AdaBoost and gradient boosting are effective non-parametric techniques in machine learning to build ensembles of weak learners. Ensemble methods use multiple weak learning algorithms to obtain a better combined predictive performance. Tree-based models use decision trees as a predictive model that can be used for regression and classification. The tree is used as a set of splitting rules to segment the predictor space. For regression trees, a regression model is fitted at each node of the tree. The idea is to build a highly accurate predictive model by combining many relatively weak learners. This is achieved through boosting, which is a method of iteratively adding basis functions in a greedy way so that each additional basis function further reduces the selected loss function. It can be used in conjunction with many other types of learning algorithms to improve the performance (Freund and Schapire, 1999). At each iteration, a weak learner is built to fit a subset of the data and then the output of the weak learners is combined into a weighted sum that represents the final output of the boosted classifier. This leads to improving the learning process significantly. First, we train a weak model using samples of the training data according to the specified weight distribu-

tion then for the second iteration more weight is given to the misclassified sample, ones not correctly classified by the first weak learner, and less weight is given to the samples that are classified correctly. Then a model is trained using the samples specified by the updated weight distribution.

The AdaBoost algorithm is adaptive, due to the fact that subsequent weak learners are updated, to adjust those instances misclassified by previous classifiers. However, in some problems, this can make the algorithm sensitive to noisy data and outliers. Otherwise, it can be less susceptible to the overfitting problem than other learning algorithms. Even if the individual learners are weak, as long as the performance of each one is slightly better than random guessing, they can be combined and the final model will converge to a strong learner (Freund and Schapire, 1995).

We use screening with boosted trees to solve our problem by building an ensemble of weak learners, such as decision trees. Trees are used because of their ability to model nonlinear effects and high-order interactions. A boosted model is used as a screening device for interaction candidates by taking a set of all unique interactions from the collection of trees. Lowering the shrinkage parameter and increasing the number of trees improves the false discovery rate, but at a significant cost to speed.

Tree-based modeling algorithms are an active area of research in nonparametric statistics and machine learning. These include, random forests, boosted trees (GBM), decison trees, Bayesian trees and treed Gaussian processes.

3.3.1 Generalized Boosting Models

Boosting is a modeling technique that has been widely used in machine learning, data mining and statistical computations. Generalized boosted models (GBM) is an ensemble method that aggregates simple tree-based models into a final model. Boosting methods generalize the model by allowing the optimization of an arbitrary differentiable loss function $\mathcal{L}$.This is achieved through a sequential procedure and can be described as a method for iteratively building an additive model. $F_T(x)$ is the end learner we are trying to achieve through a pool

of weak learners as follows,

$$F_T(x) = \sum_{t=1}^{T} \alpha_t h_{jt}(x). \tag{3.1}$$

The weight α_t is given at each iteration t to each basis function and h_{jt} is a large pool of "weak learner" j as called in machine learning. These create a large pool of candidate predictors. Consider, h_{jt} to be the basis function selected as the "best candidate" to be modified at iteration t. Boosting has the ability to improve the accuracy of any given algorithm in its design, to adaptively select the next increment at each step, and to improve the fit of the model using the misclassified predictions. The sequential procedure iteratively adds trees to the ensemble to increase emphasis on poorly predicted cases. This is done by sequentially fitting the deviance residuals from the previous model. At each iteration, it searches for the basis function which goes in the direction of the gradients and thus gives the best decrease in the loss, and changing its coefficient accordingly. It searched at each iteration for the basis function which gives the steepest descent in the loss function, and change its coefficient accordingly. The final model is a linear combination of basis functions to optimize a given loss function or the intermediate trees. This is an attempt to find a linear combination of the members of some basis of functions to optimize a given loss function over any given sample, resulting in a better final model. The original boosting algorithm (Freund and Schapire, 1999) has been widely used. As opposed to other machine learning methods, boosting does not overfit.

Algorithm 1. *Gradient Boosting Algorithm*

1. Begin with a random initial vector regression coefficients $\beta^0 = 0$

2. For iteration $t = 0, 1, \ldots, T$ *:*

(a) Let the initial prediction be $F_i = \beta^{(t-1)\top} h_{jt}(x_i)$ *for* $i = 1, \ldots, n$ *in the initial model.*

(b) Set $\omega_i = \frac{\partial \mathcal{L}(y_i, F_i)}{\partial F_i}$ *for* $i = 1, \ldots, n$ *be the weight distribution chosen to minimize the loss function.*

(c) Identify the gradient as $j_t = \arg\max_j \mid \sum_i \omega_i h_{jt}(x_i) \mid$.

(d) Set $\beta_j^{(t)} = \beta_j^{(t-1)} - s_t$ *and* $\beta_k^{(t)} = \beta_k^{(t-1)}$, $k \neq j_t$.

(e) Update the model $F_i = F_{i-1} - \sum_i \omega_i h_j(x_i)$

Note that the gradient is given by $\sum_i \omega_i h_j(x_i) = \frac{\partial \mathcal{L}(y_i, F_i)}{\partial \beta_{jt}}$ where $\mathcal{L}(y, F)$ is an arbitrary differentiable loss function specified and the model gets updated according to this metric. $\beta^{(t)}$ is the current coefficient vector at iteration t and s_t is the step size of each interaction. This represents a general coordinate descent algorithm for a gradient boosting model which falls in the "weak learner" space. The algorithm attempts to find the optimal value of j_t and the direction of $\mid s_t \mid$. The sign of s_t will always be - sign$(\sum_i \omega_i h_{jt}(x_i))$, since the target is to reduce the loss. The idea is to apply a steepest descent step to this maximization problem. For example, the original AdaBoost algorithm uses the exponential loss $\mathcal{L}(y, F) = \exp(\text{-}yF)$, and an implicit line search to find the step size s_t once a "direction" j is chosen (Friedman, 2002).

Another issue to point out concerning the gradient boosting method is regularization, by shrinkage, which consists of modifying the updating rule $F_i = F_{i-1} - s_t \cdot \omega_i h_{jt}(x_i)$, for s_t between 0 and 1.

On the downside, interpreting an ensemble model such as GBM is extremely difficult. The output of the boosting models has no direct interpretation and can be interpreted only through implicit methods such as variable importance. Focusing on the predictability, GBM originates from paradigm of machine learning, by avoiding typical statistical parameter estimation in favor of algorithmically learning the relationship between covariates and response. As a result, it returns highly accurate results due to the high flexibility of the model.

3.4 Modeling Interactions

A statistical interaction occurs when the effect of one independent variable on the response variable changes depending on the level of another independent variable. It simply means that the relationship between levels of one variable is not constant for all levels of another variable. An example, in auto insurance, the age curve for driving experience, does not have the same shape for males and females. Here we discuss learning linear pairwise interactions and building a model that includes them. This is discussed in Chapter 4 for applications of

the group-lasso ("two-norm") via hierarchical group-lasso regularization in modeling claims frequency and severity.

Learning interactions is a challenging problem due to the number of variables involved. This could vary from thousands to millions, with a candidate interaction search space of about a billion or trillion terms. To define interactions, assume a response and explanatory variables, for which we expect interactions to be present if the response cannot be explained by additive functions of the given variables (Hastie and Lim, 2015). So an interaction exists in f, between x and y, when $f(x, y)$ cannot be expressed as $g(x)+h(y)$ only, for any functions g and h.

Actuaries care about including interactions in their models to increase their predictive power. Since interactions are considered a non-linear effect, adding them to a linear model has to be specified manually. Examining interactions manually can be a tedious and sometimes impossible task in models with a large number of candidate interactions. Therefore, we explore other more efficient methods to solve this problem. This is solved by introducing the overlapped group-lasso.

3.4.1 Strong and Weak Hierarchy

The goal here is to fit the first order interaction model in a way that obeys a strong hierarchy. A mode is described as to obey strong hierarchy when an interaction model includes those variables, that have both of its main effects present. A weak hierarchy means that it is sufficient for either of the main effects to be present. Since main effects can be viewed as deviations from the global mean, and interactions are deviations from the main effects, it usually does not make sense to have interactions without main effects. If a variable has significant interactions then it makes sense that the main effect is a significant variable as well. Including both in a model should yield better results. Therefore, it is preferable to use interaction models that are hierarchical and satisfy strong hierarchy for predictive modeling.

Let $\mathbb{E}(Y \mid X_1 = i, X_2 = j) = \mu_{ij}$, the conditional mean of Y given that X_1 takes level i and X_2 takes level j. This holds for categorical variables at any arbitrary number of levels.

Note that for categorical variables each level is treated as a variable on its own. But it also holds for continuous variables, so let $\mathbb{E}(Y \mid Z_1 = i, Z_2 = j) = \mu_{ij}$ where Z_1 and Z_2 are two continuous variables that are jointly distributed to predict the mean Y. In our model, we define interactions between two distinct continuous variables or interactions between any two levels of a categorical variable.

Proposition 1. *Definition of Interaction*
There are 4 *possible cases that satisfy a strong hierarchy assumption:*

1. $\mu_{ij} = \mu$ *(no main effects, no interactions),*

2. $\mu_{ij} = \mu + \theta_1^i$ *(one main effect* Z_1 *or* X_1*),*

3. $\mu_{ij} = \mu + \theta_1^i + \theta_2^j$ *(two main effects),*

4. $\mu_{ij} = \mu + \theta_1^i + \theta_2^j + \theta_{1:2}^{ij}$ *(main effects and interaction).*

This defines θ_i, for $i = 1, \ldots, p$, that represents the main effect coefficients for any of the p predictors, and $\theta_{i:j}$ denotes the interaction coefficients as defined in the four cases above. These satisfy a strong hierarchy since interactions are only present when both main effects are present as well. In what follows, the terms "main effect coefficients" and "main effects" will be used interchangeably, and likewise for interactions.

3.4.2 First Order Interaction Model

Given a response Y,

$$Y_i = \sum_{j=0}^{p-1} X_{ij}\theta_j + \sum_{j<l} X_{i,j:l}\theta_{j:l} + \xi_i, \tag{3.2}$$

where $\xi_i \sim N(0, \sigma^2)$ are the model errors, $X_{i,j:l}$ is the variables interaction and $\theta_{j:l}$ the corresponding interaction coefficient between j and l. For example the logistic model for binary responses $0, 1$,

$$\text{logit}(P(Y_i = 1|X)) = \sum_{j=0}^{p-1} X_{ij}\theta_{ij} + \sum_{j<l} X_{i,j:l}\theta_{j:l}. \tag{3.3}$$

The models are fit by minimizing an appropriate choice of loss function $\mathcal{L}$. These equations represent models that are unpenalized, yet they could be overparametrized and in fact they usually are, leading to overfitting. Thus we impose the relevant constraints for the coefficients θ. Particularity we will be imposing an ℓ_1 penalty. This function can be transformed into an optimization problem with constraints as follows:

$$\arg\min_{\mu,\theta} \mathcal{L}(Y_i, X_{i,j:j\leq p-1}, X_{i,j:l}; \theta), \tag{3.4}$$

subject to the relevant constraints. $\mathcal{L}$ can be any loss function depending on the response model. For a quantitative response model typically the squared error loss is used,

$$\mathcal{L}(Y_i, X_{i,j:j\leq p-1}, X_{i,j:l}; \theta) = \frac{1}{2}\left\| Y_i - \sum_{j=0}^{p-1} X_{ij}\theta_j + \sum_{j<l} X_{i,j:l}\theta_{j:l} \right\|_2^2, \tag{3.5}$$

and for the binomial response model a logistic loss is used and extensions include any convex loss function.

$$\begin{aligned} \mathcal{L}(Y_i, X_{i,j:j\leq p-1}, X_{i,j:l}; \theta) = -[Y_i^T(&\sum_{j=0}^{p-1} X_{ij}\theta_j + \sum_{j<l} X_{i,j:l}\theta_{j:l}) \\ &-1^T \log(1 + \exp(\sum_{j=0}^{p-1} X_{ij}\theta_j + \sum_{j<l} X_{i,j:l}\theta_{j:l}))]. \end{aligned} \tag{3.6}$$

Extensions include exponential family members, where the appropriate loss function can be derived. The negative log-likelihood is used as the loss function for count response models and for skewed continuous responses, i.e. Poisson and gamma models, respectively. Here, the Poisson model is derived as

$$\begin{aligned} \mathcal{L}(Y_i, X_{i,j:j\leq p-1}, X_{i,j:l}; \theta) = [\exp(&\sum_{i=0}^{p-1} X_{ij}\theta_j + \sum_{j<l} X_{i,j:l}\theta_{i,j:l}) \\ &-Y_i^T(\sum_{i=0}^{p-1} X_{ij}\theta_j + \sum_{j<l} X_{i,j:l}\theta_{j:l}) + \log(Y_i!)]. \end{aligned} \tag{3.7}$$

This can also be derived for other exponential family members with the appropriate loss and link functions. Since here the coefficients are unpenalized they satisfy the strong hierarchy condition. If a penalty is added, results will deviate from a strong hierarchy. The problem is how to fit interaction models whose solutions are sparse (variable selection effect) and also satisfy a strong hierarchy. A lasso penalty will achieve sparsity, but there is no guarantee that the solutions will have any form of hierarchy as well. This is the goal of this work, to find a solution, that specifically solves both problems simultaneously. For actuarial applications this solution will be also derived for Poisson and gamma models.

3.4.3 Strong Hierarchy Through Overlapped Group-Lasso

Strong hierarchy in interaction models can be achieved by adding an overlapped group-lasso penalty to the objective function in Problem (3.4) of Section 3.4.2. The results that follow hold for both squared error and logistic loss (Hastie and Lim, 2015) and can be extended to other loss functions. Consider the case with two categorical variables with levels L_1 and L_2 levels and indicator matrices are given as X_1 and X_2. The problem to solve becomes,

$$\begin{aligned}\arg\min_{\mu,\alpha,\tilde{\alpha}} \frac{1}{2} \left\| Y - \mu 1 - X_1\alpha_1 - X_2\alpha_2 - [X_1X_2X_{1:2}] \begin{bmatrix} \tilde{\alpha_1} \\ \tilde{\alpha_2} \\ \alpha_{1:2} \end{bmatrix} \right\|_2^2 \\ +\lambda(\|\alpha_1\|_2 + \|\alpha_2\|_2 + \sqrt{L_2 \|\tilde{\alpha_1}\|_2^2 + L_1 \|\tilde{\alpha_2}\|_2^2 + \|\alpha_{1:2}\|_2^2})\end{aligned} \tag{3.8}$$

subject to

$$\sum_{i=1}^{L_1} \alpha_1^i = 0, \qquad \sum_{j=1}^{L_2} \alpha_2^j = 0, \qquad \sum_{i=1}^{L_1} \tilde{\alpha}_1^i = 0, \qquad \sum_{j=1}^{L_2} \tilde{\alpha}_2^j = 0 \tag{3.9}$$

and

$$\sum_{i=1}^{L_1} \alpha_{1:2}^{ij} = 0 \text{ for fixed j}, \qquad \sum_{j=1}^{L_2} \alpha_{1:2}^{ij} = 0 \text{ for fixed i}, \tag{3.10}$$

for the interaction effect. Here, X_1 and X_2 each have two different coefficient vectors α_i and $\tilde{\alpha}_i$. This will result in the desired penalty, the overlapped penalty. The actual main effects

θ_1 and θ_2 are given by

$$\begin{aligned} \theta_1 &= \alpha_1 + \tilde{\alpha_1}, \\ \theta_2 &= \alpha_2 + \tilde{\alpha_2}, \\ \theta_{1:2} &= \alpha_{1:2}. \end{aligned} \tag{3.11}$$

To satisfy the strong hierarchy property and to obtain estimates that satisfy this, the term $\sqrt{L_2 \left\|\tilde{\alpha_1}\right\|_2^2 + L_1 \left\|\tilde{\alpha_2}\right\|_2^2 + \left\|\alpha_{1:2}\right\|_2^2}$ is responsible for imposing the required feature, since either $\tilde{\alpha_1} = \tilde{\alpha_2} = \tilde{\alpha_{1:2}} = 0$ or all are nonzero, i.e. interactions are always present with both main effects. α_i represents the parameter space $\tilde{\alpha}_i$ and α_i for the main effects, while $\alpha_{1:2}$ is for interaction effects. Hence $\tilde{\alpha}$ represents the penalized coefficients obeying strong hierarchy. These are always present when an interaction effect is present. Since the group-lasso has the property of "all zero" or "all nonzero" estimates, we also have that,

$$\theta_{1:2} \neq 0 \Longrightarrow \theta_1 \neq 0 \Longrightarrow \theta_2 \neq 0 \tag{3.12}$$

satisfying a strong hierarchy.

The parameterization of the constraints is a tricky problem, specially because the coefficients get penalized. Any representation of the problem that does not preserve symmetry will result in unequal penalization schemes for the coefficients. The symmetry of parameters is so important because it avoids overparameterization and this is avoided by the sum to zero constraint of the variable levels. Intuitively, the problem becomes more complicated as the number of variables and levels increases. This problem can be solved by an equivalent unconstrained group-lasso problem. This is advantageous since the problem can be now be represented in a symmetric way, thus avoiding the need for careful choices of parametrization. In addition, we only have to fit a group-lasso without constraints on the coefficients, which is a already well studied problem in the literature.

This is presented through two Lemmas. The first one states that because an intercept is fitted, the coefficient estimates $\hat{\beta}$ of categorical variables will have mean zero.

Lemma 1. *X is given to be an indicator matrix. The solution,*

$$\arg\min_{\mu,\beta} \frac{1}{2} \left\|Y - \mu 1 - X\beta\right\|_2^2 + \lambda \left\|\beta\right\|_2 \tag{3.13}$$

satisfies

$$\sum_{l=1}^{L} \hat{\beta}_l = 0 \tag{3.14}$$

and the same holds for other loss functions.

It follows that if $\hat{\mu}$ and $\hat{\beta}$ are solutions to the equation, then so are $\hat{\mu}+c1$ and $\hat{\beta}-c1$. This holds for X being an indicator matrix and from this it follows that $X \cdot c1 = c1$. However the norm, $|| \beta - c1 ||_2$ is minimized for $c = \hat{\beta}$. In the following lemma it is evident that if two intercepts are included in the model, one penalized and the other unpenalized, then the penalized intercept will be estimated to be zero. Thus the same fit can be achieved with a lower penalty by taking $\mu \leftarrow \mu + \tilde{\mu}$.

Lemma 2. *The optimization problem*

$$\arg\min_{\mu,\beta} \frac{1}{2} \|Y - \mu 1 - \tilde{\mu} 1 - ...\|_2^2 + \lambda \sqrt{\|\tilde{\mu}\|_2^2 + \|\beta\|_2^2} \tag{3.15}$$

has solution $\hat{\tilde{\mu}} = 0$ *for all* $\lambda > 0$. *The same result holds for other loss functions.*

This states that if two intercepts are included in the model, the penalized one will always be estimated to be zero. This is because the same fit can be achieved by taking only one of the intercepts μ.

It can be shown that,

$$\|\beta_{1:2}\|_2 = \sqrt{L_2 \|\tilde{\alpha_1}\|_2^2 + L_1 \|\tilde{\alpha_2}\|_2^2 + \|\alpha_{1:2}\|_2^2} \tag{3.16}$$

where $\tilde{\alpha_1}$,$\tilde{\alpha_2}$ and $\alpha_{1:2}$ satisfy constraints in Equations (3.9) and (3.10). For fixed levels i and j, the interaction term can be decomposed into

$$\beta_{1:2}^{ij} = \tilde{\alpha}_1^i + \tilde{\alpha}_2^i + \alpha_{1:2}^{ij}. \tag{3.17}$$

The $(L_1 L_2)$-vector $\beta_{1:2}$ can be written as

$$\beta_{1:2} = I_1 \tilde{\alpha}_1 + I_2 \tilde{\alpha}_2 + \alpha_{1:2}, \tag{3.18}$$

where I_1 is a $L_1L_2 \times L_1$ indicator matrix and I_2 is a $L_1L_2 \times L_2$ indicator matrix. And thus it can be shown from Equation 3.18 that the additive components are mutually orthogonal as follows,

$$\begin{aligned} \|\beta_{1:2}\|_2^2 &= \|I_1\tilde{\alpha}_1\|_2^2 + \|I_2\tilde{\alpha}_2\|_2^2 + \|\alpha_{1:2}\|_2^2 \\ &= L_2\,\|\tilde{\alpha}_1\|_2^2 + L_1\,\|\tilde{\alpha}_2\|_2^2 + \|\alpha_{1:2}\|_2^2 \end{aligned} \tag{3.19}$$

So the penalty in the group-lasso is equivalent to the penalty in the constrained overlapped group-lasso. And it remains to show that $X_{1:2}I_1 = X_1$ and $X_{1:2}I_2 = X_2$ it follows,

$$X_{1:2}\beta_{1:2} = X_{1:2}\big(I_1\tilde{\alpha}_1 + I_2\tilde{\alpha}_2 + \alpha_{1:2}\big) = X_1\tilde{\alpha}_1 + X_2\tilde{\alpha}_2 + X_{1:2}\alpha_{1:2} \tag{3.20}$$

Eventually the following theorem shows that the overlapped group-lasso reduces to a group-lasso and we can still obtain estimates that satisfy

Theorem 1. *Solving the constrained optimization problem (3.8) and (3.10) is equivalent to solving the unconstrained problem.*

$$\begin{aligned} \arg\min_{\mu,\beta} \frac{1}{2}\,\|Y - \mu 1 - X_1\beta_1 - X_2\beta_2 - X_{1:2}\beta_{1:2})\|_2^2 \\ +\lambda\big(\,\|\beta_1\|_2 + \|\beta_2\|_2 + \|\beta_{1:2}\|_2\,\big). \end{aligned} \tag{3.21}$$

Theorem 1 shows that we can use the group-lasso to obtain estimates that satisfy a strong hierarchy, without solving the overlapped group-lasso with constraints. And it is shown that the main effects and interactions can be extracted as,

$$\begin{aligned} \hat{\theta}_1 &= \hat{\beta}_1 + \hat{\tilde{\alpha}}_1, \\ \hat{\theta}_2 &= \hat{\beta}_2 + \hat{\tilde{\alpha}}_2, \\ \hat{\theta}_{1:2} &= \hat{\beta}_{1:2}. \end{aligned} \tag{3.22}$$

As a proof, it is shown how the group-lasso objective function can be transformed to an overlapped group-lasso with the appropriate constraints on the parameters. First, we begin

by rewriting the equation (3.8):

$$\arg\min_{\mu,\tilde{\mu},\alpha,\tilde{\alpha}} \frac{1}{2}\left\| Y - \mu 1 - X_1\alpha_1 - X_2\alpha_2 - \begin{bmatrix}1 X_1 X_2 X_{1:2}\end{bmatrix} \begin{bmatrix}\tilde{\mu}\\ \tilde{\alpha_1}\\ \tilde{\alpha_2}\\ \alpha_{1:2}\end{bmatrix} \right\|_2^2 + \lambda\left(\|\alpha_1\|_2 + \|\alpha_2\|_2 + \sqrt{L_1 L_2 \tilde{\mu}^2 + L_2 \|\tilde{\alpha_1}\|_2^2 + L_1 \|\tilde{\alpha_2}\|_2^2 + \|\alpha_{1:2}\|_2^2}\right). \tag{3.23}$$

$\hat{\tilde{\mu}}$ will be estimated to be equal to zero, from Lemma 2. Thus the solution did not change. The first two constraints in (3.9) are shown in Lemma 1 to be satisfied by the estimated main effects $\hat{\beta}_1$ and $\hat{\beta}_2$. And we have shown that,

$$\|\beta_{1:2}\|_2^2 = L_2 \|\tilde{\alpha_1}\|_2^2 + L_1 \|\tilde{\alpha_2}\|_2^2 + \|\alpha_{1:2}\|_2^2, \tag{3.24}$$

where $\tilde{\alpha_1}$, $\tilde{\alpha_2}$ and $\alpha_{1:2}$ satisfy the constraints (3.9) and (3.10) of the initial objective function. Thus the penalty in the group-lasso problem is equivalent to the penalty in the constrained overlapped group-lasso. And it follows that the loss functions in both problems are also the same.

3.4.4 Interaction Between Two Continuous Variables

Interactions terms can be appropriately represented as

- $X_1 * X_2 = X_{1:2}$ for categorical variables,
- $X * [1 \quad Z] = [X \quad (X * Z)]$ for one categorical variable and one continuous variable.

For continuous variables we define Z_1 and Z_2 to be two continuous variables. The appropriate form of the interaction term is given by

$$\begin{aligned} Z_{1:2} &= [1 \quad Z_1] * [1 \quad Z_2] \\ &= [1 \quad Z_1 \quad Z_2 \quad (Z_1 * Z_2)]. \end{aligned} \tag{3.25}$$

Then it follows, that the linear interaction for Z_1 and Z_2 is given by

$$\mathbb{E}[Y|Z_1 = z_1, Z_2 = z_2] = \theta_1 z_1 + \theta_2 z_2 + \theta_{1:2} z_1 z_2. \tag{3.26}$$

Consequently, the overlapped group-lasso would be,

$$\begin{aligned}\arg\min_{\mu,\tilde{\alpha},\alpha,\tilde{\alpha}} \frac{1}{2} \left\| Y - \mu \cdot 1 - Z_1\alpha_1 - Z_2\alpha_2 - \begin{bmatrix} Z_1 & Z_2 & (Z_1 * Z_2) \end{bmatrix} \begin{bmatrix} \tilde{\mu} \\ \tilde{\alpha_1} \\ \tilde{\alpha_2} \\ \tilde{\alpha_{1:2}} \end{bmatrix} \right\|_2^2 \\ + \lambda \left(\|\alpha_1\|_2 + \|\alpha_2\|_2 + \sqrt{\|\tilde{\alpha}_1\|_2^2 + \|\tilde{\alpha}_2\|_2^2 + \|\alpha_{1:2}\|_2^2} \right),\end{aligned} \tag{3.27}$$

which is equivalent to,

$$\begin{aligned}\arg\min_{\mu,\beta} \frac{1}{2} \|Y - \mu \cdot 1 - Z_1\beta_1 - Z_2\beta_2 - ([1 \;\; Z_1][1 \;\; Z_2])\beta_{1:2}\|_2^2 \\ + \lambda(\|\beta_1\|_2 + \|\beta_2\|_2 + \|\beta_{1:2}\|_2),\end{aligned} \tag{3.28}$$

replacing α's by β's.

3.4.5 Interaction Between Two Categorical Variables

To complete the derivation of interaction we will also present the case of an interaction between two categorical variables. Note, that for categorical data, interactions are taken at each level of the variable. For example, if we have an interaction between two categorical variables with levels n and m, respectively, we will have a matrix of interaction coefficients of dimensions $(n \times m)$. For simplicity, we denote the two categorical variables X_1 and X_2. The appropriate form of the interaction term is given by the product representation,

$$\begin{aligned}X_{1:2} &= [1 \quad X_1] * [1 \quad X_2] \\ &= [1 \quad X_1 \quad X_2 \quad (X_1 * X_2)],\end{aligned} \tag{3.29}$$

assuming that each categorical variable has two levels. Then it follows, that the linear interaction for X_1 and X_2 is given by

$$\begin{aligned}\mathbb{E}[Y|X_1 = x_1, X_2 = x_2] = \theta_{11}x_{11} + \theta_{12}x_{12} + \theta_{21}x_{21} + \theta_{22}x_{22} \\ + \theta_{11:21}x_{11}x_{21} + \theta_{11:22}x_{11}x_{22} + \theta_{12:21}x_{12}x_{21} + \theta_{12:22}x_{12}x_{22}.\end{aligned} \tag{3.30}$$

This makes it evident how fast the number of interaction possibilities can increase. Consequently, the overlapped group-lasso would be, again assuming one level for simplicity,

$$\begin{aligned}\arg\min_{\mu,\tilde{\alpha},\alpha,\tilde{\alpha}} \frac{1}{2} \left\| Y - \mu \cdot 1 - X_1\alpha_1 - X_2\alpha_2 - \begin{bmatrix} X_1 & X_2 & (X_1 * X_2) \end{bmatrix} \begin{bmatrix} \tilde{\mu} \\ \tilde{\alpha_1} \\ \tilde{\alpha_2} \\ \tilde{\alpha_{1:2}} \end{bmatrix} \right\|_2^2 \\ + \lambda \left(\|\alpha_1\|_2 + \|\alpha_2\|_2 + \sqrt{\|\tilde{\alpha}_1\|_2^2 + \|\tilde{\alpha}_2\|_2^2 + \|\alpha_{1:2}\|_2^2} \right),\end{aligned} \tag{3.31}$$

which is equivalent to,

$$\begin{aligned}\arg\min_{\mu,\beta} \frac{1}{2} \|Y - \mu \cdot -1X_1\beta_1 - X_2\beta_2 - ([1 \;\; X_1][1 \;\; X_2])\beta_{1:2}\|_2^2 \\ + \lambda(\|\beta_1\|_2 + \|\beta_2\|_2 + \|\beta_{1:2}\|_2),\end{aligned} \tag{3.32}$$

again replacing α's by β's.

3.4.6 Interaction Between a Categorical Variable and a Continuous Variable

We consider having a categorical variable X with L levels and a continuous variable Z. Let the mean $\mu_i = \mathbb{E}[Y|X = i, Z = z]$. Modeling results can fall into one of the following 4 cases:

- $\mu_{ij} = \mu$ (no main effects, no interactions),
- $\mu_{ij} = \mu + \theta_1^i$ (one main effect X_1 or Z_1),
- $\mu_{ij} = \mu + \theta_1^i + \theta_2 z$ (two main effects),
- $\mu = \mu + \theta_1^i + \theta_2 z + \theta_{1:2}^i z$ (main effects and interaction).

An overlapped group-lasso objective function with X being the indicator matrix repre-

sentation for categorical variables and Z for continuous variables is given by,

$$\begin{aligned}\arg\min_{\mu,\alpha,\tilde{\alpha}} \frac{1}{2}\left\| Y-\mu\cdot 1-X\alpha_1-Z\alpha_2-[X \quad Z \quad (X*Z)]\begin{bmatrix}\tilde{\alpha_1}\\ \tilde{\alpha_2}\\ \tilde{\alpha_{1:2}}\end{bmatrix}\right\|_2^2 \\ +\lambda(||\ \alpha_1\ ||_2+||\ \alpha_2\ ||_2+\sqrt{||\ \tilde{\alpha_1}\ ||_2^2+L\ ||\ \tilde{\alpha_2}\ ||_2^2+||\ \alpha_{1:2}\ ||_2^2}),\end{aligned} \tag{3.33}$$

subject to

$$\sum_{i=1}^{L}\alpha_1^i=0,\quad \sum_{i=1}^{L}\tilde{\alpha}_1^i=0,\quad \sum_{i=1}^{L}\alpha_{1:2}^i=0, \tag{3.34}$$

and the constraints $\sum_{i=1}^{L}\theta_1^i=0$ and $\sum_{i=1}^{L}\theta_{1:2}^i=0$. By solving this objective function under the constraints, the estimated interactions obtained satisfy a strong hierarchy. This is obtained through the nature of the square root term in the penalty. The main effects and interactions are given by

$$\begin{aligned}\hat{\theta}_1&=\hat{\alpha}_1+\hat{\tilde{\alpha}}_1,\\ \hat{\theta}_2&=\hat{\alpha}_2+\hat{\tilde{\alpha}}_2,\\ \hat{\theta}_{1:2}&=\hat{\alpha}_{1:2}.\end{aligned} \tag{3.35}$$

Theorem 2. *Solving the parameter in* (3.34) *under the constraints in (3.35) is equivalent to solving the following problem,*

$$\begin{aligned}\arg\min_{\mu,\beta}\frac{1}{2}\|Y-\mu\cdot 1-X\beta_1-Z\beta_2-(X*[1 \quad Z])\beta_{1:2}\|_2^2 \\ +\lambda(||\ \beta_1\ ||_2+||\ \beta_2\ ||_2+||\ \beta_{1:2}\ ||_2).\end{aligned} \tag{3.36}$$

Additional parameters $\tilde{\mu}$ are introduced to the objective function of the overlapped group-lasso.

$$\begin{aligned}\arg\min_{\mu,\mu,\tilde{\alpha},\tilde{\alpha}}\frac{1}{2}\left\| Y-\mu\cdot 1-X\alpha_1-Z\alpha_2-[1\ X\ Z\ (X*Z)]\begin{bmatrix}\tilde{\mu}\\ \tilde{\alpha_1}\\ \tilde{\alpha_2}\\ \tilde{\alpha_{1:2}}\end{bmatrix}\right\|_2^2 \\ +\lambda\left(\|\alpha_1\|_2+\|\alpha_2\|_2+\sqrt{\|\tilde{\alpha}_1\|_2^2+L\,\|\tilde{\alpha}_2\|_2^2+\|\alpha_{1:2}\|_2^2}\right).\end{aligned} \tag{3.37}$$

Adding the parameter $\hat{\tilde{\mu}}$ does not change the solution since it will be equal to zero as we have showed in Lemma 2. The interaction parameter $\|\beta_{1:2}\|_2^2$ can be decomposed as follows:

$$\|\beta_{1:2}\|_2^2 = L\,\|\tilde{\mu}\|_2^2 + \|\tilde{\alpha}_1\|_2^2 + L\,\|\tilde{\alpha}_2\|_2^2 + \|\alpha_{1:2}\|_2^2\,. \tag{3.38}$$

This shows how the penalties in both problems are equivalent. Note that $\|\tilde{\alpha}_1\|$ is an $(L \times 1)$-vector and thus $\sum_{i=1}^{L} \tilde{\alpha_2^i} = 0$ and similarly for $\alpha_{1:2}$. It follows that we can write the interaction parameter as,

$$\beta_{1:2} = \begin{bmatrix} \tilde{\mu} \cdot 1_{L\times 1} \\ \tilde{\alpha_2} \cdot 1_{L\times 1} \end{bmatrix} + \begin{bmatrix} \tilde{\alpha_1} \\ \alpha_{1:2} \end{bmatrix}. \tag{3.39}$$

It can also be shown by direct computation that the loss functions are equivalent,

$$\begin{aligned} \big(X * [1\ Z]\big)\beta_{1:2} &= \big[X\ (X * Z)\big]\beta_{1:2} \\ &= \big[X\ (X * Z)\big]\left(\begin{bmatrix} \tilde{\mu} \cdot 1_{L\times 1} \\ \tilde{\alpha_2} \cdot 1_{L\times 1} \end{bmatrix} + \begin{bmatrix} \tilde{\alpha_1} \\ \alpha_{1:2} \end{bmatrix} \right) \\ &= \tilde{\mu} \cdot 1 + X\tilde{\alpha_1} + Z\tilde{\alpha_2} + (X * Z)\alpha_{1:2}. \end{aligned} \tag{3.40}$$

Thus parameterizing the interaction as $X * \big[1\ Z\big]$ allows us to accommodate interactions between continuous and categorical variables.

3.5 Modeling Hierarchical Interactions With Boosted Trees and Adaptive Screening

As explained in the previous sections we conclude that gradient boosting are effective approaches to building ensembles of weak learners such as decision trees. Boosted trees are able to model nonlinear effects and high-order interactions that linear models are not able to capture automatically. A simple example, is a depth 2 tree, which represents an interaction between the variables involved in the two splits. This suggests that boosting with depth-2 trees is a way of building a first-order interaction model. Note that the interactions are hierarchical, because in finding the optimal first split, the boosting algorithm is looking

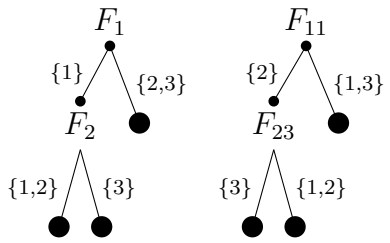

Figure 3.1: Screening with Boosted Trees

for the best main effect. The second split is then made, conditional on the first split. If we boost with T trees, then we end up with a model that has at most T interaction pairs. Given p variables the possible pairwise interaction space is $\binom{p}{2}$. However, if factors have multiple levels, the model takes interactions at each level of the factor and thus the possible interaction space would be the sum of all levels choose 2 instead. The following diagram gives an illustration of the boosting iterations with categorical variables.

In the first tree in Figure 3.1 , levels 2 and 3 of F_1 are not involved in the interaction with F_2. In this example the interaction is taken with respect to level 1. Therefore, for categorical variables, each tree in the boosted model does not represent an interaction among all the levels of the two variables, but only among a subset of the levels. To obtain the full interaction structure, a fully split tree could be used, but this approach is not developed for two reasons. First, boosting is an iterative procedure and is quite slow even for moderately sized problems and using a fully split tree will slower runtime. Second, for categorical variables with many levels, it is expected that the interactions only occur among a few of the levels. As a consequence, if this is true, then a complete interaction that is weak for every combination of levels might be selected over a strong partial interaction. But it is the strong partial interaction that we are interested in, since the overall objective is to find solutions that are sparse.

Boosting is feasible because it is a greedy algorithm which means that it attempts to make the locally optimal solution at each stage with the hope of finding a global optimum. For example, if we have p variables, an exhaustive search of all variable combinations involves $O(p^2)$ variables, whereas boosting operates with $O(p)$. A boosted model is used as a screening

device for interaction candidates. To do that, the set of all unique interactions is taken from the collection of trees. For example, as illustrated above, we would add $F_{1:2}$ and $F_{11:23}$ to our candidate set of interactions.

Boosting as a screening procedure is quite cumbersome because it involves selecting tuning parameters, the amount of shrinkage and the proper number of trees. Good results are obtained when lowering the penalty value and increasing the number of trees. However, this is at the expense of computational speed. The efficient method used for the screening approach is based on the idea of computing inner products that can be integrated strong rules for discarding predictors in lasso-type problems. This is a method, that basically discards large number of inactive variables that are most likely redundant and should not be added to the active set of variables.

The advantage of strong rules is the speed of convergence of the algorithm since a smaller set "the strong set" of variables that are more likely to be nonzero is used instead of using all candidates. Karush Kuhn Tucker (KKT) conditions are checked after the algorithm has converged to ensure that all discarded variables are actually equal to zero and we picked the right ones, since the strong rules can mistakenly discard active predictors. Otherwise, those variables that do not satisfy the KKT conditions then have to be added to the current set of nonzero variables, and we fit on the combined expanded set. However, from multiple experiments, this does not happen often. The strong rules calculation for group-lasso has to be conducted by computing $s_i = || X_i^T(Y - \hat{Y}) ||_2$ for every group of variables X_i. Then the strong rules filter is applied, where a group i is discarded if the test $s_i < 2\lambda_{\text{current}} - \lambda_{\text{previous}}$ is satisfied (Tibshirani et al., 2012). The filter is applied through the difference in the sequence of the λ penalty values. If it is feasible for all $p + \binom{p}{2}$ groups, no screening needs to be done and the group-lasso is fitted on the groups that passed the strong rules filter. However, if it is not feasible we approximate by screening only the groups that correspond to the main effects. Then, all pairwise interactions are taken for the variables that passed the screen.

The KKT conditions for group i would be,

$$\begin{aligned} s_i < \lambda \qquad for \qquad \hat{\beta} = 0, \\ s_i = \lambda \qquad for \qquad \hat{\beta} \neq 0, \end{aligned} \tag{3.41}$$

where s_i is computed for the strong rules from checking the KKT conditions for the solutions at the previous λ, thus, integrating screening with the strong rules. A candidate set for the group-lasso is composed of a specific number of variables with the highest score and pairwise interactions of all the variables. The number of variables is chosen in a way as to make the computation feasible for the group-lasso.

To compute the fit of λ_{k+1} from λ_k we need to obtain the residuals. First we assume that we have fitted λ. Let $r_{\lambda_k} = Y - \hat{Y}_{\lambda_k}$ denote the residuals of the current fit. When the KKT conditions are checked for the λ_k fit, the variable score $s_i = || X_i^T r_{\lambda_k} ||_2$ is computed. Then the group-lasso is fitted on the candidate set and the procedure is repeated with the new residual $r_{\lambda_{k+1}}$. This screen can be easily computed since it is based on inner products between each predictor and the outcome, that guarantee a coefficient will be zero in the solution vector (Tibshirani et al., 2012). The screen can integrate well with the strong rules by reusing inner products computed from the fit for a previous λ. Thus a reduction in the number of variables that need to be entered into the optimization.

3.6 Modeling Interactions in Property and Casualty Insurance Data

The common method for detecting interactions used by actuaries is to plot the data in three dimensions to visualize the relations in multiple regression and the existence of interactions. The existence of variable interactions becomes evident when we can see a varying interaction surface with respect to the response variable Y. This indicates that the effect of one independent variable X_1 on the response changes depending on the level of another variable X_2, for example in Figure 3.2. However, figures cannot tell you whether a pattern is significant

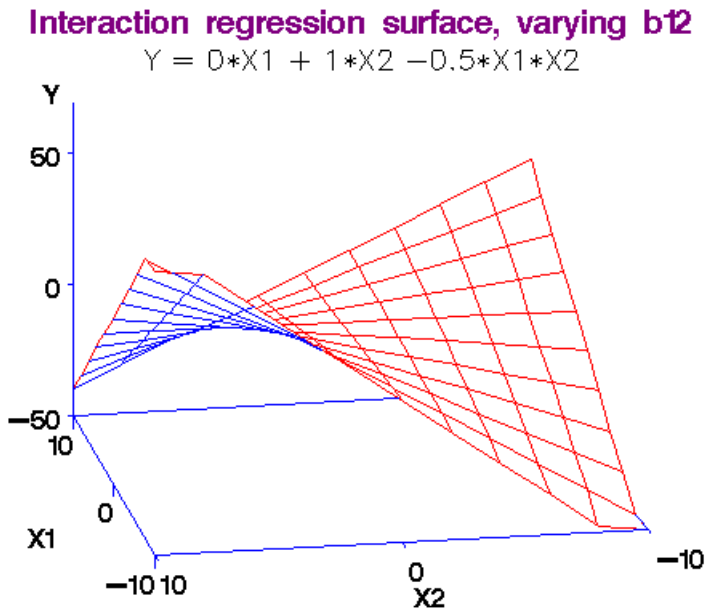

Figure 3.2: Interaction Regression Surface

or not and when results of a statistical test are needed.

Traditionally, another frequent practice is the analysis of main effects of one factor at the expected values of the other factors. However, this leads to mixing both main and interaction effects. Another common method is backward elimination and factorial analysis. This is called post hoc analysis of interaction in factorial experiments and it depends mainly on ANOVA results and significance of covariates after a model is fit. For example in the Emblem software this is done by checking the significance of each possible combination of factors. The first problem is that this can be time consuming and second is that it could be inaccurate for large numbers of variables. These shortcomings go back to the limitations of linear models and their inability to capture interactions.

Chapter 4

Actuarial Applications

4.1 Introduction

The high dimensionality of data poses significant challenges in building interpretable models. Therefore, regularization has been commonly employed to obtain more stable and interpretable models. The model introduced learns pairwise interactions in Poisson regression and gamma regression models, also satisfying the strong hierarchy property; for a nonzero estimated interaction both its associated main effects are included in the model.

The model accommodates continuous and categorical variables with an arbitrary number of levels. The lasso framework allows for constraints on the main effects and their corresponding parameters. The resulting fit is parsimonious and interpretable while having the ability to handle large numbers of variables and selecting from them. This is fitted using the *glinternet*2 package in R, which stands for "group-lasso interaction network". We extend the *glinternet* package to include Poisson and gamma families to model the frequency and severity of insurance claims, respectively. Including interactions remains a challenge for actuaries since most of the models used are in the linear framework. For any response variable and given explanatory variables we expect to find interactions, specially if the model can not be explained by additive functions of the variables (Hastie and Lim, 2015).

The model introduced is solved by dividing the solution path into two phases, first the

screening phase, when a candidate set of main effects and interactions is found. The second phase follows with variable selection and model fitting on the candidate set using group-lasso for a grid of values for the λ regularization parameter. The model starts by fitting $\lambda = \lambda_{\max}$, for which no variables are included, and then decreases the value of λ to allow variables to enter the model. For the screening procedure, two methods are used. First one is the boosting with depth-2 trees which enforces hierarchy when an interaction is selected. Here, an interaction cannot be chosen until a split has been made on one of its two associated main effects. The second method is an adaptive screening procedure that is based on strong rules for discarding predictors in lasso-type problems. It has been also showed that group-lasso enforces a strong hierarchy in the solution. For claims data specifically, we expect a number of interactions since variables represent individual's characteristics. The model proposed introduces a family of methods that incorporate interactions automatically.

4.2 Regularized Claim Model

The purpose of using a regularized model for claim predictions is to perform variable selection and interaction detection. Modern data is usually high-dimensional so conducting variable selection on data sets with large number of covariates is too cumbersome and inconclusive. The group-lasso proposes a solution to this problem. To model insurance data and fit frequency-severity models the lasso is derived for Poisson and gamma families.

4.2.1 Regularized Poisson Model

For a response variable Y taking integer values, i.e. count data (claim frequency), we consider a Poisson regression, where the conditional probability distribution Y_i, given $X_i = x \sim$ Poisson($\mu(x)$). Poisson regression is used for count data under the assumption of Poisson errors. The log link function, which turns out to be the canonical link, is used to model its

positive mean in a log scale as follows,

$$\log(\mu(x)) = \sum_{j=0}^{p-1} \beta_j x_j = \eta. \tag{4.1}$$

The negative log-likelihood and loss function for observations X_i and response Y_i equals

$$-\sum_{i=1}^{n} \mathcal{L}(Y_i, X_i; \beta) = \sum_{i=1}^{n} -Y_i(X_i^\top \beta) + \exp(X_i^\top \beta). \tag{4.2}$$

The first term is linear and hence convex in $\beta = (\beta_0, \ldots, \beta_{p-1})$. The second term is a composition of a convex and a linear function and hence convex in β, and since the sum of convex functions is convex, the loss function is convex in β. The lasso ℓ_1 regularization for Poisson GLM is given in equation (2.3) to optimize the penalized log-likelihood,

$$\hat{\beta} = \arg\min_{\beta} \mathcal{L}(Y, X, \beta) + \lambda \left\|\beta\right\|_1 \tag{4.3}$$

where $\mathcal{L}(Y, X; \beta)$ is the negative log-likelihood. In general, when dealing with data in Poisson models, the use of an offset is recommended. This is because, counts are often based on different exposure times, such as the time an insurance policy is in effect. Even though insurance companies usually sell policies with a term of one year, the actual in force time of the policy can be less for various reasons. Therefore, the Poisson rate is relative to a unit exposure time, so if an observation is exposed to a specific number or fraction of units of time called "exposure" and given as t_i for each observation, then the expected count would be $\frac{\mu}{t}$, and the log mean would be explained by the GLM as:

$$\log\left(\frac{\mu(x)}{t}\right) = \sum_{i=1}^{n} X_i^\top \beta. \tag{4.4}$$

The offset is a vector of length equal to the number of observations n included in the linear predictor:

$$\begin{aligned} \log(\mu(x)) - \log(t) &= \sum_{i=1}^{n} X_i^\top \beta \\ \log(\mu(x)) &= \log(t) + \sum_{i=1}^{n} X_i^\top \beta. \end{aligned} \tag{4.5}$$

The claim frequency is defined as the number of claims divided by the policy exposure time, i.e. the average number of claims per unit time. In R, the code to adjust for offset in GLM is given below.

Listing 4.1: R code for Poisson GLM

```
glm(y ~ offset(log(exposure)) + x, family=poisson(link=log) )
```

The same paradigm is used in the *glinternet*2 model to offset count data and account for policy exposure time.

4.2.2 Regularized Gamma Model

For a response variable Y taking continuous values (claim severity), we consider a gamma GLM. The claim severity is the total claim size divided by the number of claims to give the average size per claim. To model the severity the gamma model is assumed where the conditional probability distribution of Y_i given $X_i = x \sim$ Gamma (α, ν). The log link function is used even though it is not the canonical log link defined by the exponential dispersion family. The reason for that is that we want to build a multiplicative model so a log-link function is appropriate as follows,

$$\log(\mu(x)) = \sum_{i=1}^{n} X_i^\top \beta = \eta. \tag{4.6}$$

For the gamma GLM the inverse link is the canonical link here but in the actuarial context the log link is the most common choice. The probability density function of the gamma distribution is given by,

$$f(y, x, \alpha, \nu) = \sum_{i=1}^{n} \frac{1}{\Gamma(\alpha)} x_i^{(\alpha-1)} \nu^\alpha e^{x_i/\nu}, \tag{4.7}$$

and the corresponding negative log-likelihood equals,

$$-\sum_{i=1}^{n} \mathcal{L}(Y_i, X_i; \beta) = -(\alpha - 1) \sum_{i=1}^{n} \log X_i + \frac{X_i}{\nu} + \alpha \log \nu + \log \Gamma(\alpha). \tag{4.8}$$

The lasso ℓ_1 regularization for a gamma GLM is given by,

$$\hat{\beta} = \arg\min_{\beta} \mathcal{L}(Y, X; \beta) + \lambda \left\| \beta \right\|_1. \tag{4.9}$$

The parameters α and ν of the gamma can be determined by matching moments from the data as $\alpha = \frac{mean(Y)^2}{var(Y)}$ and $\nu = \frac{var(Y)}{mean(Y)}$. The gamma model is then used to fit claim severities.

4.2.3 Algorithm and Optimization

The algorithm used in *glinternet*2 to solve the group-lasso optimization problem is general, not specifically written for learning interactions. However, we re-use it here due to the speed of convergence. Let Y denote the vector of n of observed responses and $X = [x_1, x_2, \ldots, x_p]$ denote the generic feature matrix with p columns. The fast iterative soft thresholding (FISTA) method (Beck and Teboulle, 2009) is an approach to solve the lasso estimation problem. Since the group-lasso is a general version of the lasso, the FISTA can be adapted for group-lasso with some small changes. The FISTA is basically a generalized gradient method with a first order method of Nesterov style acceleration. The algorithm does not change when going from squared loss to logistic loss and further it can be used for other losses. The only condition is that the objective function is convex and differentiable. The gradient computation and parameter updates can be parallelized and can take advantage of adaptive momentum restarts which is often observed with accelerated gradient methods. These demonstrate that adaptively restarting the momentum factor, based on gradient condition, can speed up the convergence rate of FISTA. The logic behind is, that the momentum should be reset to zero whenever the gradient at the current step and the momentum point are in different directions.

The FISTA algorithm is presented, where at each iteration we take a step of size s in the direction of the gradient to solve the majorization minimization scheme given as:

$$M(\beta) = \mathcal{L}(Y, X; \beta_0) + (\beta - \beta_0)^\top g(\beta_0) + \frac{1}{2s} \mid\mid \beta - \beta_0 \mid\mid_2^2 \lambda \sum_{j=1}^{p-1} \|\beta_i\|_2 \,. \tag{4.10}$$

Here, the $g(\beta_0)$ is the gradient (derivative) of the negative log-likelihood $\mathcal{L}(Y, X; \beta)$ evaluated at β_0. Optimally, the step size s is chosen to be large for the start and then backtracked until the condition is satisfied. The step size should be carefully chosen since there is always the transition between convergence and divergence. Too large a step size can cause the algorithm to keep iterating infinitely i.e diverge, while reducing it too much can cause the algorithm to take so much time or to never reach an optimal value. The common solution to this problem is backtracking, which was introduced by (Becker and Grant, 2011) and is

used in our application to adaptively initialize the step size as follows:

$$s = \frac{\left\|\beta^{(k)} - \beta^{(k-1)}\right\|_2}{\|g_k - g_{k-1}\|_2}. \tag{4.11}$$

This defines how the step size changes at each iteration from the initialized value of s using backtracking. The Table 4.1 gives the algorithm for the optimization problem. The

Algorithm 1: FISTA with adaptive restart
input: Initial Parameter $\beta^{(0)}$, matrix of features X, responses Y, regularization parameter λ, and step size s
output: $\hat{\beta}$
Initialize $x^{(0)} = \beta^{(0)}$ and $\rho = 1$.
for k = 0, 1, ..., do
$g^{(k)} = -X^T(Y - X\beta^{(k)})$;
$x^{(k+1)} = (1 - \frac{s\lambda}{\|\beta^{(k)} - sg^{(k)}\|_2})(\beta^{(k)} - sg^{(k)})$;
if $(\beta^{(k)} - x^{(k+1)})^T(x^{(k+1)} - x^{(k)}) > 0$ then $\rho_k = 1$
$\rho_{k+1} = (1 + \sqrt{1 + 4\rho_k^2})/2$;
$\beta^{k+1} = x^{k+1} + \frac{\rho_k - 1}{\rho_{k+1}}(x^{(k+1) - x^{(k)}})$;
end

Table 4.1: Algorithm Steps to Solve the Model Parameter Estimation Problem

solution is achieved by using the normal generalized gradient known as FISTA where the soft-thresholding operator is given as:

$$[S_\lambda(X)]_i = \left\{ \begin{array}{ccc} & X_i - \lambda & if\ X_i > \lambda \\ 0 & & if\ -\lambda \geq X_i \leq \lambda \\ & X_i + \lambda & if\ X_i < -\lambda \end{array} \right\} \tag{4.12}$$

And the the model conducts K-fold cross validation by partitioning the training set into K subsamples, where one subsample is used as a validation set for testing the model fitted by the remaining K-1 subsamples. This process is repeated K times and each subsample is

used only once for validation. The K results are averaged to produce the estimate of the model parameters.

4.3 Simulation Study with Group-lasso Interaction Network

A simulated study was conducted to test how well *glinternet* retrieves interactions. A data set was simulated with 10 variables which consist of 7 continuous and 3 categorical variables with different levels. The response was simulated with the 10 variables, 10 interactions and some random noise. Then the model was fitted and using 10-fold cross validation the best model was chosen to minimize the cross validation error. Here, is a subset of the model fit giving the grid of 50 λ values and the corresponding number of variables and interactions captured:

Fit	Lambda	ObjValue	Categorical	Continuous	CatCat	ContCont	CatCont
1	3.07e-03	4.310	0	0	0	0	0
2	2.80e-03	4.300	0	3	0	0	0
3	2.54e-03	4.270	0	4	0	0	0
.	..	..	..	..	...	..	...
6	1.92e-03	4.080	0	5	0	1	0
.	..	..	..	..	...	..	...
48	3.71e-05	0.660	1	3	1	4	5
49	3.37e-05	0.646	1	3	1	4	5
50	3.07e-05	0.633	1	3	1	4	5

Table 4.2: Example of the Glinternet Output

Table shows the number and type of coefficient captured at each λ value

Running a 10-fold cross validation with errors, shown in Figure 4.1, reveals that the

model with the lowest λ value with all 10 variables and 10 interactions is the optimal one. This trend is only desirable here since the purpose of the simulation study is to check how the model retrieves main effects and interactions. However, for real data a saturated model with all the variables will be over-fitting. Specially, if the data set include large number of covariates. Therefore, an opposite error trend will be observed for real data. Examples on real data sets will be shown in the next sections.

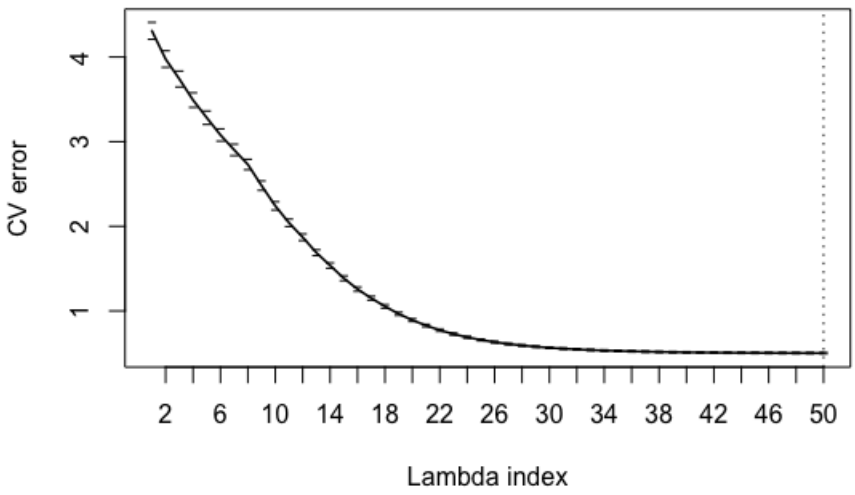

Figure 4.1: Cross validation Error for Simulation Study

After checking the false discovery rate, it turned out that the model retrieved the same interaction with which the response has been simulated and the false discovery rate is therefore 0. The model also gives the order at which each variable has been captured since it starts with a penalty high enough to force all coefficients to zero. Looking more closely at the coefficients, it is evident that each level of the categorical variables has a coefficient value and each corresponding interaction effect has a coefficient. Here is a subset showing the output of the model coefficients:

In Table 4.2, the output of the model is illustrated. The number of the categorical and continuous variables included in each fit is given respectively in order of addition to the

model as the value of the penalty decreases. Besides the main effects, the output shows the number and type of interaction included at each penalty value. In Tables 4.8 and 4.9, in the Appendix, the coefficients of the main effects and interaction effects are illustrated for the fit with the lowest penalty.

The same study was repeated for the *glinternet*2 for a gamma distribution. The algo-

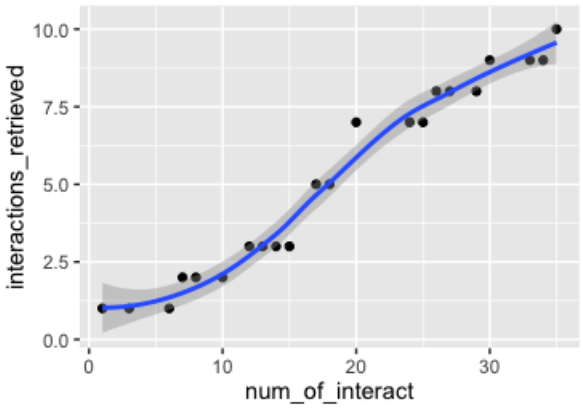

Figure 4.2: Discovery Rate in Glinternet2

rithm also retrieved the 10 main effects and the 10 interactions that were injected to simulate the response. Figure 4.2 shows how many interactions are found and how many of these were the ones that the model was simulated with. Notice, the dots indicate how many additional interactions were found.

4.4 Example 1: Singapore Automobile Insurance

In this example the Singapore Automobile Insurance data set is examined using different techniques to compare results. The data is obtained from General Insurance Association of Singapore and is available through the University of Wisconsin. It can be found at the following website (http://instruction.bus.wisc.edu/jfrees/jfreesbooks/Regression%20Modeling/BookWebDec2010/data.html). The goal, is to understand how driver characteristics affect the Singapore accident experience with an emphasis on variables' interactions. It is important for pricing actuaries to understand these relationships

so that they can charge the right price for the risk they cover.

4.4.1 Data Description

The Singapore Automobile Insurance data set consists of 5 main features. In addition to these, other variables are created as combination of these main variables to get more signal from selected groups. The first variable is the "Vehicle Type" which indicates whether the vehicle insured is either automobile (A) or other (O). The second variable "Vehicle Age" is an integer and gives the age of the vehicle in years. The third variable is binary indicating the "Gender" of the insured driver, 1 for female and 0 for male. Forth, is the variable "Age" which gives the age of the principle driver of the vehicle in years. Finally, the variable, "NCD" stands for the no claim discount. This variable is treated as a categorical variable and it is based on the previous accident records of the policyholder. A higher discount indicates that the prior accident record was good, given in 6 levels at intervals of 10, from $0, \ldots, 50$.

The data contains other variables so that in total it consists of 14 variables. They were not

Observation	SexInsured	VehicleType	ClmCount	ExpWeights	NCD	DrivAge	VehAge
1	0	O	0	0.6680356	30	18	0
2	0	O	0	0.5667351	30	18	0
3	0	O	0	0.5037645	30	18	0
4	0	O	0	0.9144422	20	18	0

Table 4.3: Example of the first four observations of Singapore the data set:

shown since they are created to indicate specific groups of policy holders.

4.4.2 Modeling Data

The model is fitted using the *glinternet*2 function in R, which is an extension of the *glinternet* function that includes Poisson and gamma families to model claim frequency

and severity. We added these in order to be able to apply the model to insurance claims. The features of the *glinternet* are desirable because of its ability to do variable selection and automatic detection of interactions and then fit a generalized linear model.

To perform a statistical analysis and model evaluation, the data is split into training and testing set. The training set is used to fit the model and then the test set is used for model validation. Since the Singapore Automobile Insurance data set only includes claim counts, a frequency model but no severity, will be fitted using a Poisson distribution. The same data set will be also fitted on a Poisson GLM and GBM with Poisson losses for comparison.

4.4.3 Fitted Models and Empirical Results

The model is chosen optimally based on minimum cross validation error. A 10-fold cross validation is run on 20 different λ values. The *glinternet*2 model performs cross validation internally to carry out model selection based on the best number of main effects and interactions that decrease cross validation error. A gradient boosted model using the GBM package in R was used to fit a frequency model with a Poisson loss function in addition to a standard and a lasso GLM.

Results of the Singapore Automobile Insurance data set using a Poisson GLM are given in Table 4.4 illustrating all the coefficient values. Then a lasso GLM was used for the frequency model with a penalty grid of $\lambda = 0.00005, 0.0001, 0.001, 0.005, 0.01$, to find the optimal value and corresponding subset we run a cross validation to select the best variable subset. The model has optimally chosen the 5 main variables to be the best model subset out of 14 variables. Figure 4.3 gives a plot of the cross validation error grid, showing how the Poisson deviance changes with the number of variables included. It is evident that errors increase substantially when the number of variables decreases, but they remain constant for 5 to 13 variables. This basically means, that adding only 5 variables in the model gives the same signal as 13. There is sometimes a trade-off between adding variables and minimizing error. Ideally, we want to minimize the out-of-sample error and decrease the number of covariates in the model without loss of predictability. The two gains curves in Figures 4.4 and 4.5 show

Variable Name	Estimate	Std. Error	z-Value	P-Value
(Intercept)	-1.598e+00	6.224e-01	-2.567	0.0103 *
VehAge	1.555e-02	4.250e-02	0.366	0.7145
DriAge	4.588e-04	1.796e-02	0.026	0.9796
SexM	6.018e-02	1.630e-01	0.369	0.7121
SexU	1.145e+01	4.756e+02	0.024	0.9808
VehG	-1.189e+01	4.756e+02	-0.025	0.9801
VehM	-1.335e+01	4.756e+02	-0.028	0.9776
VehP	-1.103e+01	4.756e+02	-0.023	0.9815
VehQ	-1.137e+01	4.756e+02	-0.024	0.9809
VehS	-1.166e+01	4.756e+02	-0.025	0.9804
VehT	-2.344e+01	6.103e+02	-0.038	0.9694
VehW	-1.198e+01	4.756e+02	-0.025	0.9799
VehZ	-1.157e+01	4.756e+02	-0.024	0.9806
NCD	-1.674e-02	2.943e-03	-5.688	1.29e-08 ***

Table 4.4: Coefficients of the Poisson GLM Fit for 5 Variables

that both models are equally predictable. Clearly, the model selected the right variables to capture the same signal with less variables.

Fitting the frequency with a group-lasso interaction network reveals that adding interactions can improve model predictability, keeping a low number of variables in the model. The model is fitted with a grid of 20 λ values. As the value of λ decreases more variables are captured and consequently more interactions. A sample of the model output given in Table 4.5 shows the number and type of interactions captured at each λ value. In Tables 4.10 and 4.11, the main effect and interaction coefficients for the fit are shown. By looking closer into the coefficients, it is clear that the model satisfies strong hierarchy and if an interaction is present in the model, both of the main effects are present as well. The algorithm has retrieved a maximum of 9 interactions with the order of inclusion given in Table 4.6. As the penalty

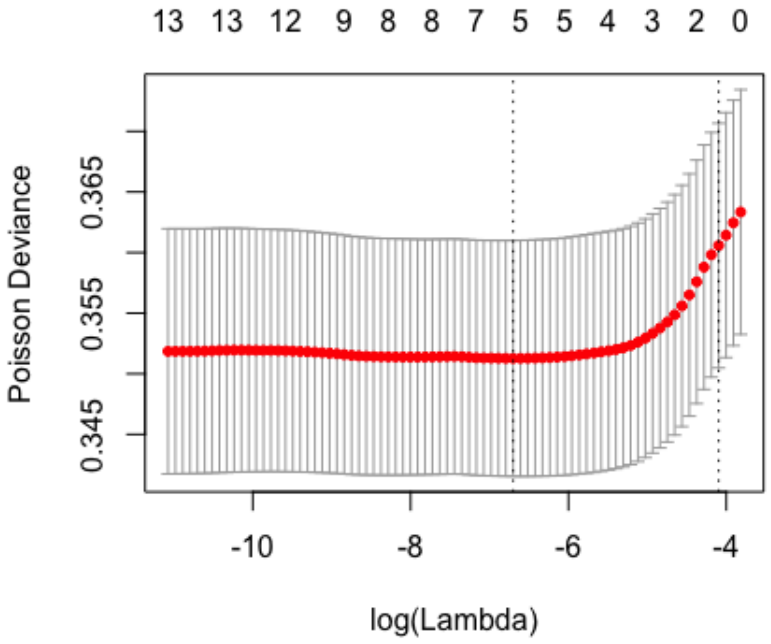

Figure 4.3: Cross Validation Error Plot for lasso Penalty Grid

Bottom: Penalty value; Top: Number of variables included; Left: Poisson Deviance

decreases, more interactions are included in the model. The most common interaction to test in auto insurance modeling is Gender × Driver Age and therefore it makes sense that it was the second one to be captured by the algorithm. The variable "No Claim Discount" indirectly indicates whether there has been a claim in the past or not. It is expected in a frequency model that the first interaction captured to be NCD × Driver Age. Table 4.10 in the Appendix gives the coefficients of the set with the lowest λ value and largest interaction set.

Using the Glinternet2 fit with 5 main effects and 9 interactions to predict out-of-sample predictions, it shows clearly in Figure 4.6 that predictions have improved compared to the previous models, the GLMs with 5 rather than 14 variables and the gradient boosting model. The improvement in predictability is due to the addition of interactions and the selection of the most significant main effects.

Running a gradient boosting model with 3000 tree splits and interactions depth 3 shows

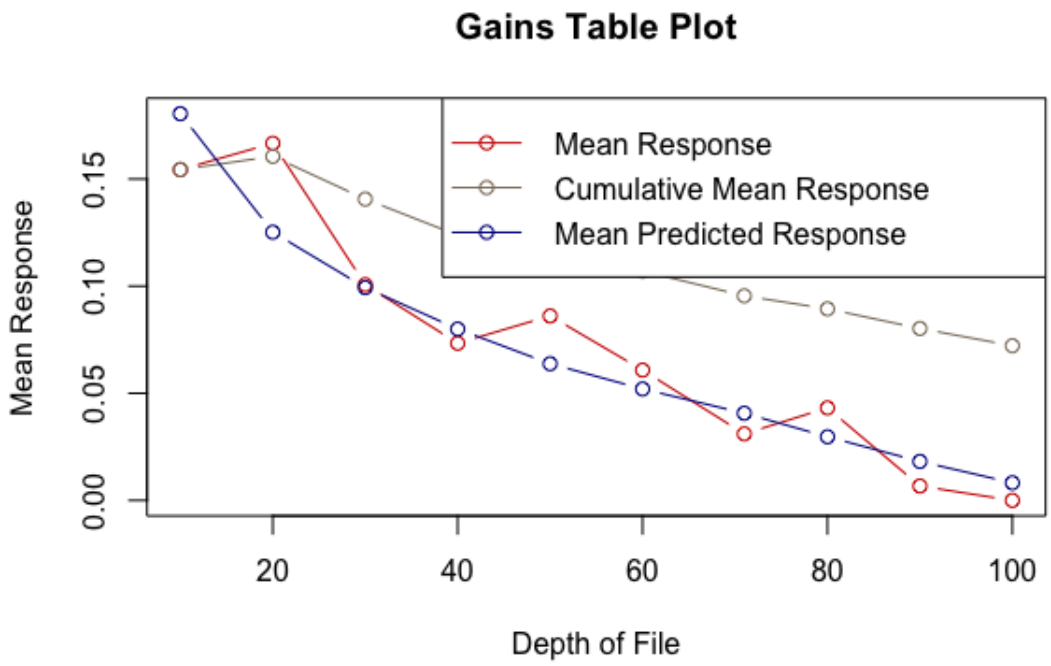

Figure 4.4: Gains of a lasso GLM with 5 Variables

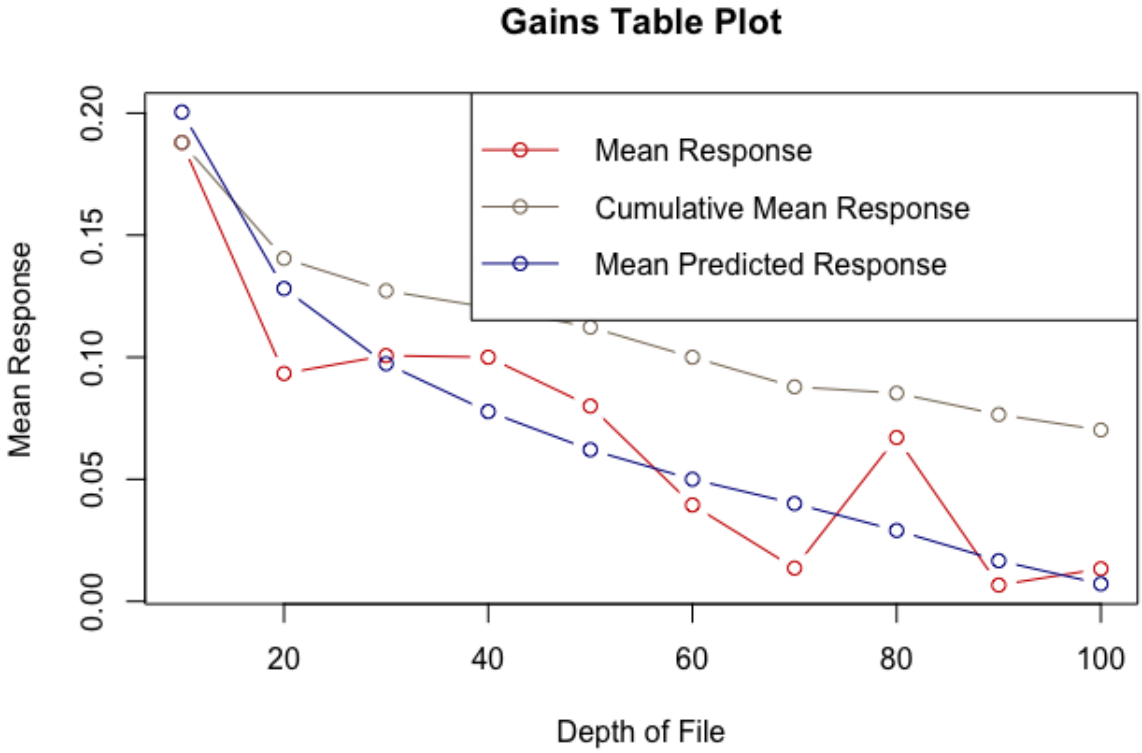

Figure 4.5: Gains of a GLM with 14 Variables

Fit	lambda	ObjValue	Categorical	Continous	CatCat	ContCont	CatCont
1	4.68e-04	-Inf	0	0	0	0	0
2	4.15e-04	3.84e+13	0	1	0	0	0
3	3.68e-04	3.84e+13	0	1	0	0	0
4	3.26e-04	3.84e+13	0	1	0	0	1
5	2.89e-04	3.84e+13	0	1	0	0	2
6	2.56e-04	3.84e+13	0	1	0	1	2
7	2.26e-04	3.84e+13	0	1	0	1	3
..	.....	.....	.....	.....	.....	.....	....
14	9.69e-05	3.84e+13	0	1	1	1	3
15	8.59e-05	3.84e+13	1	1	2	1	3
16	7.61e-05	3.84e+13	1	1	2	1	3
17	6.74e-05	3.84e+13	1	1	2	1	4
18	5.97e-05	3.84e+13	2	1	3	1	4
19	5.29e-05	3.84e+13	2	1	3	1	4
20	4.68e-05	3.84e+13	2	1	3	1	5

Table 4.5: Glinternet2 Poisson Fit for 5 Variables and Selected Interaction

that the most significant variable is Vehicle Type with relative influence of 24.83%, followed by the 4 other main variables. The 14 variables have a non-zero influence. Looking at the gains curve Figure 4.7 for GBM, it is evident that only 4 prediction groups are recognizable. This means, that the model only predicts observations to fall in 4 distinct groups and cannot differentiate further between observations. The model fails to segment further between observations. Generally, tree based model can fail to capture linear relationships. These limitations are compensated for by training a tree based model with tuned parameters such as more tree splits and more interaction depths. For this example, this is not applicable since the data contains only a few thousand observations. It is more appropriate for bigger data sets, where the algorithm can learn by training over millions of observations. Eventually,

Glinternet2: Interactions Detected
1. No Claim Discount × Driver Age
2. Gender × Driver Age
3. Driver Age × Vehicle Age
4. Vehicle Type × Driver Age
5. Gender × Vehicle Type
6. Vehicle Type × No Claim Discount
7. Vehicle Type × Vehicle Age
8. Gender × No Claim Discount
9. Gender × Vehicle Age

Table 4.6: Order of Interactions Included in the Model

the same signal can possibly be captured but at a higher computational cost. Therefore, the linearity characteristic is not always undesirable when trying to capture linear signals.

4.4.4 Model Comparison

To compare the fitted models, a summary is presented in Table 4.7 with all the models fitted, the R packages used and the parameters estimated for each model.

Results of the fitted models for out-of-sample predictions reveal that the model with fewer variables (lasso vs. GLM) can capture the same signal and generalizes better. Lasso regularized GLMs are capable of performing variable selection to capture the same signal and reduce the number of variables included in the model. This is useful because it chooses the variables that can predict the data well without overfitting. However, these results also show that global models do not capture nonlinearities and interactions among variables. This indicates that results obtained from the group-lasso interaction network model are more conclusive given their ability to capture linear and non-linear effects while performing variable selection.

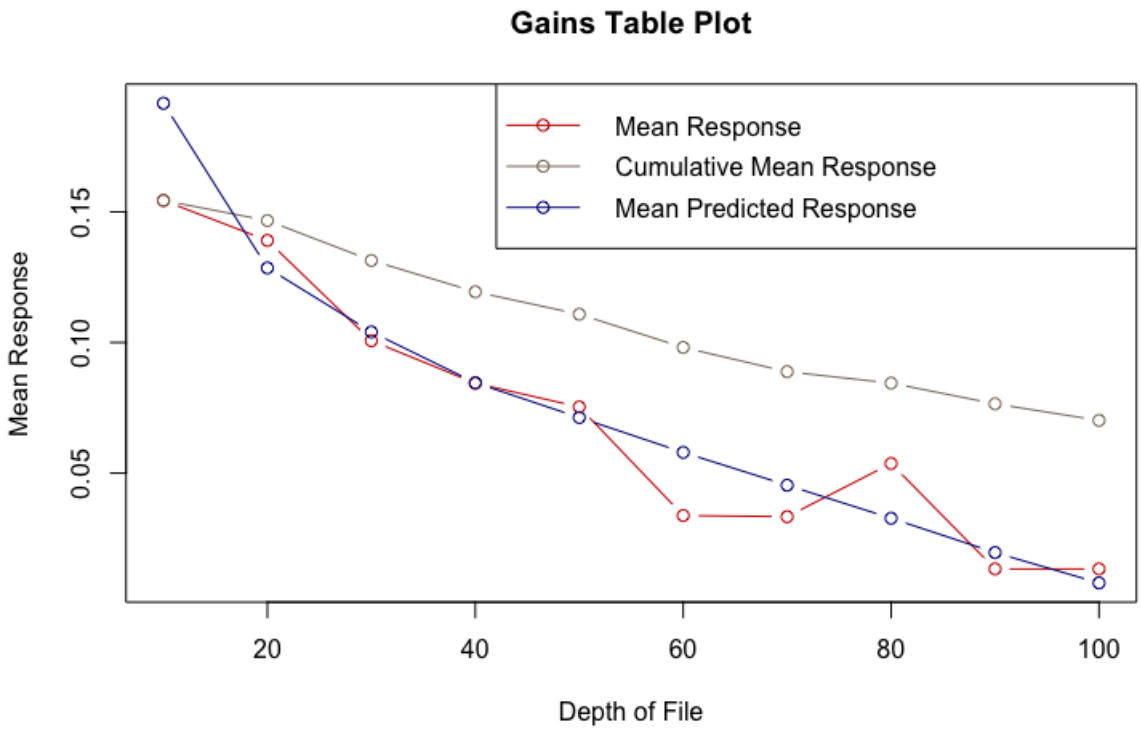

Figure 4.6: Gains for Group-Lasso Interaction Network

Comparing models with the Mean Square Error (MSE) is not the optimal, but it is hard to find a common metric for model comparison. An example is given in Figure 4.8 with the MSE for train and test errors, showing a lasso GLM with different λ values for the fitted models. Clearly, a higher penalty value, which would lead to lower number of covariates in the model, returns lower error rates. This shows again that variable selection is essential for building models that can generalize well on out-of-sample predictions. It is to be noted, that the train error decreases as the penalty increase which is mainly because the objective function to be minimized on likelihood and not least square errors. Generally, low training error could indicate that the model overfits and would then poorly generalizes out of sample results.

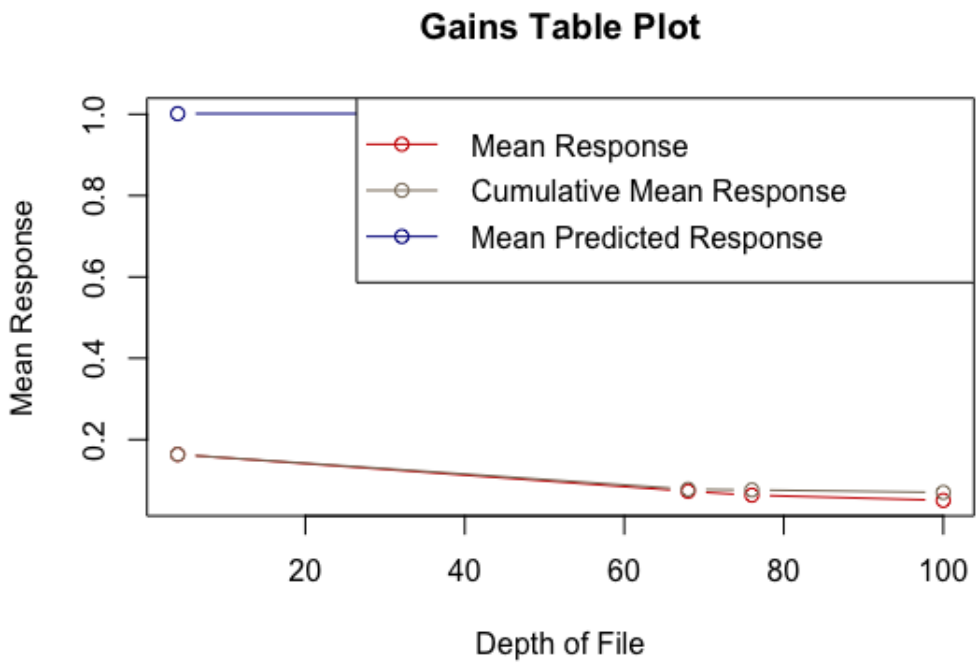

Figure 4.7: Gains for Gradient Boosting Model

4.5 Example 2: Ontario Collision Data

The model was applied to a subset of the Ontario collision coverage data from a large Canadian insurer. The fitted subset is composed of ten categorical and continuous variables combined. *Glinternet*2 returns results for a grid of 50 λ values and the corresponding error. Cross Validation error results from the frequency model, accounting for policy exposure, are shown in Figure 4.9. It is evident, that the cross validation error increases as the λ value decreases and more variables are captured by the model. Recall that a λ equal to zero returns an unpenalized model. We conclude that the cross validation error rate is lower when imposing a penalty, which goes back to the fact that a regularized model with fewer variables does not over-fit and thus predicts better than a model fitted with all available variables. The vertical dotted line shows optimal the λ value that minimizes the cross validation error.

We proceed by presenting the Lift chart for model assessment in Figure 4.10 which shows

Models Fitted and Comparison		
Method	R Package	Parameters
Generalized Linear Model	glm	Regression coefficients
lasso	glmnet	Regression coefficients, selection of optimal λ via CV
Group-Lasso	glinternet2	Regression coefficients, selection of optimal λ via CV for main and interaction effects
GBM	gbm	Number of trees, interaction depth set to 2 or 3 , learning rate lr = 0.01

Table 4.7: Summary of the Models

the trend of the predictions vs the trend of the actual data. The predictions are ordered in descending order and the trend is shown by grouping the data into ten buckets of equal size and comparing the predicted means of each group vs the observed mean. This helps identify groups that are over- or under-estimated by looking at the vertical distance between the predicted mean and the observed mean. Figure 4.10 shows the predictions obtained from the model with the optimal λ value. The trend observed is that model predictions over-estimate a specific risk segment, it matches the mean response values for another segment and under-estimates for other risk segments. Could that be an indication that we should increase premiums for "bad risks" and decrease premium for "good risks"?

The same model is fitted again on a different dataset. Both datasets are for the same coverage however for different geographical regions in Canada. From Figure 4.11, it is evident that the predictions give the same trend as the mean response values. However, the predicted values are slightly inflated over the real mean response values. The model does overestimate. Nevertheless, the observed means curve is decreasing and the predicted means curve is not far from it.

*Glinternet*2 was also used here to predict claim severities. In general, to build a severity model it was first fitted only on observations that had recorded at least one claim in the past.

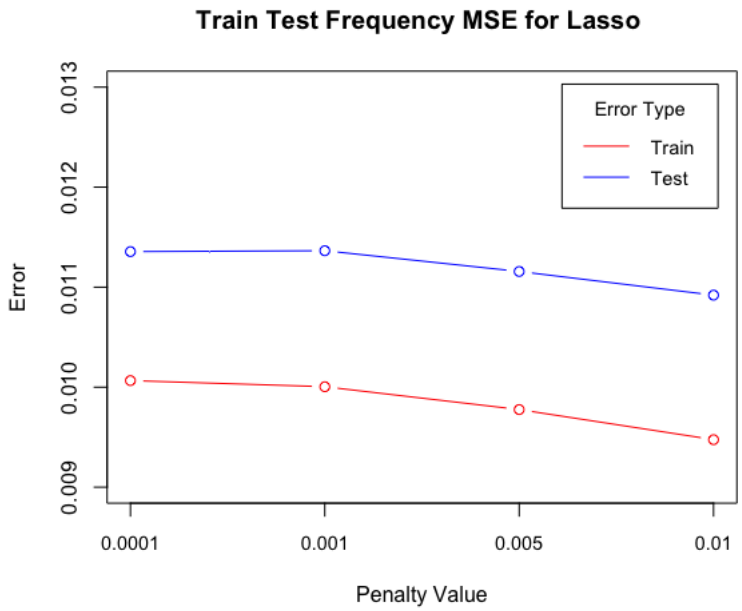

Figure 4.8: MSE For Test Train lasso with Different Penalty Values

This model is then used predict claim severities of all observations. In practice, this model is then used in combination with the frequency model to get the aggregate loss cost or also called pure premium. Similar to the Poisson model, the gamma model has an increasing cross validation error trend, except for the first few penalty values shown in Figure 4.12. Cross validation error sharply decreases, but then it increases again as the penalty decreases, at a flatter rate compared to the frequency model. For this example, it is not as clear that a penalty improves the model accuracy. However, a difference in error rates is still recognizable through the λ grid, showing that as the penalty decreases, the cross validation error increases. This indicates that a model with a higher penalty does not over-fit and can generalize better for out-of-sample predictions.

For severity predictions, the Lift chart in Figure 4.13 shows the same decreasing trend as that of the severity mean response. However, the mean predicted responses show an over-estimation of the mean response at all levels of risk. It is noticeable that the model does not differentiate much between predictions in different risk classes, which can be due to: (1) the model was fitted to a much smaller dataset (only policies that did file a claim) and (2) due

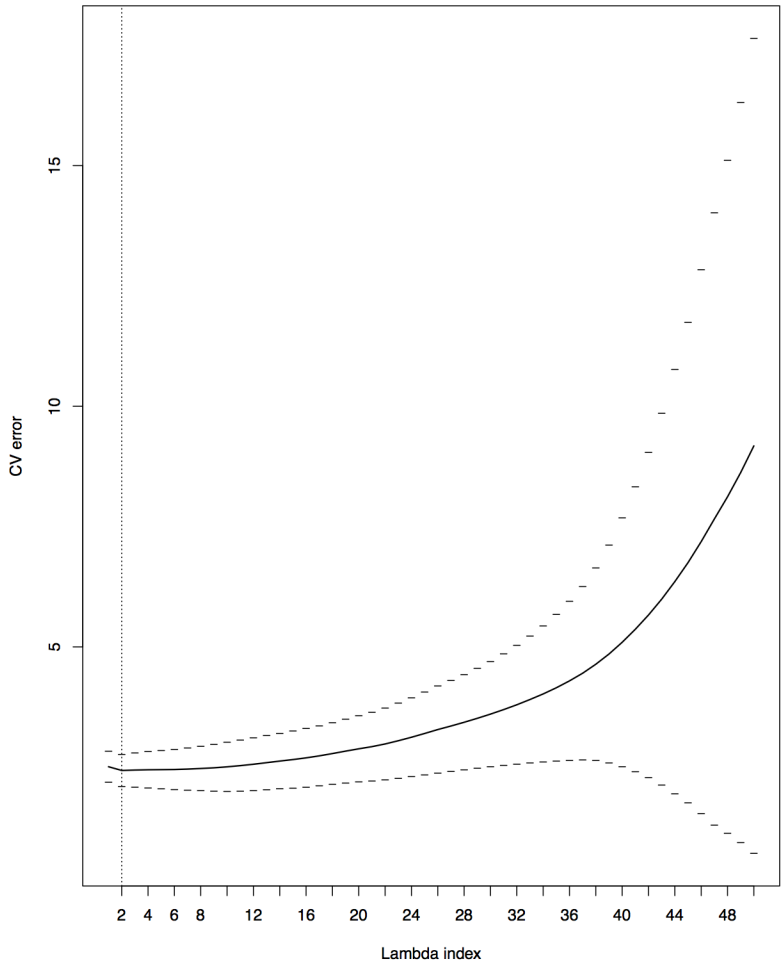

Figure 4.9: Cross validation Error for Frequency Model Fit

to limitations of the algorithm in terms of memory. The overall trend is the same as that of the observed data.

It is sometimes tricky to find a common metric to compare models. Therefore, we used few different metrics to validate our results and conclusions. Comparisons between a GLM, GLMNET and Glinernet2 for a claim frequency model are conducted, since the package "glmnet" does not include a gamma distribution. And adding that family to "glmnet" was not the focus of the work. All results are presented for the train and test sets to validate the model out of sample performance. We use the mean square error (MSE) as a common metric, shown in Figure 4.14 for all models. We can conclude that *glinternet*2 improves both

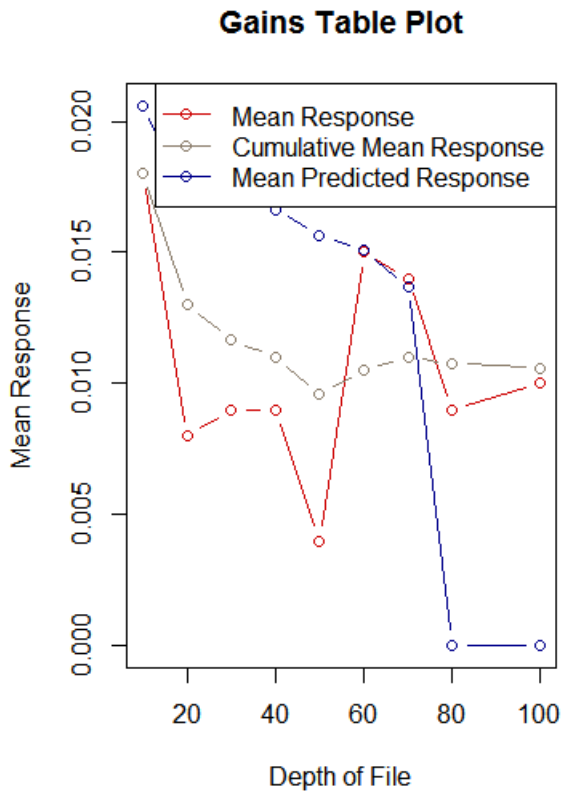

Figure 4.10: Lift Chart for Glinternet2 Frequency Predictions 1

the train and test sets errors. A lasso GLM with the GLMNET package was fitted to the same data set with a penalty of 0.001. Results in Figure 4.15 show that the observed mean and predicted mean are almost superimposed. However, these are in-sample predictions.

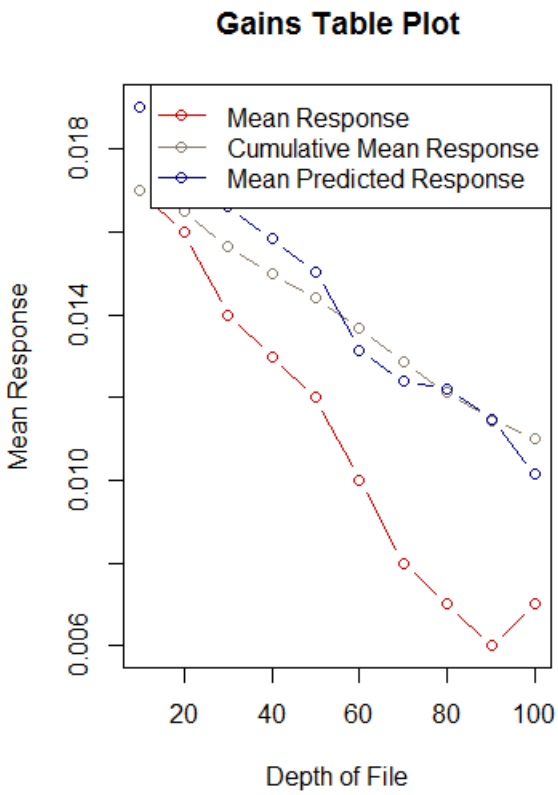

Figure 4.11: Lift Chart for Glinternet2 Frequency Predictions 2

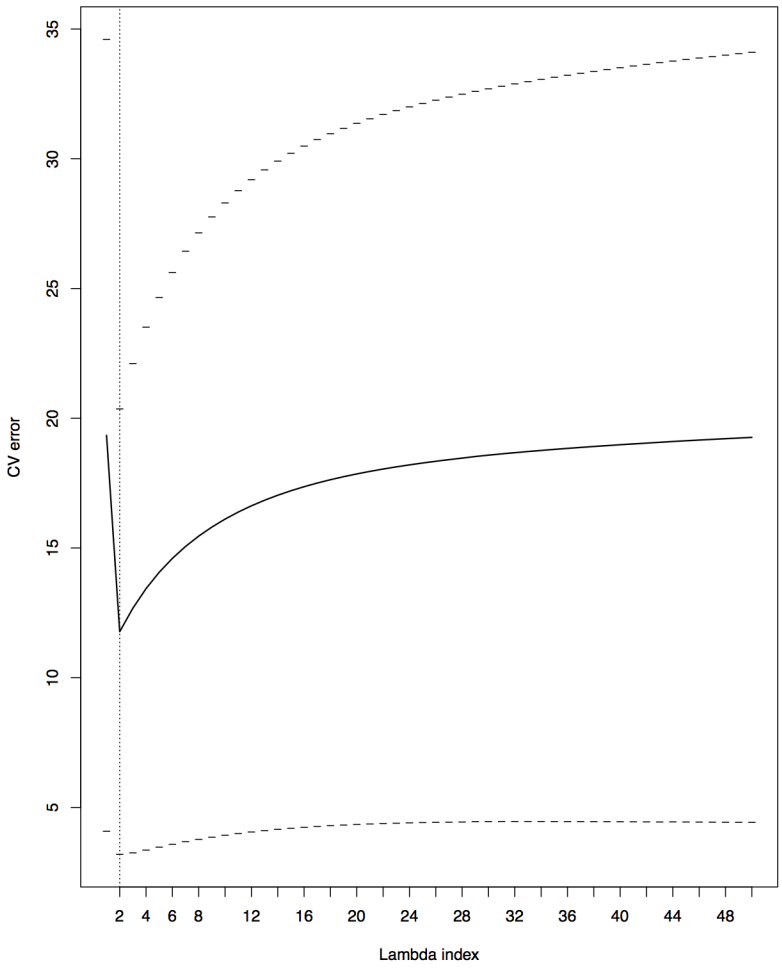

Figure 4.12: Cross Validation Error for the Severity Model Fit

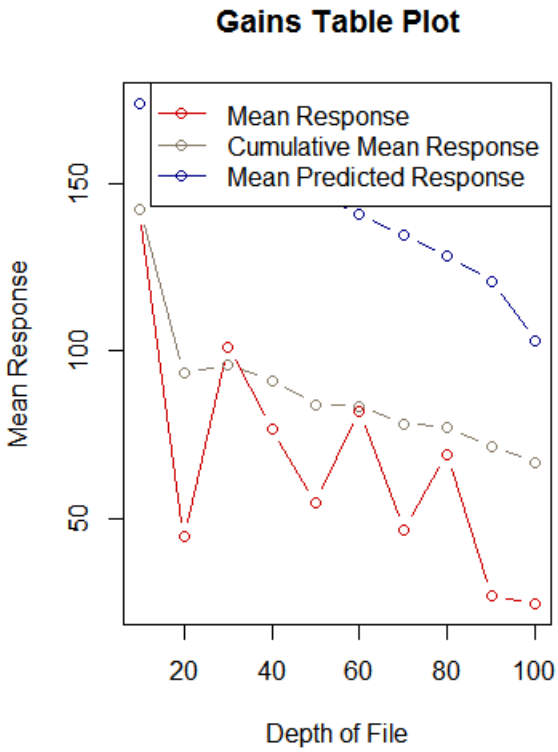

Figure 4.13: Lift Chart for Glinternet2 Severity Predictions

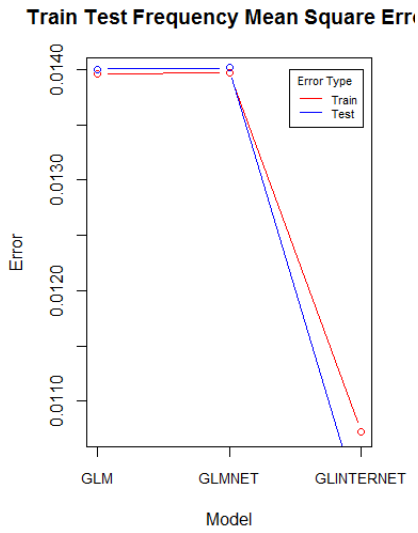

Figure 4.14: Mean Square Error for GLM, GLMNET, Glinternet

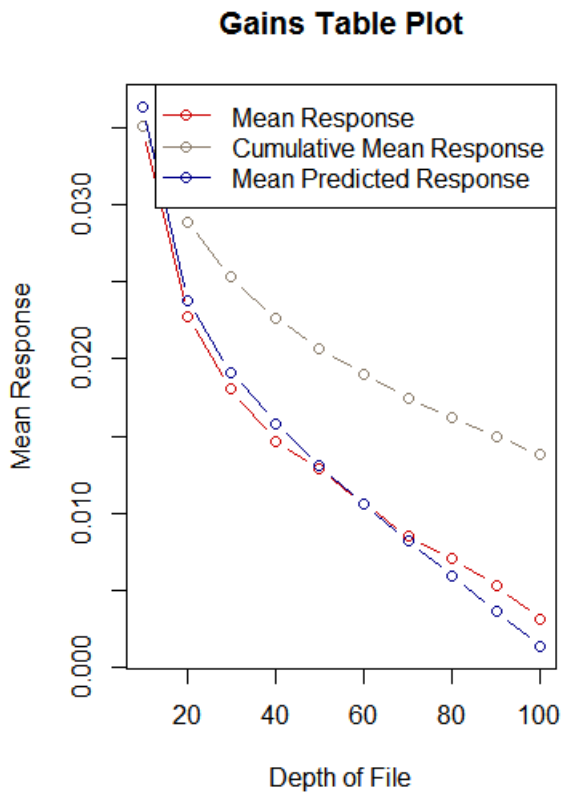

Figure 4.15: Lift Chart for GLMNET with Penalty 0.0001 for Frequency Predictions

Conclusions

Dealing with high dimensional data is a challenge in terms of computational power and fitting methods. Using regularization methods for high-dimensional data analysis is necessary to avoid overfitting and helps in variable selection without losing in model predictability. Nonetheless, linear models are not always sufficiently discerning. So adding interaction effects in the model is important to improve model predictability.

Detecting interactions is a tricky problem when the number of variables is large and it is important to have an automatic way to carry this out. The group-lasso interaction network supposes a framework that combines the detection of non-linear effects and the advantages of linear models. The code developed is still slow and runs out of memory quickly, due to the high number of iterations that the model conducts to find the candidate set and fit the model. Improving the code for fitting this model is a suggestion for further work.

Methods based on a machine learning techniques can add value to the limitations of linear models. Results obtained from models that combine linear and non-linear models effects return better predictions for insurance data, compared to standard linear approaches. Combing linear and non-linear modeling techniques composes a good representation of frequency-severity claims data. Generally, multivariate analytical techniques focus on individual level data, so that estimates of risks are more granular. The greater discrimination (larger number of risk groups) between individual risks the more accurate the pricing. This incremental level of accuracy in predicting losses enables insurers to price policies more accurately than competitors, hence improving portfolio profitability and providing a substantial long term competitive advantage.

In my future actuarial work and research endeavor I want to continue improving existing models through analytical and machine learning techniques. These show promising results to solve current issues in actuarial modeling such as finding interactions, modeling dependencies, variability, over-fitting and predictive power.

Bibliography

A. Beck and M. Teboulle. A fast iterative shrinkage thresholding algorithm for linear inverse problems. *Journal of Imaging Science and Technology*, 2(1):183–202, 2009.

C. E. J. Becker, S.R. and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, Stanford, California, 2007.

P. Buhlmann and S. Van de Geer. *Statistics for High-Dimensional Data Methods*. Springer Verlag, 2011.

E. W. Frees. *Regression Modeling with Actuarial and Financial Applications*. Cambridge University Press, 2009.

E. W. Frees, R. A. Derrig, and G. Meyers. *Predictive Modeling Applications in Actuarial Science*. Cambridge University Press, 2014.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1995.

Y. Freund and R. E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *Technical Report, Department of Statistics, Stanford University*, 2010.

J. H. Friedman. Stochastic gradient boosting. *Journal Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

A. Gelman and J. Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2007.

J. Guszcza. Hierarchical growth curve models for loss reserving. *Casualty Actuarial Society E-Forum*, 2008.

T. Hastie and M. Lim. Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics*, 24(3):627–654, 2015.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, Stanford, 2008.

G. James, D. Witten, and R. Hastie, Trevor Tibshirani. *An Introduction to Statistical Learning*. Springer, Stanford, 2014.

P. McCullagh and J. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.

N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society*, 58(1):267–288, 1996.

R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society*, 2012.

L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group lasso. *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

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

Appendix

Model Output from Chapter 4

Variable	Level	Coefficient
mainEffectsCoefcat[[1]]	1	-0.1968526
mainEffectsCoefcat[[1]]	2	0.1968526
mainEffectsCoefcat[[2]]	1	-0.978796651
mainEffectsCoefcat[[2]]	2	0.006785007
mainEffectsCoefcat[[2]]	3	0.971441586
mainEffectsCoefcat[[3]]	1	-0.151647083
mainEffectsCoefcat[[3]]	2	-0.001517815
mainEffectsCoefcat[[3]]	3	0.159276457
mainEffectsCoefcont[[1]]	..	0.9448233
mainEffectsCoefcont[[2]]	..	0.8008501
mainEffectsCoefcont[[3]]	..	0.9566645
mainEffectsCoefcont[[4]]	..	0.9457888
mainEffectsCoefcont[[5]]	..	0.762109
mainEffectsCoefcont[[6]]	..	0.2521866
mainEffectsCoefcont[[7]]	..	0.4854388

Table 4.8: Glinternet 10 Main Effect Coefficients of Simulation Study

Variable	Level	Coefficient
interactionsCoefcatcat[[1]]	cat2-0-cat3-0	0.058579801
interactionsCoefcatcat[[1]]	cat2-1-cat3-0	-0.001547802
interactionsCoefcatcat[[1]]	cat2-2-cat3-0	-0.057031999
interactionsCoefcatcat[[1]]	cat2-0-cat3-1	-0.058579801
interactionsCoefcatcat[[1]]	cat2-1-cat3-1	0.001547802
interactionsCoefcatcat[[1]]	cat2-2-cat3-1	0.057031999
interactionsCoefcontcont[[1]]	..	0.5216175
interactionsCoefcontcont[[2]]	..	0.6071549
interactionsCoefcontcont[[3]]	..	0.6140319
interactionsCoefcontcont[[4]]	..	0.1526564
interactionsCoefcatcont[[1]]	1	-0.0722799571
interactionsCoefcatcont[[1]]	2	0.0003683313
interactionsCoefcatcont[[1]]	3	0.0719116257
interactionsCoefcatcont[[2]]	1	-0.732349782
interactionsCoefcatcont[[2]]	2	0.001258306
interactionsCoefcatcont[[2]]	3	0.731091476
interactionsCoefcatcont[[3]]	1	-8.522093e-01
interactionsCoefcatcont[[3]]	2	-1.873492e-05
interactionsCoefcatcont[[3]]	3	8.522280e-01
interactionsCoefcatcont[[4]]	1	-0.691077684
interactionsCoefcatcont[[4]]	2	-0.001458876
interactionsCoefcatcont[[4]]	3	0.692536560
interactionsCoefcatcont[[5]]	1	-0.49415496
interactionsCoefcatcont[[5]]	2	-0.00626617
interactionsCoefcatcont[[5]]	3	0.50042113

Table 4.9: Glinternet 10 Interaction Coefficients of Simulation Study

Variable	Level	Coefficient
mainEffectsCoefcat[[1]]	1	-1.222511e-04
mainEffectsCoefcat[[1]]	2	-2.069183e-05
mainEffectsCoefcat[[2]]	1	-5.633661e-05
mainEffectsCoefcat[[2]]	2	-6.238605e-05
mainEffectsCoefcat[[3]]	1	-0.000148417
mainEffectsCoefcat[[3]]	2	0.000000000
mainEffectsCoefcont[[1]]	..	1.880254e-05
mainEffectsCoefcont[[2]]	..	-1.872523e-06

Table 4.10: Glinternet2 5 Main Effect Coefficients for Singapore Insurance Data

Variable	Level	Coefficient
interactionsCoefcatcat[[1]]	cat1-0-cat2-0	2.060756e-05
interactionsCoefcatcat[[1]]	cat1-1-cat2-0	-2.060756e-05
interactionsCoefcatcat[[1]]	cat1-0-cat2-1	-2.060756e-05
interactionsCoefcatcat[[1]]	cat1-1-cat2-1	2.060756e-05
interactionsCoefcatcat[[2]]	cat1-0-cat3-0	3.325506e-07
interactionsCoefcatcat[[2]]	cat1-1-cat3-0	-3.325506e-07
interactionsCoefcatcat[[2]]	cat1-0-cat3-1	-3.325506e-07
interactionsCoefcatcat[[2]]	cat1-1-cat3-1	3.325506e-07
interactionsCoefcatcat[[3]]	cat2-0-cat3-0	1.503586e-06
interactionsCoefcatcat[[3]]	cat2-1-cat3-0	-1.503586e-06
interactionsCoefcatcat[[3]]	cat2-0-cat3-1	-1.503586e-06
interactionsCoefcatcat[[3]]	cat2-1-cat3-1	1.503586e-06
interactionsCoefcontcont[[1]]	..	8.038371e-08
interactionsCoefcatcont[[1]]	1	1.813307e-06
interactionsCoefcatcont[[1]]	2	-1.813307e-06
interactionsCoefcatcont[[2]]	1	4.816211e-07
interactionsCoefcatcont[[2]]	2	-4.816211e-07
interactionsCoefcatcont[[3]]	1	2.141412e-06
interactionsCoefcatcont[[3]]	2	-2.141412e-06
interactionsCoefcatcont[[4]]	1	-3.948223e-11
interactionsCoefcatcont[[4]]	2	3.948223e-11
interactionsCoefcatcont[[5]]	1	-1.34592e-07
interactionsCoefcatcont[[5]]	2	1.34592e-07

Table 4.11: Glinternet2 9 Interaction Coefficients for Singapore Insurance Data